



**SOSYAL BİLİMLER LİSESİ ÖĞRENCİLERİNE  
BLOK TABANLI PROGRAMLAMA  
ÖĞRETİMİNİN KAYGI, BİLİŞSEL YÜK VE  
BAŞARIYA ETKİSİ**

**Alper ÜNAL**

**Yüksek Lisans Tezi**

**Bilgisayar ve Öğretim Teknolojileri Eğitimi Ana Bilim  
Dalı**

**2019**

(Her hakkı saklıdır.)

T.C.  
ATATÜRK ÜNİVERSİTESİ  
EĞİTİM BİLİMLERİ ENSTİTÜSÜ  
BİLGİSAYAR VE ÖĞRETİM TEKNOLOJİLERİ EĞİTİMİ ANA BİLİM DALI  
BİLGİSAYAR VE ÖĞRETİM TEKNOLOJİLERİ EĞİTİMİ BİLİM DALI

**SOSYAL BİLİMLER LİSESİ ÖĞRENCİLERİNE BLOK TABANLI  
PROGRAMLAMA ÖĞRETİMİNİN KAYGI, BİLİŞSEL YÜK VE BAŞARIYA ETKİSİ**  
(The Effect of Block-Based Programming on Social Science High School Students' Anxiety,  
Cognitive Load and Achievement)

YÜKSEK LİSANS TEZİ

Alper ÜNAL

Danışman: Dr. Öğr. Üyesi Fatma Burcu TOPU

Erzurum  
Mayıs, 2019

## KABUL VE ONAY TUTANAĞI

Alper ÜNAL tarafından hazırlanan “Sosyal Bilimler Lisesi Öğrencilerine Blok Tabanlı Programlama Öğretiminin Kaygı, Bilişsel Yük ve Başarıya Etkisi” başlıklı çalışması 08/05/2019 tarihinde yapılan tez savunma sınavı sonucunda başarılı bulunarak jürimiz tarafından Bilgisayar ve Öğretim Teknolojileri Eğitimi Ana Bilim Dalı, Bilgisayar ve Öğretim Teknolojileri Eğitimi Bilim Dalında yüksek lisans tezi olarak kabul edilmiştir.

Jüri Başkanı: Prof. Dr. Yüksel GÖKTAŞ

*Atatürk Üniversitesi*

Danışman: Dr. Öğr. Üyesi Fatma Burcu TOPU

*Atatürk Üniversitesi*

Jüri Üyesi: Doç. Dr. Ünal ÇAKIROĞLU

*Trabzon Üniversitesi*



Bu tezin Atatürk Üniversitesi Lisansüstü Eğitim ve Öğretim Yönetmeliği'nin ilgili maddelerinde belirtilen şartları yerine getirdiğini onaylarım.

29 Mayıs 2019



Prof. Dr. Mustafa SÖZBİLİR

Enstitü Müdürü

## ETİK VE BİLDİRİM SAYFASI

Yüksek Lisans Tezi olarak sunduğum “Sosyal Bilimler Lisesi Öğrencilerine Blok Tabanlı Programlama Öğretiminin Kaygı, Bilişsel Yük Ve Başarıya Etkisi” başlıklı çalışmanın tarafımdan bilimsel etik ilkelere uyularak yazıldığını ve yararlandığım eserleri kaynakçada gösterdiğimi beyan ederim.

08 / 05 / 2019  
  
Alper ÜNAL

Tezle ilgili patent başvurusu yapılması / patent alma sürecinin devam etmesi sebebiyle Enstitü Yönetim Kurulunun .../.../.... tarih ve ..... sayılı kararı ile teze erişim 2 (iki) yıl süreyle engellenmiştir.

Enstitü Yönetim Kurulunun .../.../.... tarih ve ..... sayılı kararı ile teze erişim 6 (altı) ay süreyle engellenmiştir.

## TEŞEKKÜR

Yüksek lisans tezi olarak tamamladığım bu çalışmada; süreç boyunca hep yanımda olan, bana yol gösteren, desteğini hiçbir zaman esirgemeyen gösterdiği katkılardan dolayı danışmanım Dr. Öğr. Üyesi Fatma Burcu TOPU'ya saygılarımı ve teşekkürlerimi sunuyorum.

Önerilerinden dolayı savuma sınavımda bulunan Atatürk Üniversitesi öğretim üyesi Prof. Dr. Yüksel GÖKTAŞ'a ve Trabzon Üniversitesi öğretim üyesi Doç. Dr. Ünal ÇAKIROĞLU'na teşekkürlerimi sunuyorum. Veri toplama araçlarının geliştirilmesinde emeği geçen ve uygulama sürecinin sağlıklı yürütülmesine katkı sağlayan meslektaşlarıma şükranlarımı sunuyorum.

Öğrenim hayatımın her aşamasında manevi desteklerini yanımda hissettiğim eşime ve aileme bana gösterdikleri anlayıştan ötürü sevgilerimi sunuyorum.

Alper ÜNAL

## ÖZ

### YÜKSEK LİSANS TEZİ

#### SOSYAL BİLİMLER LİSESİ ÖĞRENCİLERİNE BLOK TABANLI PROGRAMLAMA ÖĞRETİMİNİN KAYGI, BİLİŞSEL YÜK VE BAŞARIYA ETKİSİ

Alper ÜNAL

Mayıs 2019, 157 sayfa

**Amaç:** Bu çalışmada; sosyal bilimler lisesi bilgisayar bilimi dersinde Blockly görsel programlama aracının kullanıldığı ve kullanılmadığı hazırlık sınıflarındaki programlama eğitiminin öğrencilerin kaygı seviyelerine, bilişsel yüklerine ve akademik başarılarına etkisi incelenmiştir. Ayrıca araştırmada öğrencilerin programlama kaygılarını, bilişsel yüklerini ve başarılarını etkileyen faktörler ortaya konmuş ve uygulamaya yönelik öğrenci görüşleri belirlenmiştir.

**Yöntem:** Çalışmada, karma araştırma desenlerinden Çeşitleme kullanılmıştır. Çalışma grubu, Alanya Sosyal Bilimler Lisesi’nde öğrenim gören toplam 90 öğrenciden oluşmaktadır. “Bilgisayar Programlama Kaygı Ölçeği”, üç alt boyuttan oluşan “Bilişsel Yük Ölçeği”, “Programlama Başarı Testi” ve yarı yapılandırılmış görüşme formu kullanılmıştır.

**Bulgular:** Öğrencilerin kaygı seviyeleri arasında anlamlı bir fark görülmemiştir. Öğrencilerin bilişsel yük seviyelerinde ise gruplar arası anlamlı bir fark görülmezken blok tabanlı programlama aracı kullanılan grubun bilişsel yük seviyesinin daha az olduğu saptanmıştır. Öğrencilerin programlama başarı seviyelerinde ise blok tabanlı programlama öğretimi yapılan grup lehine anlamlı bir farkın olduğu tespit edilmiştir. Çalışmanın nitel boyutunda ise öğrencilerin programlama kaygılarını, bilişsel yüklerini ve başarılarını etkileyen faktörler olduğu görülmüştür.

**Sonuç:** Blok tabanlı programlama aracı kullanımının öğrencilerin programlama kaygılarını olumlu yönde etkilediği, bilişsel olarak daha az yükledikleri ve akademik olarak daha başarılı oldukları görülmektedir. Bu çalışmanın ortaöğretim müfredatına eklenen bilgisayar bilimi dersi ile ilgili olarak yapılacak çalışmalara öncülük yapacağı, ayrıca ileri paragramlama becerisi gerektiren ve ortaöğretim düzeyinde yapılacak programlama eğitimi temelli çalışmalara da yol göstereceği düşünülmektedir.

**Anahtar Kelimeler:** blok tabanlı görsel programlama, programlama kaygısı, bilişsel yük.

## ABSTRACT

### MASTER'S THESIS

#### THE EFFECT OF BLOCK-BASED PROGRAMMING ON SOCIAL SCIENCE HIGH SCHOOL STUDENTS' ANXIETY, COGNITIVE LOAD AND ACHIEVEMENT

Alper ÜNAL

May 2019, 157 Pages

**Purpose:** In this study, the effect of programming education on anxiety levels, cognitive loads and achievements of the students, given in the computer science course of social sciences high school' preparatory classes, in some of which Blockly visual programming tool is used and in some of which not, was examined. Besides, factors affecting their programming anxieties, cognitive loads and achievements was put forward in the present study, and students' perspectives were determined about this study.

**Method:** In the study, the method of Triangulation in the mixed research patterns was used. The study group consisted of a total of 90 students, of which the Alanya Social Sciences High School. "Computer Programming Anxiety Scale", three subdimensional "Cognitive Load Scale", "Programming Achievement Test" and a semi-structured interview form were used.

**Findings:** There is not a significant difference between the the anxiety levels of the students. While it wasn't seen between the cognitive levels of the groups, that of the group utilizing block-based programming was found lower. A significant difference between the achievement level was found in favor of the group given block-based programming education. It was found from the qualitative aspect of the study that factors affecting students' the programming anxieties, cognitive loads, and achievement.

**Conclusion:** It was found that the utilization of block-based programming positively affect the programming anxieties of the students, and they are loaded less cognitively, and became academically more successful. It is believed that the this study will guide the studies of computer science included in the secondary school curriculum, and the programming based studies requiring advanced programming skills to be conducted on secondary school level.

**Keywords:** block-based visual programming, programming anxiety, cognitive load.

## İÇİNDEKİLER

KABUL VE ONAY TUTANAĞI .....	i
ETİK VE BİLDİRİM SAYFASI.....	ii
TEŞEKKÜR .....	ii
ÖZ.....	iii
ABSTRACT .....	iv
İÇİNDEKİLER.....	v
TABLolar DİZİNİ.....	viii
ŞEKİLLER DİZİNİ .....	ix
BİRİNCİ BÖLÜM.....	1
Giriş .....	1
Araştırmanın Amacı.....	6
Araştırmanın Önemi ve Gerekçesi.....	6
Sınırlılıklar .....	8
Varsayımlar.....	9
Terim ve Tanımlar .....	9
İKİNCİ BÖLÜM .....	11
Kuramsal Çerçeve ve İlgili Araştırmalar .....	11
Programlama Öğretimi .....	11
Programlama dilleri. ....	12
Programlama eğitiminde yaşanan güçlükler.....	14
Görsel Programlama Öğretim Araçları.....	15
Blok tabanlı görsel programlama aracı olarak Blockly. ....	18
Görsel programlama ile ilgili araştırmalar.....	24
Programlama Öğretimi ve Kaygı.....	33
Programlama Öğretimi ve Bilişsel Yük.....	35
İçsel bilişsel yük. ....	35
Dışsal bilişsel yük. ....	36
Etkili bilişsel yük. ....	36
Bölüm Özeti.....	38
ÜÇÜNCÜ BÖLÜM.....	40



Yöntem .....	40
Araştırma Deseni .....	40
Çalışma Grubu .....	41
Veri Toplama Araçları .....	43
Bilgisayar programlama kaygı ölçeği .....	43
Bilişsel yük ölçeği .....	44
Programlama başarı testi .....	45
Görüşme formu .....	47
Veri toplama araçlarının geçerliliği ve güvenilirliği .....	48
Uygulama Süreci .....	50
Veri Analizi .....	52
Nicel veri analizi .....	52
Nitel veri analizi .....	54
Araştırmacının Rolü .....	54
Çalışmanın Geçerlik ve Güvenirliği .....	55
Bölüm Özeti .....	56
DÖRDÜNCÜ BÖLÜM .....	57
Bulgular .....	57
Öğrencilerin Programlamaya Yönelik Kaygı Seviyelerinin Gruplara Göre Farkı .....	57
Öğrencilerin Bilişsel Yük Seviyelerinin Gruplara Göre Farkı .....	58
Öğrencilerin Programlama Başarı Puanlarının Gruplara Göre Farkı .....	61
Gruplara Göre Öğrencilerin Programlamaya Yönelik Kaygıları, Bilişsel Yükleri ile Akademik Başarıları Arasındaki İlişki .....	63
BG ve PG Öğrencilerinin Programlama Öğretim Sürecinde Kaygılarını, Bilişsel Yüklerini ve Programlama Başarılarını Etkileyen Faktörler .....	64
Programlama kaygısını etkileyen faktörler .....	65
Bilişsel yükü etkileyen faktörler .....	68
BG ve PG öğrencilerinin programlama öğretim sürecine yönelik görüşleri .....	74
BEŞİNCİ BÖLÜM .....	80
Tartışma ve Sonuç .....	80
Blok Tabanlı Görsel Programlama Aracının Kullanıldığı (BG) ve Kullanılmadığı (PG) Öğretim Ortamlarında Öğrencilerin Programlama Kaygıları .....	80
Blok Tabanlı Görsel Programlama Aracının Kullanıldığı (BG) ve Kullanılmadığı (PG) Öğretim Ortamlarında Öğrencilerin Bilişsel Yükleri .....	81
Blok Tabanlı Görsel Programlama Aracının Kullanıldığı (BG) ve Kullanılmadığı (PG) Öğretim Ortamlarında Öğrencilerin Programlama Başarıları .....	84
BG ve PG Öğrencilerinin Programlamaya Yönelik Kaygıları, Bilişsel Yükleri ile Akademik Başarıları Arasındaki İlişki .....	85

BG ve PG Öğrencilerinin Programlama Öğretim Sürecindeki Kaygılarını, Bilişsel Yüklerini ve Akademik Başarılarını Etkileyen Faktörler.....	86
Programlama kaygısını etkileyen faktörler.....	86
Bilişsel yükü etkileyen faktörler.....	87
Programlama başarısını etkileyen faktörler.....	89
BG ve PG Öğrencilerinin Programlama Öğretim Sürecine Yönelik Görüşleri.....	89
Öneriler.....	92
KAYNAKÇA.....	94
EKLER.....	112
Ek-1. Bilgisayar Programlama Kaygı Ölçeği.....	112
Ek-2. Bilişsel Yük Ölçeği (Haftalık).....	113
Ek-3. Başarı Testi (Blockly).....	128
Ek-4. Başarı Testi (Python Editör).....	137
Ek-5. Görüşme Formu.....	142
Ek-6. Blockly Tabanlı Programlama Grubu Kamera Görüntüleri.....	143
Ek-8. Blockly Grubu ve Python Grubu Programlama Etkinlik İçerikleri.....	146
Ek-9. Blok Tabanlı Programlama Araçlarıyla İlgili Araştırmalar.....	152
Ek-10. Araştırma İzni.....	155
ÖZ GEÇMİŞ.....	158

## TABLolar DİZİNİ

Tablo 1. Görsel Programlama Ortamları .....	17
Tablo 2. 2012-2019 Yıllarında Yaygın Olarak Kullanılan Blok Tabanlı Programlama Araçları.....	25
Tablo 3. Yaygın Olarak İncelenen Değişkenler.....	25
Tablo 4. Araştırma Sorularına Göre Araştırma Desenleri, Veri Toplama Araçları ve Veri Analiz Teknikleri.....	41
Tablo 5. Veri Toplama Araçlarına Göre Öğrenci Sayıları .....	42
Tablo 6. Gruplara Göre Ön-Test Programlama Başarı Düzeyleri Arasındaki Fark.....	42
Tablo 7. Bilişsel Yük Ölçeğindeki Sorular ve Ait Oldukları Alt Boyutlar .....	45
Tablo 8. Programlama Başarı Testi Sorularının Kazanımlara Göre Dağılımı .....	46
Tablo 9. Programlama Başarı Testi Madde Analizleri .....	46
Tablo 10. Uygulama Takvimi .....	50
Tablo 11. Hata Varyanslarının Homojenliği.....	53
Tablo 12. Çalışmada Alınan Geçerlik ve Güvenirlik Önlemleri .....	55
Tablo 13. Gruplara Göre Ön-Test ve Son-Test Programlama Kaygı Seviyeleri .....	57
Tablo 14. Öğrencilerin Son-Test Programlama Kaygı Seviyelerinin Gruplara Göre Farkı ....	57
Tablo 15. Gruplara Göre Öğrencilerin Bilişsel Yük Seviyeleri .....	59
Tablo 16. Öğrencilerin Toplam Bilişsel Yük Seviyelerinin Gruplara Göre Farkı .....	59
Tablo 17. Öğrencilerin 1. ve 10. Hafta Toplam Bilişsel Yük Seviyelerinin Gruplara Göre Farkı .....	60
Tablo 18. Grup İçi 1. ve 10. Hafta Toplam Bilişsel Yük Seviyeleri Farkı .....	60
Tablo 19. Gruplara Göre İçsel, Dışsal, Etkili Bilişsel Yük Arasındaki Farka Yönelik MANOVA Sonuçları.....	61
Tablo 20. Gruplara Göre Ön Test ve Son Test Programlama Başarı Seviyeleri.....	61
Tablo 21. Öğrencilerin Son-Test Programlama Başarı Seviyelerinin Gruplara Göre Farkı ...	62
Tablo 22. BG Öğrencilerinin Ön-Test ve Son-Test Programlama Başarı Seviyeleri Arasındaki Fark .....	62
Tablo 23. PG Öğrencilerinin Ön-Test ve Son-Test Programlama Başarı Seviyeleri Arasındaki Fark .....	63
Tablo 24. Gruplara Göre Kaygı, Bilişsel Yük ve Akademik Başarı Arasındaki İlişki.....	63
Tablo 25. Programlama Kaygısını Etkileyen Faktörler .....	65

Tablo 26. <i>Bilişsel Yükü Etkileyen Faktörler</i> .....	68
Tablo 27. <i>BG ve PG Öğrencilerinin Sürece Yönelik Görüşleri</i> .....	74



## ŞEKİLLER DİZİNİ

Şekil 1. Çalışmanın kuramsal çerçevesi.....	5
Şekil 2. 21. yüzyıl becerileri (Partnership21, 2018). ....	12
Şekil 3. Blockly arayüzü. ....	19
Şekil 4. 2017 yılında dünyadaki en popüler programlama dilleri.....	20
Şekil 5. Python ile diğer programlama dillerinin söz dizimlerinin karşılaştırıldığı bir örnek. .	21
Şekil 6. Blockly'nin Python sözdizimine uygun hale getirildiği “değişkene değer atama” örneği.....	22
Şekil 7. Koşul ifadeleri (Blockly kod blokları ve Python dili sözdizimi yapısı). ....	23
Şekil 8. Döngü yapısı (Blockly kod blokları ve Python dili sözdizimi yapısı).....	24
Şekil 9. Programlama kaygısını tetikleyen etmenler. ....	34
Şekil 10. Bilişsel yükü oluşturan alt boyutlar. ....	37
Şekil 11. Veri toplama süreci.....	43
Şekil 12. Veri toplama araçlarıyla ilgili geçerlilik ve güvenilirlik önlemler .....	49
Şekil 13. Uygulama süreci. ....	52
Şekil 14. Nicel veri analiz süreci. ....	52
Şekil 15. Nitel veri analiz süreci. ....	54
Şekil 16. Gruplara göre haftalık bilişsel yük seviyeleri. ....	58

## BİRİNCİ BÖLÜM

### Giriş

Endüstri 4.0 devrimiyle birlikte insan gücü ile yapılan işlerin bilgisayar ve internet teknolojileri kullanılarak makinalar tarafından yürütülmesi gündemde yer almaya başlamıştır. Bu sayede fabrikalar kendini yönetebilir hale gelmiş ve Bilişim Teknolojileri (BT) daha etkin bir şekilde sanayi alanında boy göstermeye başlamıştır (Bulut & Akçacı, 2017; Ertuğrul & Deniz; Fırat & Fırat, 2017; Koren & Klamma, 2018; Sener & Eleveli, 2017; Tanriogen, 2018; Yıldız, 2018). Bu değişimle birlikte BT farklı alanlarda da (askeri, tıp, uzay bilimleri, sanayi, ticaret, mühendislik vs.) genişleme göstermeye başlamıştır. Buna bağlı olarak hem diğer branşlardaki hem de eğitim alanındaki bireylerde aranan bazı özelliklerde revize ihtiyacı ortaya çıkmıştır (Şuman, 2017). OECD (2017) mevcut değişimi, öğrencilerin sosyal ve bilişsel beceri gibi özelliklerinin değişimi, sürekli çevrimiçi kalma durumunda olduklarından öğrencilerin teknolojiye bağlı kalma durumu olarak ifade etmektedir. 21. yüzyıl becerileri olarak adlandırılan bu özelliklere sahip insan gücü yetiştirmeye yönelik çeşitli çalışmalar yapılmakta (Pedro, 2006), tüm ülkeler bu değişimlere ayak uydurmaya çalışmaktadır. Son yıllarda yapılan çalışmalarda genelde algoritmik düşünme özelle kodlama becerisi de farklı üst düzey düşünme becerilerini kapsayan 21. yüzyıl bireylerinin taşıması gereken beceri olarak nitelendirilmektedir (Sayın & Seferoğlu, 2016, Shin, Park, & Bae, 2013). Bahsedilen becerilerin kazanılmasında ve bu becerilere yönelik yeni içeriklerin geliştirilmesinde yine BT önemli bir araç olmuştur (Uluyol & Eryılmaz, 2015).

Yapılan birçok araştırmada BT temel alınan programlama dersi kazanımlarının 21. yüzyıl öğrenci becerilerinin birçoğunu sağlamayı mümkün kıldığı, bu becerileri geliştirebildiği ve iyileştirebildiği görülmektedir (Jancheski, 2017). Bu nedenle birçok ülke programlama derslerini öğretim programlarına dahil etmeye başlamış birçoğu da bu dersi zorunlu tutmuştur (Grover & Pea, 2013). 2014 yılının Ekim ayında programlama dersleri üzerine Avrupa'da bir araştırma gerçekleştirilmiştir. Avrupa Komisyonu tarafından yapılan araştırma İngiltere, Danimarka, İtalya, Bulgaristan, Estonya, Çek Cumhuriyeti, Polonya, Litvanya, İrlanda, Kıbrıs, Portekiz ve Yunanistan gibi ülkelerin eğitim müfredatlarında programlama dersinin yer aldığını göstermiştir (BK, 2017). İngiltere'de Temel Bilgisayar Programlama dersi radikal değişiklikler yapılarak müfredata yeniden girmiştir (Burns, 2012).

İngiltere 2014 yılını “kodlama yılı ” ilan etmiş ve Avrupa Birliği adına bir adım atmıştır (Öymen, 2014).

2013 yılından itibaren Amerika’da 24 eyalette K-12 okullarında haftanın çeşitli günleri programlama dersleri verilmeye başlamıştır (Vaidyanathan, 2013). Ayrıca belirli özel okulların programlama etkinlikleri düzenledikleri görülmektedir (Borchers, 2013). Estonya, 7 yaş ve üzeri olan öğrencilerine programlama eğitimi vermeye başlamış, Güney Kore 2017 yılından itibaren kademeli olarak ilkokul ve ortaokullarda programlama eğitiminin zorunlu tutulacağını açıklamıştır (Özçakmak, 2014). Hindistan’da, ilkokul çağından başlamak üzere algoritma öğretimi, temel programlama dili becerileri ve ileri düzey programlama eğitimi kademe kademe uzun yıllardan beri eğitim müfredatında yer almaktadır. Avustralya’da 2015 yılı itibarıyla öğrencilere programlama eğitimi verilmeye başlanmış ayrıca ilkokul düzeyinde de bilgisayar dersleri kapsamında bir proje yürütülmektedir. Çin’de ise kodlama eğitimi okul öncesi eğitim dönemi ile birlikte başlamaktadır (Saygıner & Tüzün, 2017).

Dünya’da durum böyleyken Türkiye de bu değişime kayıtsız kalmamıştır. 2012 yılı sonunda, Milli Eğitim Bakanlığı (MEB) Talim Terbiye Kurulu Başkanlığı (TTKB) “Bilişim Teknolojileri ve Yazılım” dersi öğretim programında 5., 6., 7. ve 8. Sınıflar için “Programlama ve özgün ürün geliştirme” ünitesi içerisinde “Bir problemi çözmek ve projeyi gerçekleştirmek için strateji geliştirebilir, çözüm üretirken farklı bakış açılarını ve yaklaşımları kullanabilir”, “Yazarlık ve programlama dillerini tanıyabilir, en az bir yazarlık/programlama dilini etkili biçimde kullanabilir” kazanımlarına yer verirken (TTKB, 2012), yine MEB 2017 yılında güncellenen ve yazılım içeriklerini dahil ettiği “Bilişim Teknolojileri ve Yazılım” dersi öğretim programında 5. ve 6. Sınıflar için ”Problem Çözme ve Programlama” ünitesi içerisinde “Problemin çözümü için bir algoritma geliştirir”, “Blok tabanlı programlama aracının ara yüzünü ve özelliklerini tanır”, “Blok tabanlı programlama aracında sunulan bir programı verilen ölçütlere göre geliştirerek düzenler” kazanımlarına yer vermiştir (TTKB, 2017). Yine aynı yıl içerisinde MEB ortaöğretim kurumları için hazırlanan “Bilgisayar Bilimi” dersi öğretim programında öğrencilere kazandırılması hedeflenen becerileri; “bilgi-İşlemsel, analitik, eleştirel, algoritmik, matematiksel ve yaratıcı düşünme, problem çözme, etkili iletişim kurma, iş birliği, karar verme, çıkarımda bulunma, algoritma tasarlama, yazılım geliştirme olarak belirlemiştir. Ayrıca bu dersin Fen Liselerinin 9. ve 10. Sınıflarında 2’şer saat zorunlu, hazırlığı olan Anadolu Liselerinde ve Sosyal Bilimler Liselerinde hazırlık sınıfında 4’er saat zorunlu ve diğer tüm liselerin 9., 10., 11. ve 12. Sınıflarında 2’şer saat seçmeli olarak okutulmasına karar verilmiştir (TTKB, 2017).

21. yy bireylerinde bulunması gereken becerileri kazandırmaya katısı olduğu tüm dünyada kabul gören programlama öğretimine yönelik yapılan çalışmalar incelendiğinde yaşanan bazı engeller, problemler ve zorluklar göze çarpmaktadır. Ders içeriğinin öğrenciler tarafından somut olmayan bir yapı olarak görülmesi, karmaşık ve çözümlenemeyen şekilde algılanması öğrencilerin zorlanmasındaki en büyük sebeplerdendir (Esteves & Mendes, 2004; Fujiwara, Fushida, Tamada, Igaki & Yoshida, 2012; Gomes & Mendes, 2007; Jenkins, 2002; Ozoran, Cagiltay & Topalli, 2012; Sivasakthi & Rajendran, 2011). Programlama kavramlarının öğrencilere yabancı dilde sunulması, programlama dilinin içerisinde kendine özgü metotları ve kavramları anlamakta zorlanmaları ve programlamanın doğasında olan iş ve işlemlerin sıralanışı, planlaması ve belirli bir düzen izlenmesi yaşanan zorluklara örnek olabilecek diğer unsurlar olarak söylenebilir (Arabacıoğlu, Bülbül & Filiz, 2007; Bayman & Mayer, 1983; Fesakis & Serafeim, 2009; Hongwarittorn & Krairit, 2010; Ozoran *vd.*, 2012; Pea & Kurland, 1984). Moser (1997) programlama öğretiminin günlük deneyimler için sıkıcı, göz korkutucu ve çok katmanlı bir beceri olduğunu ifade etmiştir. Sivasakthi ve Rajendran (2007) programlama öğreniminin çoklu bilgi ve beceriye ihtiyaç duyulan bir yapıya sahip olduğunu ve ilerleyen dönemlerde bir bilgi mirasına ihtiyaç olduğunu vurgulamışlardır. Ayrıca öğrencilerin cinsiyet farkı, programlama tecrübelerinin olmaması, matematik alanındaki başarısızlıkları ve problem çözme becerilerindeki eksiklikleri farklı öğretim düzeylerinde programlama öğretimindeki başarısızlığa neden olan faktörlerden bazıları olarak göze çarpmaktadır (Ferrer-Mico, Prats-Fernández & Redo-Sanchez, 2012; Fesakis, Gouli & Mavroudi, 2013; Lau & Yuen, 2011; Sullivan & Bers, 2013; Yurdugül & Aşkar, 2013). Bu durum göz önünde bulundurularak bu çalışmada programlama öğretim süreci zamana yayılmış, haftalık kazanımlarla parçalara bölünerek ve farklı programlama öğretim araçları kullanılarak öğrencilerin adım adım programlama tecrübesi edinmeleri sağlanmaya çalışılmıştır.

Öğrencilerin programlama öğrenim sürecinde başarısızlıklarına neden olabilecek etkenlerden birinin de bilgisayar programlamaya karşı kaygı düzeyleri gösterilebilir. Bilgisayarı kullanmaktan korkma, donanımına zarar verme endişesi, bilgisayarla alakalı kişinin daha önce oluşturmuş olduğu olumsuz duygu ve inançlar bilgisayar kaygısına neden olabilir (Chua, Chen & Wong, 1999; Owolabi, Olanipekun & Iwerima, 2018; Sam, Othman & Nordin, 2005). Bilgisayar kaygısı ise; programlama kaygısını tetikleyen unsur olarak görülmektedir. Nitekim bilgisayar temelli bilim dallarının temel ilkelerinin kazandırılması ve programcılığın giriş unsurlarının öğrenciye aktarılması programlama kaygısı nedeniyle öğrencilerde isteksizliğe yol açabilmektedir (Figuroa Jr & Amoloza, 2015). Programlamaya karşı duyulan kaygı, öğrencilerin bu konudaki yeteneklerini yanlış değerlendirmesine,

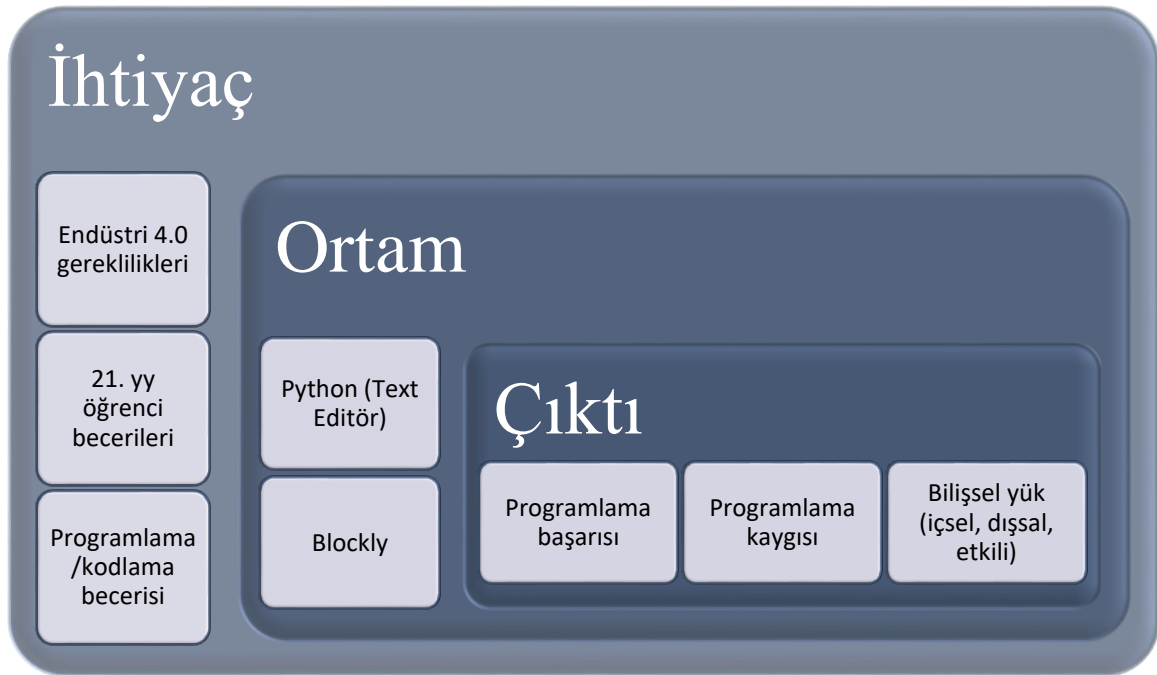


dolayısıyla da bilişsel süreçlerinde bir şema blokajına neden olabilir (Connolly, Murphy, & Moore, 2009). Öğrencilerin programlamaya yönelik kaygılarını azaltabilmek için onları sürece dâhil edecek çeşitli etkinlikler yapılabilir. Bu çalışmada da farklı programlama öğretim araçları kullanılarak bu süreci etkileme durumu incelenmiştir.

Programlama öğrenim sürecinin öğrenciler tarafından sıkıcı ve zor olarak algılandığı öğrenme ortamlarında (Moser, 1997) öğrencilerin bilgi işlem kapasitelerini etkili bir şekilde kullanmaları, çaba sarf etmeleri gerekmektedir. Nitekim öğrencilerin programlama derslerinde başarısız olmalarının ve programlama öğrenim sürecinin zor geçmesinin nedenlerinden biri olarak öğrencilerin sürecin başından sonuna kadarki bilişsel yük düzeylerinin azalmaması gösterilebilir (Kelleher & Pausch, 2005; Stachel vd., 2013; Yousoof, Sapiyan, & Kamaluddin, 2006). Bilişsel yük, öğrenen zihnin, öğretici tarafından verilen görevlerini yerine getirirken, bilişsel yapısına etki eden yük miktarı ya da öğrencinin süreç içerisindeki bir amacı yerine getirirken zihinsel süreçlerindeki yükü temsil eden çok boyutlu bir yapı olarak tanımlanabilir (Kılıç & Karadeniz, 2004; Paas & Van Merriënboer, 1994; Sweller, 1994). Programlama öğretimi süresince öğrencilere verilen kavramların, işlemlerin karmaşık soyut ve çok adımlı oluşu öğrenme ortamındaki öğrencilerin bilişsel olarak yüklenmelerine, programlama öğrenimi sürecinin olumsuz devam etmesine hatta yarıda kalmasına, süreçten kopmalarına neden olabilir. Ayrıca süreç boyunca etki eden bilişsel yük miktarı, yanlış şema edinimi ve ileride ihtiyaç olabilecek şema kullanımını etkileyebilir (Leahy & Sweller, 2004). Öğrenme ortamı ve işlenecek bellek yapısı göz önünde bulundurularak öğrencilerin uzun süreli bellekte kalıcı olarak bilgileri saklayabilmesinin sağlanması, karmaşık ve zor süreçlerden kaçınılması programlama öğretimi sürecini hafifletecek hamlelerden olabilir. Bu noktada öğretim materyalinin kolaylaştırılması ve anlaşılır hale getirilmesi ile bilişsel yüklenme azaltılabilir (Sweller, 2003). Bu çalışmada da görsel ve text tabanlı programlama araçları kullanılarak öğrenme sürecini etkileme durumu incelenmiştir. 10 haftalık öğrenme süreci boyunca öğrencilerin bilişsel yüklenmeleri içsel, dışsal ve etkili bilişsel yük olmak üzere 3 alt boyutta her hafta tespit edilmiş ve süreç sonunda her bir alt boyut hem ayrı ayrı değerlendirilmiş hem de tüm alt boyutların oluşturduğu toplam bilişsel yük hesaplanmıştır.

Programlama öğretim sürecinin öğrenciler tarafından karmaşık ve sıkıcı olarak nitelendirilmesi, text tabanlı programlama araçlarının öğrencilere soyut kavramları daha çok öne sürmesi, programlama öğretiminde yeni yaklaşımların ve öğretim araçlarının ortaya çıkmasına neden olmuştur. Buna bağlı olarak programlama dilinde mevcut olan sözdiziminin görsel ifadeler içeren şekliyle kullanıldığı (Kurihara, Sasaki, Wakita, & Hosobe, 2015;

Weintrop & Wilensky, 2018), görsel öğeleri ön planda olan blok tabanlı programlama araçları geliştirilmiştir (Papavlasopoulou, Giannakos, & Jaccheri, 2019; Prisacari & Danielson, 2017). Günümüzde ilköğretim, ortaöğretim ve hatta lisans öğrenim seviyelerinde görsel programlama araçları yaygın biçimde kullanılmaktadır (Cheng, 2019; Kalelioğlu, 2015; Topalli & Cagiltay, 2018; Tsai, 2019). Alanyazında öğrenci düzeyi ve hedef kazanımlara göre tercih edilen görsel programlama araçlarının farklılık gösterdiği, daha çok Scratch, Alice, Blockly, Microsoft Small Basic, Kodu Game Lab gibi görsel programlama araçlarının kullanıldığı gözlenmiştir (Baltalı, 2016; Culic, Radovici & Vasilescu, 2015; Fowler, 2012; Fowler, Fristce, & MacLauren, 2012; Pérez-Marín, Hijón-Neira, Bacelo, & Pizarro, 2018; Rodger vd., 2010; Sykes, 2007). Bu çalışmada da lise öğrencilerine ileri seviye programlama mantığının kazandırılması için Google tarafından desteklenen ve ücretsiz, düzenlenebilir görsel blok yapısına sahip ve 4 farklı programlama dili desteği sunarak hem görsel tabanlı hem de text tabanlı programlama aracı olarak kullanılabilme özelliğiyle diğer blok tabanlı araçlardan ayrılan Blockly (Bottoni & Ceriani, 2015) temel alınmıştır. MEB tarafından da programlama öğretimi için tercih edilen Blockly görsel programlama aracını kullanma ve kullanmama durumlarına göre ortaöğretim düzeyindeki öğrencilerin programlama öğrenme süreçleri, programlama kaygıları, bilişsel yükleri ve bu sürecin öğrencilere etkileri incelenmiştir. Bu bağlamda çalışmanın kuramsal çerçevesi Şekil 1.'de özetlenmiştir.



Şekil 1. Çalışmanın kuramsal çerçevesi.

Şekil 1’de görüldüğü üzere çalışmanın kuramsal çerçevesini dijital çağda üreten birey

yetiřtirmek için programlama becerisi ihtiyacı, bu beceriyi kazandırmak için farklı programlama araçlarının kullanıldığı öğrenme ortamları ve öğrenme süreci çıktısı olarak programlama başarısı, kaygısı ve bilişsel yük oluşturmaktadır.

### **Arařtırmanın Amacı**

Bu çalışmada; sosyal bilimler lisesi bilgisayar bilimi dersinde görsel programlama aracının kullanıldığı ve kullanılmadığı hazırlık sınıflarında verilen programlama eğitiminin öğrencilerin kaygı düzeylerine, bilişsel yüklerine ve akademik başarılarına etkisi incelenmiştir. Ayrıca arařtırmada öğrencilerin yapılan programlama öğretim sürecindeki programlama kaygılarını, bilişsel yüklerini ve başarılarını etkileyen faktörler ortaya konmuş ve uygulamaya yönelik öğrenci görüşleri belirlenmiştir. Bu bağlamda yanıt aranacak arařtırma soruları řunlardır:

Blok tabanlı görsel programlama öğretim aracının kullanıldığı ve kullanılmadığı öğrenme ortamlarında sosyal bilimler lisesi hazırlık sınıfı öğrencilerinin;

1. Programlamaya yönelik kaygıları arasında anlamlı bir farklılık var mıdır?
2. Bilişsel yükleri arasında anlamlı bir farklılık var mıdır?
  - 2.1. İçsel bilişsel yükleri arasında anlamlı bir fark var mıdır?
  - 2.2. Dışsal bilişsel yükleri arasında anlamlı bir fark var mıdır?
  - 2.3. Etkili bilişsel yükleri arasında anlamlı bir fark var mıdır?
3. Programlama başarıları arasında anlamlı bir farklılık var mıdır?
4. Programlamaya yönelik kaygıları, bilişsel yükleri ve programlama başarıları arasında anlamlı bir ilişki var mıdır?
5. Programlama öğrenim sürecindeki kaygılarını, bilişsel yüklerini ve başarılarını etkileyen faktörler nelerdir?
6. Uygulama sürecine yönelik görüşleri nelerdir?

### **Arařtırmanın Önemi ve Gerekçesi**

Endüstri 4.0 devrimiyle birlikte tüm dünyada ve ülkemizde son yıllarda programlama eğitiminin hızla yaygınlaşması dikkat çekici durumdur. Özellikle 21. yüzyıl öğrenci becerilerinin kazandırılması yönünde programlama eğitiminin avantajları alan yazında da belirtilmekte, dolayısıyla birçok ülke kodlama becerilerinin öneminin farkına varmakta ve programlama eğitimi zorunlu ders olarak öğretim programlarına eklenmektedir. Böylece

içinde bulunduğumuz bilişim çağında tüketim toplumu modelinden üreten birey modeline dönüş ivme kazanmaktadır (Akpınar & Altun, 2014; Gökoğlu, 2017; Sayın & Seferoğlu, 2016; Noone & Mooney, 2018; Pérez-Marín, Hijón-Neira, Bacelo, & Pizarro, 2018; Qian & Clark, 2016).

Ülkemizde de programlama eğitimi özellikle 2012 yılından sonra önem kazanmış ve ortaokul seviyesinden itibaren öğretim programlarına dâhil edilmiştir. Ancak daha çok algoritma mantığının geliştirilmesine yönelik uygulamalar yapılmaya başlanmış ve kapsamlı bir programlama öğretimini içeren çalışmaların henüz yaygın olarak yapılmadığı görülmüştür (Gezgin *vd.*, 2017; Kalelioğlu, 2015; Saygıner, 2017; Topalli & Cagiltay, 2018). Ayrıca ortaokuldan sonra ortaöğretim seviyesinde programlama öğretimine yönelik herhangi bir dersin yer almaması kazanılan programlama becerisinin gerçek hayattaki farklı durumlara uyarlanmasına ve kalıcı bilgiye dönüşmesine engel olabilecek bir durumdur. MEB'in 2016 ve 2017 yıllarında aldığı kararlarla hem ortaokullarda Bilişim Teknolojileri ve Yazılımı ders içeriği güncellenerek programlamaya yönelik kazanımlar zenginleştirilmiş hem de ortaöğretim seviyesinde Bilgisayar Bilimi dersi eklenerek programlamaya yönelik kazanımlar ders kapsamına alınmıştır. Dersin 2017-2018 eğitim-öğretim yılının başında ortaöğretim programına eklenmesi sebebiyle alan yazında bu konuda bir çalışmaya rastlanmamıştır. Bu açılarından ele alındığında çalışmanın; ortaöğretim seviyesinde olması ve ortaöğretim kurumlarında, alana özgü meslek liseleri dışında, ilk defa genel lise tarzında liselerde zorunlu ders olarak okutulacak bir müfredat üzerinde durması nedeniyle sonraki yıllarda yapılacak çalışmalara rehber niteliğinde olacağı düşünülmektedir.

Çalışmanın yapılacağı genel lise türünün sosyal bilimler lisesi olmasının ve programlama eğitiminin sadece hazırlık sınıflarında haftada dört saat verilmesinin öğrenciler üzerinde programlamaya yönelik kaygı oluşturup oluşturmadığını incelemenin MEB'in öğretim programını iyileştirme sürecine katkıda bulunması açısından da bu çalışma önemli görülmektedir. Nitekim ülkemizde genel lise seviyesinde ilk defa 2017-2018 eğitim öğretim yılında bir programlama dersinin öğretim müfredatına girmesi nedeniyle bu yönde daha önce herhangi bir araştırma yapılmamasının bu durum üzerinde önemli bir etkisinin olduğu söylenebilir. Ayrıca öğretimin yapılacağı lise türünün sosyal bilimler lisesi olmasının, öğrencilerin programlamaya olan bakış açılarını etkileyip etkilemediğini saptamak açısından kaygının varlığının araştırılması, programlama kaygısına neden olan faktörlerin ortaya konulmasının da alanyazına katkı sağlayacağı düşünülmektedir.

Ortaokul seviyesinde sadece görsel tabanlı ortamlarda yapılan, algoritma ve temel programlama eğitimi sonrası, ortaöğretimde metin tabanlı bir ortam ve görsel tabanlı

programlama aracıyla gerçek bir programlama dili kullanılarak farklı iki grupta yapılacak eğitiminin bilişsel yük açısından değerlendirilecek olması da alan yazına katkı açısından bir zenginlik olarak görülmektedir. Bunun yanı sıra incelenen bilişsel yük seviyesinin 3 alt boyutta içsel, dışsal ve etkili bilişsel yük olarak incelenmesi, öğrenme içeriği, öğretim materyali/aracı, öğrenme ortamı gibi farklı etkenlerden doğabilecek öğrencilerin bilişsel yüklenme seviyelerini azaltma sürecine getirilecek çözüm önerisinin tespiti açısından önemli görülmektedir. Yine öğrencilerin bilişsel yüklenme durumlarının sadece bir defa değil her hafta, öğretim etkinliğinden sonra ölçülmesiyle konu ve etkinlik bazında bilişsel yüklenmeye etkisinin tespiti de amaçlanmaktadır. Ayrıca bu öğretim seviyesinde yapılan programlama öğretimi çerçevesinde bilişsel yük değişkeninin incelenmesinin, alanyazında az sayıda var olan çalışmalara güç katacağı düşünülmektedir (Çakiroğlu vd., 2018; Hromkovič, Serafini, & Staub, 2017; Lister, 2011; Salleh, Shukur, & Judi, 2018).

Alan yazında çok değinilen programlama eğitiminde başarılı olup olmama durumunu etkileyen faktörlere (Baz, 2018; Erol & Kurt, 2017; Korkmaz, 2016; Olelewe & Agomuo, 2016; Owolabi, Adedayo, & Olayanju, 2018; Sullivan & Bers, 2013) ilk defa müfredata eklenen bir ders açısından ve yine ülkemizde ilk defa yapılacak bir öğretim seviyesinden bakılacaktır. Öğrencilerin uygulama süreçlerinde yaşadıkları deneyimlere dayalı olarak ortaya çıkacak sonuçlarla alanda mevcut seviyede ileride yaşanması muhtemel olumsuz durumların önüne geçilebileceği öngörülmektedir.

### **Sınırlılıklar**

Bu çalışma aşağıdaki sınırlılıkları içermektedir:

1. Uygulama yapılan çalışma grubu hazırlık sınıfındaki 90 öğrenciyle sınırlıdır.
2. Çalışma Bilgisayar Bilimi dersini alan ortaöğretim seviyedeki iki okul türünden sadece Sosyal Bilimler Lisesi'nde yapılmıştır.
3. Çalışma grubu öğrencilerinin bazılarının programlama konusunda hiçbir tecrübesi bulunmamaktadır.
4. Çalışma süresi programlama öğretimine yönelik belirlenen hedef kazanımları gerçekleştirmek amacıyla haftada 4 saat olmak üzere 10 hafta sürmüştür.
5. Araştırmacı aynı zamanda dersi veren öğretici rolünde olduğundan uygulama sürecine yönelik öğrenci görüşlerini belirlemek amacıyla yapılan yüz yüze görüşmeler öğrencilerin doğru ve samimi düşüncelerini ifade etme durumlarıyla sınırlıdır.

6. Çalışmada elde edilen sonuçlar yalnızca görsel ve text tabanlı programlama öğretimi yapılan grupların akademik başarılarını, bilişsel yüklerini, kaygılarını, bu boyutları etkileyen faktörleri ve uygulamaya yönelik öğrenci görüşlerini yansıtmaktadır.
7. Uygulama sırasında okulun bulunduğu bölgede elektrik problemlerinin oluşması nedeniyle teknik aksaklıklar meydana gelmiştir.

## **Varsayımlar**

Çalışmada;

1. Görsel ve text tabanlı programlama öğretimi yapılan gruptaki öğrencilerin benzer özelliklere sahip oldukları,
2. Uygulamaya katılan her iki grup öğrencilerinin de istekli ve gönüllü oldukları,
3. Uygulanan veri toplama araçlarına öğrencilerin verdikleri cevapların gerçek düşünceleri olduğu ve rastgele cevaplandırmadıkları,
4. Görüşme yapılan öğrencilerin görüşme sorunlarını hiçbir baskı altında olmadan cevapladıkları ve görüşlerini rahatça beyan ettikleri,
5. Uzman görüşlerine de başvuru hem nicel hem nitel verilerin analizlerinin tarafsız ve doğru şekilde yapıldığı varsayılmıştır.

## **Terim ve Tanımlar**

**Programlama:** Makinaların insanlar tarafından verilen bir görevi, bir problemin çözümünü gerçekleştirebilmesi için çeşitli komutların bir araya gelerek sonuca ulaştırılması sürecidir (Balanskat & Engelhardt, 2014; Kesici & Kocabaş, 2007).

**Blok Tabanlı Görsel Programlama:** Programlama dilinde mevcut olan sözdiziminin, şekli, rengi ve nitelikleri açısından farklı kategorilerde olan bloklar şeklinde görselleştirildiği, bir fare yardımıyla bloklardan oluşan komutların bir araya getirilerek işlerlik kazandırıldığı, sonucunda kullanıcıya geri bildirim desteği sağlayan araçlarla yapılan programlama türüdür (Burnett & McIntyre, 1995; Çatlak vd., 2015; Weintrop & Wilensky, 2015).

**Programlama Kaygısı:** Öğrencilerin programlama becerilerini yanlış değerlendirmesiyle ortaya çıkan, kişinin programlamaya karşı önyargılarıyla oluşan çekirdek inançların büyüyen kalıplaşmış düşüncelere dönüşmesine yol açan duygudur (Connolly, Murphy, & Moore, 2007; Özmen & Altun, 2014; Rogerson & Scott, 2010).

**Bilişsel Yük:** Öğrencilerin öğretici tarafından kendilerine verilen bir görevi yerine getirirken, bilgi yapısı ve bu bilgiyi işlemesine izin veren mimari yapının kapasitesinin

sınırlarının aşılmasından dolayı öğrenciyi zihinsel olarak zorlayan yüküdür (Kılıç & Karadeniz, 2004; Paas, Renkl, & Sweller, 2003)

**İçsel Bilişsel Yük:** Anlaşılması gereken bir bilginin ya da materyalin, doğal karmaşıklığı ile ilgili olan, bilginin sunumu ya da yapılan faaliyetlerle alakalı olmayıp öğretici tarafından doğrudan kontrol altına alınamayan bellekte oluşan yük miktarıdır (Sweller, 2010).

**Dışsal Bilişsel Yük:** Öğretim tasarımı sonrası oluşan materyalin iç karmaşıklığıyla ilgili olarak meydana gelen ve öğretim içeriğiyle bağlantısı olmayan, tasarlanmış teknikler ve oluşturulacak öğretim prosedürleri ile öğreticinin yüklenmeye müdahale edebildiği bilişsel yük toplamına etki eden zihinsel yüklenmedir (Sweller, 2010).

**Etkili Bilişsel Yük:** İçsel bilişsel yük gibi sadece öğrencinin kendisiyle ya da dışsal bilişsel yük gibi öğretim tasarımı ve materyal ile ilgili olmayan süreç içerisinde yapılacak öğrenme etkinlikleri ile ilgilenen ve kontrol altına alınabilen bilişsel yüklenmedir (Sweller, 2010).

**Google Blockly:** Farklı renkteki ve işlevdeki bloklardan yararlanılarak ve sürüklenip bırak tekniği kullanılarak JavaScript, Python ve XML programlama dillerinde oluşturulan programların dışa aktarılmasını sağlayan açık kaynak kodlu görsel programlama aracıdır.

## İKİNCİ BÖLÜM

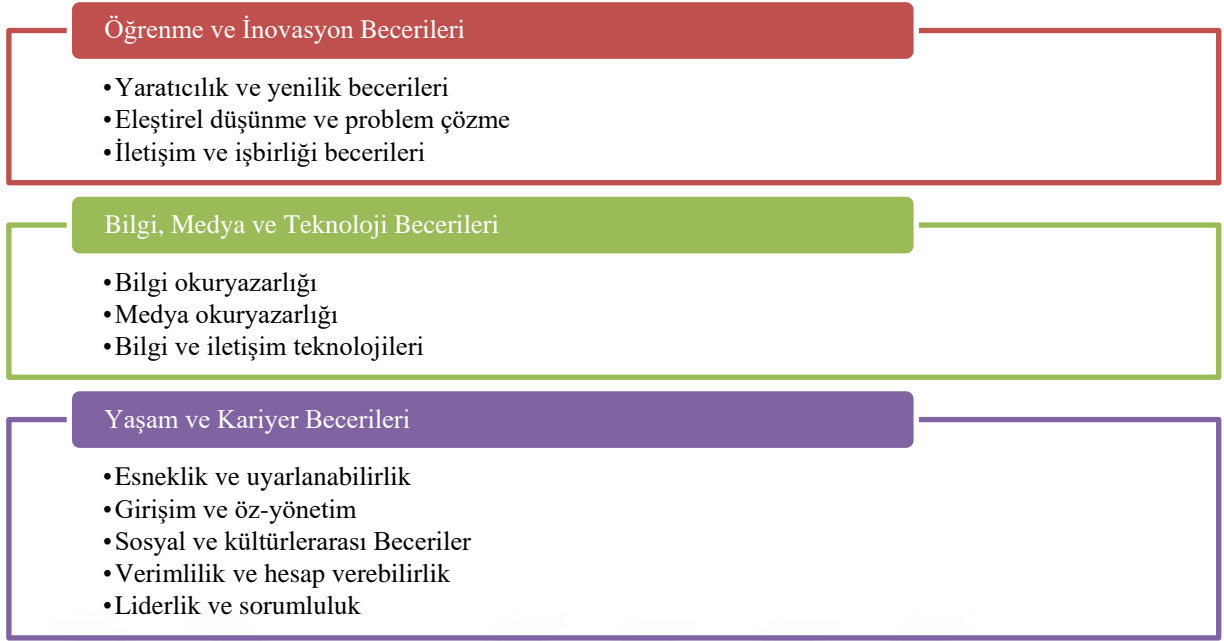
### Kuramsal Çerçeve ve İlgili Araştırmalar

#### Programlama Öğretimi

Ekonomik İşbirliği ve Kalkınma Örgütü (The Organization for Economic Co-operation and Development [OECD])’nün , “Gelecek Üretim Devrimi” olarak tanımladığı Endüstri 4.0 kavramı, insan – makine entegrasyonunun yapay zeka, bulut teknolojileri ve nesnelerin interneti gibi birçok yeni teknolojinin bir arada kullanıldığı üretim ortamı olarak tanımlanmaktadır (“Industry 4.0: A View from the OECD”, 2018; OECD, 2017). Bu nedenle birçok sektörde BT etkin bir şekilde kullanılmaya başlanmıştır ve bu alanlarda istihdam edilen bireylerde aranan özellikler önceki sanayi devrimlerine göre değişim göstermeye başlamıştır (Fırat & Fırat, 2017). Bu bağlamda Amerikan Okul Kütüphanecileri Derneği (American Association of School Librarians [AASL]), Uluslararası Eğitim Teknolojileri Topluluğu (International Society for Technology in Education [ISTE]) gibi önde gelen birçok uluslararası oluşum 21. yüzyıl öğrenci becerileri olarak ifade edilen birey özelliklerini çeşitli başlıklar altında tanımlamışlardır (AASL, 2018; Ekici, Abide, Canbolat & Öztürk, 2017; ISTE, 2016; Partnership21, 2018). ISTE 1993 yılından itibaren eğitim alanına teknolojinin entegre edilmesiyle ilgili çeşitli standartlar önermiştir. Öğrenciler için önerilen standartlar arasında donanımlı, bilgi üreten, yenilikçi, tasarımcı, bilgi-işlemsel düşünebilen, yaratıcı ve küresel işbirliğine açık olan dijital vatandaş özelliklerine sahip olma gibi üst düzey düşünme becerileri öne çıkmaktadır (ISTE, 2016). AASL (2018) ise; 21. yüzyıl öğrenci becerilerini “düşünme”, “oluşturma”, “paylaşma” ve “üretme” olmak üzere 4 temel etki alanı üzerine şekillendirmiştir. Birey özelliklerini ise “soruşturan”, “kapsayıcı”, “işbirlikçi”, “yeniden şekillendirebilen”, “keşfeden” ve “ilişki kurabilen” bireyler olarak tanımlamıştır.

21. yüzyıl becerileri, ABD, Singapur, Avustralya ve Finlandiya dâhil olmak üzere birçok gelişmiş ülkede eğitim reformlarının ortak noktası olmuştur (Eguchi, 2014). 21. yüzyıl öğrenci becerileri üzerine çalışan, bünyesinde BT, otomotiv, grafik ve sinema sektörlerinin lider kuruluşlarını barındıran, uluslararası oluşumlardan biri de Partnership21 organizasyonudur. Alan yazın incelendiğinde 21. yüzyıl becerilerinin farklı şekillerde kategorize edildiği görülmesine rağmen en kapsamlı sınıflamanın Partnership21 (2018) tarafından üç ana başlık altında tanımlandığı Şekil 2’te görülmektedir.





Şekil 2. 21. yüzyıl becerileri (Partnership21, 2018).

Partnership21 (2018) genelde algoritma, özelde kodlama becerisini de üst düzey düşünme becerilerinin kazandırılmasına yardımcı olan bir beceri olarak 21. yüzyıl öğrenci becerileri arasında göstermiş, tüm bu becerilerin gelişiminde bilişim teknolojilerinin önemli bir araç olduğunu dile getirmiştir. Bu bağlamda pek çok ülke öğretim müfredatlarını revize ederek bilişim teknolojileri ve yazılım üzerine dersler eklemişler, kodlama becerilerinin kazanımını sağlayacak şekilde öğretim planlarını hayata geçirmişlerdir (Grover & Pea, 2013). Problem çözme becerilerinin tüm alanlarda önemli bir faktör olduğu düşünüldüğünde, programlama öğretiminin özelde değil aynı zamanda genelde de önemli bir konuma sahip olduğu söylenebilir (Saeli, Perrenet, Jochems, & Zwaneveld, 2011). Programlama sektöründe var olan yetişmiş insan kaynağı, sadece alana özgü olmamakla birlikte tüm sektörlerin üst düzey düşünme becerilerine sahip birey ihtiyacının karşılanmasına yardım etmektedir (Demirer & Sak, 2016).

### **Programlama dilleri.**

Programlama, mevcut bir problem durumunda, seçilen programlama dilinin komutları bir araya getirilerek, derlenerek çalıştırılması ya da bir problem durumunun çözüm yolunun bilgisayara, onun anlayacağı dilde kodlanması ve hangi durumda hangi tepkiyi vermesi gerektiğinin öğretilmesi işidir (Arabacıoğlu, Bülbül, & Filiz, 2007; Çölkesen, 2002). Renumol, Jayaprakash ve Janakiram (2009) ise programlamayı farklı programlama dilleri kullanılarak bilgisayar programlarını yazma, test etme ve hata ayıklama işlemi olarak tanımlamışlardır. Diğer bir ifadeyle programlama; verilen bir problemin çözümü için sarf

edilen çabanın soyut halinden somut hale dönüşümü olarak nitelendirilebilir (Lye & Koh, 2014).

Programlama işlemi bazı temel aşamalardan oluşmaktadır. Beş temel basamaktan oluşan bu aşamalar problemin belirlenmesi, çözüm yollarının belirlenmesi, kodlama, yorumlama ve derleme, hataların giderilmesi olarak sıralanabilir (Brown & Wilson, 2018; Kesici & Kocabaş, 2007).

- **Problemin belirlenmesi:** Ortada bir problemin var olduğunun fark edilmesi, problemin derinlemesine incelenmesi ve etken olan durumların neler olduğunun belirlenmesi gerekir.
- **Çözüm yollarının belirlenmesi:** Problemin yok edilmesi için tüm şartlar düşünülerek bir çözüm bulunması, çözümün etkili olduğu emin olunduktan sonra bir algoritma ve akış şemasının oluşturulması gerekir. Algoritma ve akış şemasının tasarımcıya ait olma zorunluluğu olmadığı gibi daha önce kullanılan bir tasarı da değerlendirilebilir.
- **Kodlama:** Algoritma oluşumundan sonra kullanılacak olan programlama dilinin belirlenerek, bu dile uygun sözdizimi ve anlambilimi çerçevesinde programın yazılması gerekir.
- **Yorumlama ve derleme:** Derleme, yazılan kodların makine diline çevrilmesi olarak tanımlanabilir. Derleme işlemi başarıyla bittikten sonra program çalıştırılmış olacaktır.
- **Hataların giderilmesi:** Program çalışmaya başladıktan sonra ortaya çıkan yazım ve mantık hatalarının giderildiği kısımdır. Tüm hatalar giderildikten sonra derleme işlemi tekrarlanır.

Programlama işleminin yapılabilmesi için çeşitli programlama dilleri (C/C#/C++, Java/JavaScript, PHP, Python, Visual basic vs.) geliştirilmiştir. Kullanılan programlama dillerinin ortak noktası; var olan bir problemi çözmek olsa da her dilin sahip olduğu bir sözdizimi ve anlam bilgisi vardır. Bireylerin programlama dili seçimi ise gerek popülariteye göre gerekse kullanım amacına göre değişmektedir. Online bir kod deposu olan GitHub 2017 yılında dünya üzerinde en çok kullanılarak popüler olan programlama dillerini açıklamıştır. Bunlar arasında %25'lik dilimde Python ilk sırada, Java ise ikinci sırada yer almaktadır. Türkiye'de ise Python, Java ve Javascript'ten sonra üçüncü sırada en çok kullanılan programlama dilidir (Tekin, 2017).

Yorumlanmış, etkileşimli ve nesne yönelimli bir programlama dili olan Python; 1990 yılında Guido Van Rossum tarafından tasarlanmıştır ve diğer birçok programlama dili gibi ticari amaçlar da dâhil olmak üzere ücretsiz ve pratik bir şekilde herhangi bir modern bilgisayarda çalıştırılabilmektedir (Dalcin, Paz, Storti, & D'Elia, 2008; Sanner, 1999).

Tasarım aşamasında okunaklı bir dil olması amacıyla geliştirilen Python; diğer kullanımı yaygın programlama dillerine göre daha az sayıda sözdizimsel istisnai ve özel durumu bulunan, oldukça basit ve zarif bir sözdizimine sahip olmasına rağmen birçok kullanıcıya göre yine de güçlü bir dildir (Van Rossum & Fred L. Drake, Jr. (2012). Ayrıca Python'ın standart sürümü C dili ile yazılmış olan kütüphanesi tüm kullanıcılar tarafından herhangi bir lisans istenmeden indirilebilmektedir. Derlenen bir Python programının platform ayırt etmeden her ortamda çalışması da Python programlama dilinin bir diğer avantajı olarak söylenebilir (Cock vd., 2009; Luh, 1994; Van Rossum, 1999). Bu çalışmada da ortaöğretim düzeyinde ve acemi programcılara yönelik kullanım kolaylığı da dikkate alınarak Python programlama dili tercih edilmiştir.

### **Programlama eğitiminde yaşanan güçlükler.**

Herhangi bir programlama dilini kullanmaya yeni başlayan programcılar kodlama becerisini kazanırken sadece bir problemi çözmek değil, aynı zamanda bir programlama dilinin sözdizimini ve anlambilimini, çözümün bilgisayar tarafından anlaşılacak biçimde nasıl ifade edileceğini öğrenmek zorundadırlar (Mannila, Peltomäki, & Salakoski, 2006). Programlama kavramlarını, işleyişlerini bilmek ve bilgisayara özgü bir dille bütünleştirip bir algoritma oluşturmak programlama sürecinin temel aşamalarıdır (Mannila, Peltomäki, & Salakoski, 2006). Bu nedenle alan yazında kodlama becerisi edinmenin zor bir görev olduğu konusunda genel bir yargı bulunmaktadır (Tuomi, Multisilta, Saarikoski, & Suominen, 2018; Webber & Possamai, 2009). Gomes ve Mendes (2007) de programlama öğretiminde karşılaşılan güçlükleri; öğretim yöntemleri, çalışma metotları, öğrencilerin yetenekleri ve tutumları, programlamanın doğasından kaynaklı güçlükler ve psikolojik etkiler olarak 5 başlık altında detaylandırmışlardır.

- **Öğretim Yöntemleri:** Dinamik kavramların statik malzemelerle öğretilmesi öğrencilerde soyut-somut geçişini sağlayamamakta, öğretmenlerin, bir programlama dili kullanarak problem çözmeyi teşvik etmek yerine, bir programlama dili ve sözdizimsel ayrıntılarını öğretmeye daha fazla odaklanmaları.
- **Çalışma Metodları:** Öğrencilerin yanlış çalışma metodolojileri kullanması.
- **Öğrencilerin Yetenekleri ve Tutumları:** Birçok öğrencinin yeterli matematiksel ve mantıksal bilgiye sahip olmamaları.
- **Programlamanın Doğası:** Programlamanın yüksek düzeyde soyut içerikli olması ve programlama dillerinin çok karmaşık bir sözdizimine sahip olması
- **Psikolojik Etkiler:** Öğrencilerin motivasyonunun eksik olması ve öğrencilerin programlamayı yaşamlarının zor bir döneminde öğreniyor olması.

Öğrencilerin karşılaşılabileceği güçlüklerden bir diğeri ise prosedürel programlama dilinden nesneye yönelik programlamaya geçişte sorun yaşamalarıdır. Giriş seviyesinde karmaşık yapıya sahip programlama dillerinin öğrenimi önerilmemektedir (Kölling, 1999; Mazaitis, 1993). Ginat (2006) algoritma öğretimiyle ilgili çalışmasında problem çözme becerilerinin kazanımı konusunda öğrencilerin güçlükler yaşadığını, çözümü için problemi her açıdan ele alamadıklarını, yerel ve sınırlı bakış açısına sahip olduklarını ortaya koymuştur. Katai ve Toth (2010) özellikle tekrar gerektiren (döngüler ve listeler) ve iç içe karmaşık yapıların (işlemler ve karşılaştırmalar) çok fazla soyutluk içerdiğinden programlamaya yeni başlayan acemi programcıların bu konularla sık sık yüzleşmek zorunda kaldıklarını ve zorlandıklarını dile getirmişlerdir. Byrne ve Lyons (2001) sorunu geleneksel programlama dillerinde olan genel kuralların öğrenciler tarafından ezberlenmesi olarak değerlendirmiştir. Lahtinen, Ala-Mutka ve Jarvinen (2005) çalışmalarında, öğrencilerin programlamanın yapısı nedeniyle kavramalarının zor olduğunu, program tasarımı yapmakta sıkıntı yaşadıklarını ortaya koymuşlardır. Ayrıca öğrencilerin programlamaya yönelik olumsuz bir tutuma sahip olması, programlamaya dair önceki deneyimlerinden ya da bilgi eksikliğinden dolayı kafalarında programlama kaygısı oluşturmaları, program oluştururken kavramların yabancı bir dille ifade ediliyor olması, süreci öğrenciler için karmaşık hale getirebilmektedir (Connolly, Murphy & Moore, 2007; Esteves & Mendes, 2004; Ozoran, Cagiltay, & Topalli, 2012).

Tüm bu problemler nedeniyle eğitimciler programlama öğretimi için yeni arayışlara gitmişler ve soyut bilgiyi somutlaştırmak, karmaşık sözdizimlerini adım adım gösterebilecek, text tabanlı programlama ortamlarıyla aynı çıktıları verebilecek, tasarımı sayesinde programlama dilinin sözdizimsel yapısını kolaylaştırabilen nesne tabanlı görsel programlama araçlarını keşfetmişlerdir (Erol, 2015).

### **Görsel Programlama Öğretim Araçları**

Programlama öğretimi sürecinde yaşanan zorluklar, acemi programcıların süreçten erkenden kopmak istemeleri görsel programlama araçlarının ortaya çıkmasını gündeme getirmiştir. 1960 yılında Logo firması bir kaplumbağanın sağa, sola yukarı ve aşağıya hareket ettirilmesiyle başlayan bir görsel ortam oluşturmuştur. Daha çok çocuklara yönelik olarak tasarlanan Logo birçok görsel programlama aracının geliştirilmesine ilham olmuştur (Kafai, Fields, Roque, Burke, & Monroy-Hernandez, 2012).

Görsel programlama, sözdizimi ve anlambilimi kullanıcılara iletmek için birden fazla boyutun kullanıldığı programlama türüdür (Burnett & McIntyre, 1995; Tsai, 2019). GPA, metinsel programlama dillerinin aksine, yeni başlayan acemi programcılara programlamayı

öğretmek ya da mühendislik gibi farklı alanlarda yeni modeller geliştirmek için tasarlanmıştır. Görsel programlama araçlarının temelde üç ana hedefi vardır. Bunlar; programlamayı belirli uygulayıcılara ulaştırabilmek, öğrencilerin programlama becerilerini geliştirmek ve programlama görevini tamamlamalarını sağlamaktır (Burnett & McIntyre, 1995). Bu üç amaca hizmet etmek için geliştirilmiş birçok blok tabanlı programlama aracı mevcuttur. Bloklara dayalı programlama araçları yap-boz parçaları mantığıyla çalışan görsel programlama ortamlarıdır. Görsel programlama ortamlarında parçalar sadece birbirini tamamlamak üzere birleştirilmez. Bir program yapısında bütünü birleştirmek, bir algoritmayı tamamlamak asıl amaçtır (Idrees, Aslam, Shahzad, & Sarwar, 2018). Bu ortamlarda öğrenciler sadece bir fare yardımıyla komutları bir araya getirerek, işleyen programlar inşa edebilmektedirler. Oluşturulan programların geçerli olup olmadığına dair geri bildirim verilmektedir. Böylelikle programlama sürecinin altyapısını oluşturan algoritmik sözdizimi hatalarını bulmayı hızlandırmaktadır. Genellikle kategoriler halinde verilen her bloğun şekli, rengi ve niteliklerini belirten ipuçları ortam içerisinde kullanım kolaylığı sunulmaktadır. Çalışma alanında benzer yapıların gruplarını içeren kategoriler ve programların yerleştirildiği yürütme alanı bulunmaktadır. Bu sayede öğrenciler geliştirdikleri algoritma çıktılarını hareketli görseller aracılığıyla görebilmektedirler (Weintrop & Wilensky, 2015; Basawapatna, 2016).

Görsel programlama araçları taşıdığı özelliklerle, bir programlama dilinin sözdizimsel ayrıntıları hakkında programcıların endişe duymadan, görsel öğeleri basit sürükle-bırak yöntemini kullanarak karmaşık yapıdaki programları hızlı bir şekilde geliştirmelerine yardımcı olabilir (Sáez-López, Román-González, & Vázquez-Cano, 2016). Görsel programlama araçlarının (GPA) kullanıldığı öğrenme ortamı yeni bir programlama dili öğrenmede yaşanan zorlukları etkili bir şekilde azaltabileceği gibi öğrencilerin öğrenme hedefine karşı ilgi ve motivasyonunu artırabilir (Basawapatna, 2016; Chao, 2016; Yukselturk & Altıok, 2017). Tasarım tabanlı öğrenme imkânı da sunan GPA, öğrencilerin pratik uygulamalar geliştirebilecekleri ve aktif öğrenme fırsatı bulabilecekleri bir ortam sağlar (Robertson & Howells, 2008). Son yıllarda görsel programlama araçlarında önemli gelişmeler olmuş 2011 yılından sonra 15'ten fazla alana özgü görsel programlama aracı tasarlanmıştır. Idrees, Aslam, Shahzad ve Sarwar (2018)'in özellikle blok tabanlı programlama dillerini ve gelişimlerini iki kuşak şeklinde ortaya koyduğu Tablo 1'de görülmektedir.

Tablo 1. Görsel Programlama Ortamları

2. Jenerasyon Görsel Programlama Araçları		
2005-2014	Iconic Programmer	Temel Programlama Eğitimi
	B#	
	StarLogo TNG	
	Progranimate	
	devFlowcharter	
	App Inventor for Android	
	Flowcharts Interpreter	
2005-2014	Flowgorithm	Çocuk Oyunları, Animasyonlar
	Scratch	
	Snap	
	Stencyl	
	CODE	
2011	Touch Develop	Mobil
2013	Pencil Code	Eğitim
2013	Dynamo	3D Modelleme
2015	Beetle Blocks	3 Boyut
2006-2015	Microsoft VPL	Robotik
	Lego Mindstorms Software	
	Open Roberta	
	MBlock	
2015-2017	VIPLE	Programlama Eğitimi
	Blockly	
	BlockPy	

Tablo 1’de listelenen görsel programlama araçlarına yönelik alanyazında çeşitli çalışmalar mevcuttur. Bu çalışmalara bakıldığında diğer programlama araçlarına göre daha çok tercih edilen GPA’nın Scratch, Alice ve Code.org olduğu göze çarpmaktadır (Benzer & Erümit, 2017; Chang, 2014; Çatlak *vd.*, 2015; Çakiroğlu *vd.*, 2018; Ferrer-Mico *vd.*, 2012; Kalelioğlu, 2015; Noone & Mooney, 2018; Park *vd.*, 2015; Sáez-López *vd.*, 2016; Solmaz, 2014; Sykes, 2007). Bu araçların en temel ortak özelliği görsel bloklar kullanılarak algoritmaların geliştirilmesi ve sonucun ekrana animasyonla yansımalarıdır. Devam eden bölümde bu üç GPA hakkında bilgi sunulmuştur.

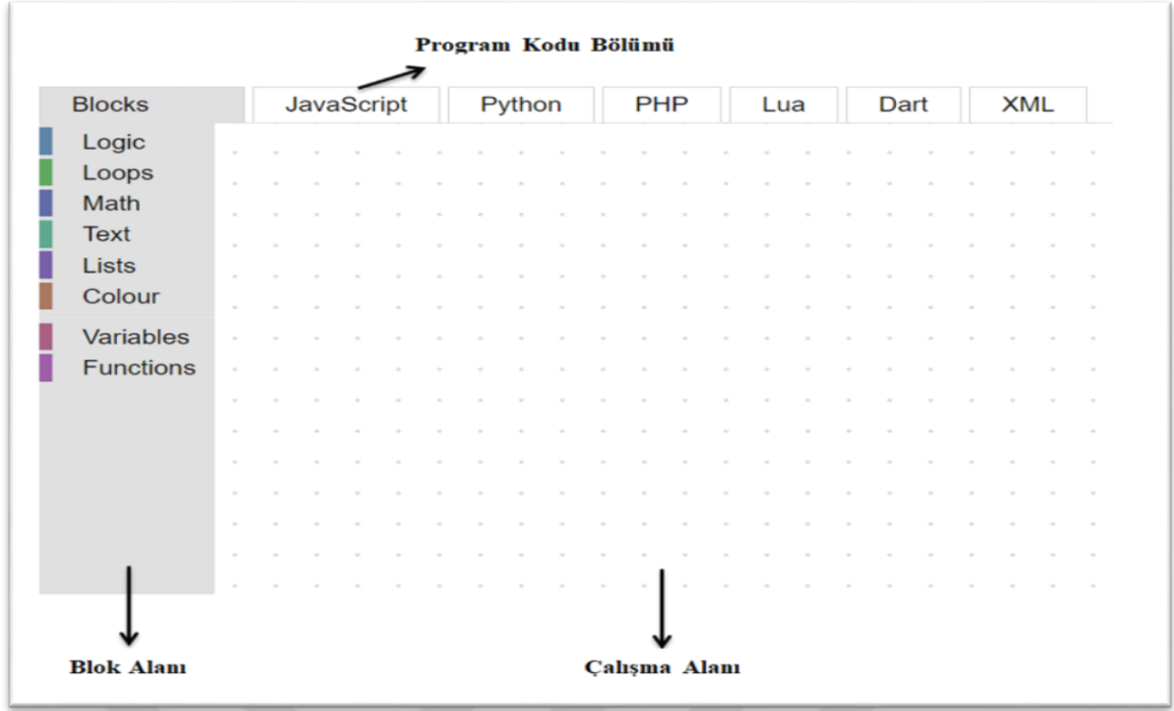
**Scratch;** 2003 yılında MIT Medya Laboratuvarı'nda Lifelong Kindergarten grubu tarafından geliştirilen, uzun kod yapıları yerine Lego stili bloklarla sürükle-bırak yaklaşımıyla öğrencilere özellikle algoritma mantığının yerleştirilmesinde kullanılmaktadır. Scratch aynı zamanda topluluklar tarafından programlanan projelerin çevrimiçi olarak yayınlandığı 40'ın üzerinde dil desteği sunarak ve ücretsiz bir web sitesi hizmeti vererek oyunlar, animasyonlar ve sunular gibi farklı sonuçlar elde edilebilen bir platformdur (Gezgin, Özcan, Ergün, Özge & Neriman, 2017; Kert & Uğraş, 2009; Malan & Leitner, 2007; Meerbaum-Salant, Armoni & Ben-Ari, 2011; J. Moreno & Robles, 2014).

**Alice;** Cornegie Mellon Üniversitesi'nde tasarlanmış, 3 boyutlu yapısıyla programlama öğretimi için kullanılan görsel programlama araçlarından biridir. Alice'i diğer programlama ortamlarından ayıran unsurlardan biri 3 boyutlu bir geri bildirim ortamına sahip olmasıdır. Kod bloklarının olduğu bölümden alınan yapılar, blokların birleştirileceği çalışma alanına sürükle-bırak yöntemiyle aktararak programlar oluşturulur. Eğlenceli yapısı sayesinde Alice acemi programcıların ilgisini çekebilmektedir (Al-Tahat, 2014; Mullins, Whitfield & Conlon, 2009; Stenerson, 2012).

**Code.org;** kod bloklarının sürükle-bırak yöntemiyle kod alanına aktarıldığı, temel seviyeden ileri seviyeye doğru kodlama ile yeni tanışan öğrencileri yönlendiren ve çevrimiçi olarak kullanılabilen blok tabanlı bir kodlama platformudur. 2013 yılında faaliyete geçen Code.org 34 dilde kullanıcılara kodlama imkânı sunmaktadır. Google, Facebook ve Microsoft gibi BT alanında faaliyet gösteren dev firmalar bu platforma destek vermektedir. Code.org portalında öğrencilere algoritma mantığı, koşul ifadeleri, problem çözme becerileri gibi kodlama eğitiminin temelini oluşturan kazanımlar oyunlar eşliğinde kazandırılmaya çalışılmaktadır. Ayrıca ücretsiz üyelik sistemi ile çalışması ve Türkçe dil desteği sağlaması öğrencilerin dünya ile eş zamanlı olarak kodlama dünyasına adım atmasına fırsat vermektedir (Code.org, 2018; Kalelioğlu, 2015).

### **Blok tabanlı görsel programlama aracı olarak Blockly.**

Blockly, Google firması tarafından 2012 yılında piyasaya sürülmüş ve 2017'den itibaren aktif gelişime açılmış olan blok tabanlı görsel bir programlama aracıdır. Blockly ilham kaynağı daha önce MIT'de yapılan çalışmalardır. Temelinde Javascript Kütüphanesi kullanılır. Diğer blok tabanlı programlama araçları gibi Blockly de programlama dillerinin karmaşık sıkıcı ortamından öğrencileri kurtarmak hedefiyle tasarlanmıştır (Fraser, 2015; Pasternak, Fenichel & Marshall, 2017). Şekil 3'te arayüzü verilen Blockly aracını diğer GPA'lardan ayıran en büyük fark blok yapılarıyla animasyonları değil, programlama öğelerini ön plana çıkarmasıdır.



Şekil 3. Blockly arayüzü.

Şekil 3'te görülen Google Blockly programı arayüzü temel olarak üç bölümden oluşmaktadır. Kod bloklarının yer aldığı “Blok Alanı”nda her bir içeriğe ait olan kod blokları farklı renklerde ifade edilmektedir. Blok alanında verilen bir değişken tüm kod bloklarında tanımlandığından kullanıcılar sadece sürükle-bırak yöntemiyle çalışma alanında blokları birleştirerek programı çalışır hale getirebilirler. “Çalışma Alanı” ise kod bloklarının bir bütün haline gelerek programın oluşmasını sağlayan bölümdür. Çalışma alanında hatalı olarak birleştirilmeye çalışılan bloklar program tarafından engellenerek kullanıcı hatalarının en aza indirilmesi hedeflenir. “Program Kod Bölümü” ise çalışma alanında, bloklar birleştirilerek oluşturulan kodun text formatının kullanıcıya sunulduğu bölümdür. Blockly 6 farklı programlama dilinde kodların text halini kullanıcıya sunarak çalışılan programlama dilinin sözdizimsel yapısının anlaşılmasını da sağlamaktadır (Fraser, 2015; Pasternak, Fenichel & Marshall, 2017). Blockly kolay kullanımı ve renkli arayüzü sayesinde ileri bir programlama dili ile çalışmak isteyenlerin de kullanabileceği gelişmiş özellikleri bulunan blok tabanlı görsel bir programlama aracıdır (Bak, Chang, & Choi, 2018). Blockly;

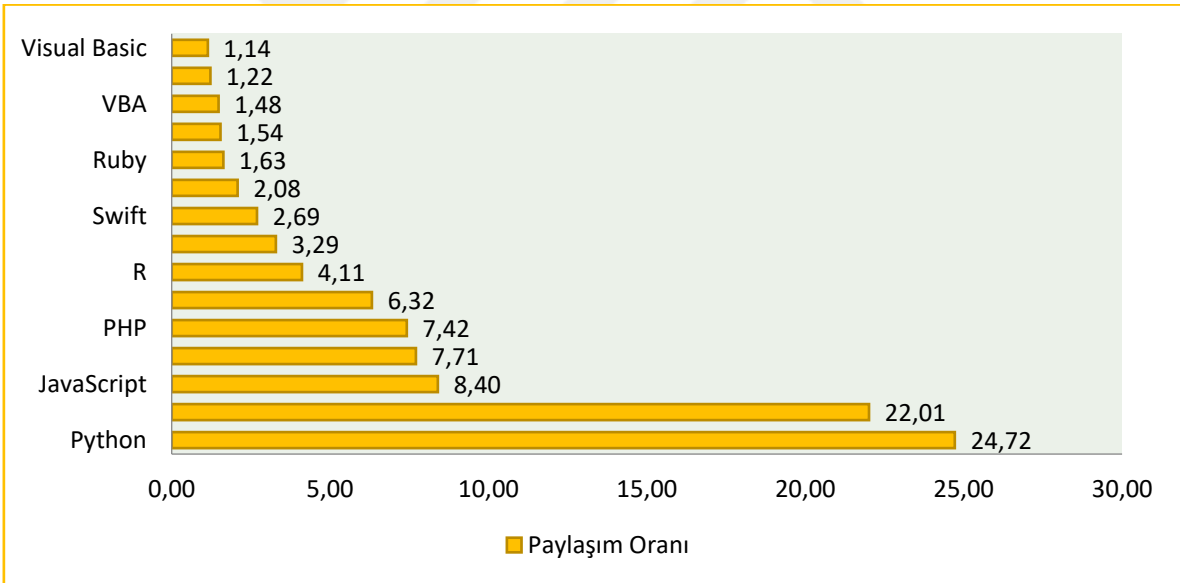
- ❖ Programın bloklarla kodlanmasının yanı sıra, Javascript, Python, PHP, Lua, Dart ve XML gibi farklı programlama dillerinde de derlenebilmesi,
- ❖ Her blok yapısının hem görev hem de yapısal olarak değiştirilebilmesi,
- ❖ Geliştirilebilir açık kod yapısıyla kullanıcıların ortama yeni bloklar ekleyip çıkarabilmesine olanak tanınması,



- ❖ Hem kod bloklarını, hem de arkada çalışan programlama diline ait söz dizimi kod çıktısını aynı anda gösterebilmesi,
- ❖ Bir web tarayıcıda ya da bir web sitesine gömülü olarak internet bağlantısı olmadan çalışabilmesi,
- ❖ Mobil ortamlarda çalışan Android ve IOS sürümlerine de sahip olması gibi özellikleriyle diğer görsel programlama araçlarından ayrılan avantajlara sahiptir.

Bu avantajları nedeniyle; MEB tarafından ortaokulda zorunlu olan algoritma öğretiminden sonra lise düzeyindeki öğrencilere bir programlama dili öğretiminin de zorunlu yapılması dikkate alınarak bu çalışma kapsamında Sosyal Bilimler Lisesi öğrencilerine blok tabanlı programlama aracı olan “Blockly” kullanılarak programlama eğitimi verilmiştir.

Programlama dillerinin ortak noktası var olan bir problemi çözmek olsa da her dilin sahip olduğu bir sözdizimi ve anlam bilgisi vardır. Kitlelerin programlama dili seçimi ise gerek popülariteye göre gerekse kullanım amacına göre değişmektedir. Online bir kod deposu olan GitHub 2017 yılında dünya üzerinde en çok kullanılan 15 programlama dilini açıklamış veriler Şekil 4’te gösterilmiştir (Tekin, 2017).



Şekil 4. 2017 yılında dünyadaki en popüler programlama dilleri.

Şekil 4 incelendiğinde Dünya üzerinde en çok depolanan programlama dilinin 2,4 milyon yazılımcı sayısı ile Python olduğu görülmektedir. İkinci sırada ise 1 milyon yazılımcı ile Python ve üçüncü sırada 986 bin yazılımcı sayısı ile Java yer almaktadır.

Şekillerde görüldüğü gibi hem dünyada hem ülkemizde en çok kullanılan programlama dillerinden biri de Python programlama dilidir. Python; yorumlanmış, etkileşimli ve nesne yönelimli bir programlama dilidir. Oldukça basit ve zarif bir sözdizimine sahip olmasına rağmen birçok kullanıcıya göre yine de güçlü bir dildir. 1990 yılında Guido

Van Rossum tarafından tasarlanan Python, diğer birçok programlama dili gibi ticari amaçlar da dâhil olmak üzere ücretsiz ve pratik bir şekilde herhangi bir modern bilgisayarda çalıştırılabilir (Dalcin *vd.*, 2008; Sanner, 1999). Python programlama yapma aşamasında okunaklı bir dil olması amacıyla tasarlanmıştır (Bogdanchikov, Zhaparov, & Suliyev, 2013). Diğer kullanımı yaygın programlama dillerine göre daha az sayıda sözdizimsel istisnai ve özel durumu mevcuttur (Van Rossum & Fred L. Drake, Jr. (2012). Şekil 5'te Python ile diğer programlama dillerinin söz dizimlerinin karşılaştırıldığı bir örnek sunulmuştur.

```
C++ dilinde:
1 #include <iostream>
2 using namespace std;
3 int main()
4 {
5     cout << " Merhaba Dünya" << endl;
6     return 0;
7 }

Şimdi ise Pascal dili ile yazalım :

1 program merhaba_dunya;
2 begin
3     writeln('Merhaba Dünya');
4 end.

Şimdi ise Java dilinden örnek yapalım :

1 class merhaba
2 {
3     public static void main(String args[])
4     {
5         System.out.println("Merhaba Dünya!");
6     }
7 }

Şimdi ise Python dili ile "Merhaba Dünya" yazalım , farkı görelim.

1 print("Merhaba Dünya")
```

Şekil 5. Python ile diğer programlama dillerinin söz dizimlerinin karşılaştırıldığı bir örnek.

Python programlama dilinin birçok avantajı bulunmaktadır. Örneğin; standart sürümü C dili ile yazılmış olan kütüphanesi tüm kullanıcılar tarafından herhangi bir lisans istenmeden indirilebilmektedir. Kolay bir öğrenim dilinin olması, öğrencilerde kod okuma kolaylığı sağladığı gibi verilmek istenen probleme odaklanılmasını kolaylaştırabilir. Programlama kütüphanesi ile kullanıcıların etkileşim içinde olması, nesne tabanlı programlama dillerinden biri olması, öğrenme olanaklarının yani öğrenmek için ulaşılabilecek kaynakların fazlalığı ve derlenen bir Python programının platform ayırt etmeden her ortamda çalışması Python programlama dilinin diğer avantajları olarak söylenebilir (Bogdanchikov *vd.*, 2013; Cock *vd.*, 2009; Luh, 1994; Van Rossum, 1999).

Bu avantajları nedeniyle MEB bilgisayar bilimi öğretim programında dersin başlangıç seviyesinde zorunlu tutulmamakla beraber Python programlama dilinin tercih edilmesini önermiş ve kaynak kitap olarak hazırladığı e-kitapta Python programlama dilini anlatmıştır (TTKB, 2018). Bu çalışma kapsamında verilen programlama eğitiminde de kullanılan programlama dili Python'dır. Buna bağlı olarak çalışmada kullanılan Blockly programlama öğretim aracı Python programlama dilinin sözdizimi yapısına uygun hale getirilmiştir. Şekil 6'da örnek olarak görüldüğü üzere her bölümün kendi içerisindeki (mantık yapıları, karar yapıları, döngüler, fonksiyonlar, değişkenler vb.) blok metinleri değiştirilmiştir. Böylece blokların, Python programlama dili sözdizimi yapısıyla farklı olmasının önüne geçilmeye çalışılmıştır. Ayrıca Blockly aracının hem kod bloklarını, hem arkaplanda çalışan Python diline ait sözdizimini hem de kod çıktısını aynı anda gösterebilme özelliği de kullanılmıştır.

Blockly değiştirilmeden önceki haliyle değişkene değer atama (her dilde çalışan kod)

```
set sayi to prompt for number with message " Lütfen bir sayı giriniz : "
```

Blockly Python'a uyumlu haliyle değişkene değer atama

```
1 sayi = float(input("lutfen bir sayı giriniz : "))  
2 print sayi
```

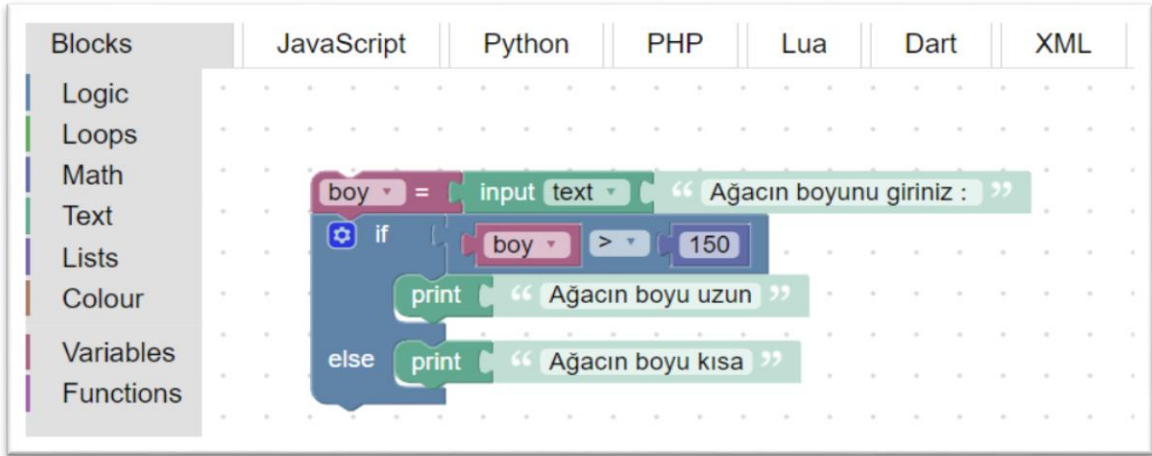
Metin tabanlı programlama ortamında değişkene değer atama

Blockly çıktı ekranı	Python çıktı ekranı
<p>Bu sayfanın mesajı</p> <p>Lütfen bir sayı giriniz :</p> <input type="text"/> <input type="button" value="Tamam"/> <input type="button" value="İptal"/>	<pre>Python 3.6.1 (default, Dec 2015, 13:05:11) [GCC 4.8.2] on linux lütfen bir sayı giriniz :  </pre>

Şekil 6. Blockly'nin Python sözdizimine uygun hale getirildiği "değişkene değer atama" örneği.

Programlama dillerinde kullanılan ve belli koşullara göre belli durumları ortaya koyan karar yapılarını çalıştıran Blockly aracındaki kod blokları Python diline uygun hale getirilmiş ve Şekil 7'de örnek olarak gösterilmiştir.

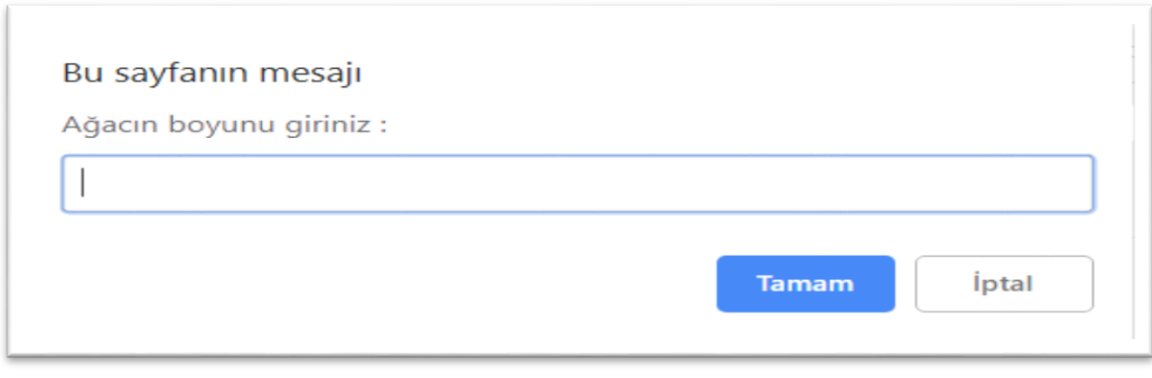
### Blockly'de Blok Ekranı



### Blockly'de Python Uyumlu Kod Ekranı

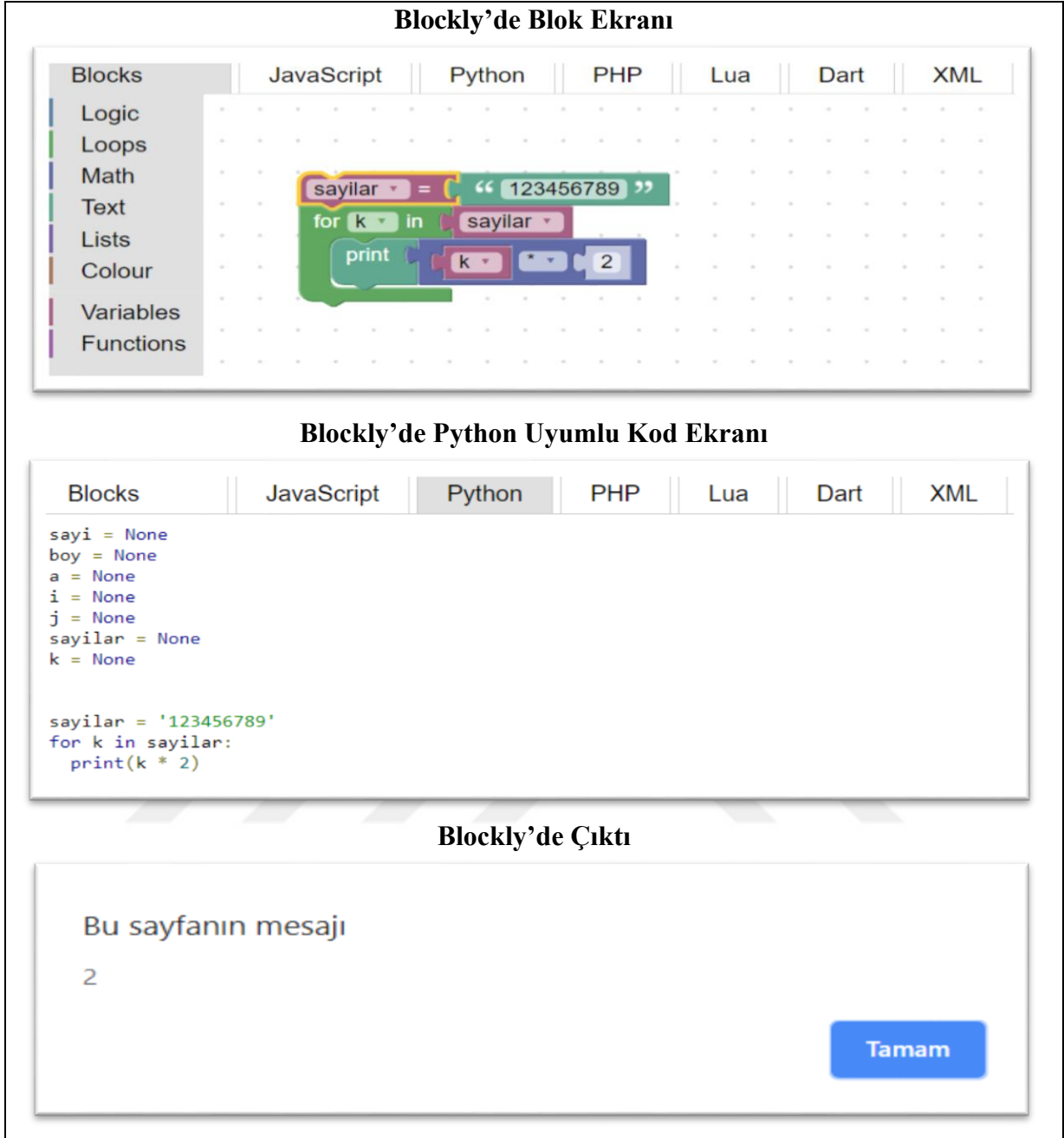


### Blockly'de Çıktı



Şekil 7. Koşul ifadeleri (Blockly kod blokları ve Python dili sözdizimi yapısı).

Döngü yapıları; tekrar edilmek istenen bir işlemi kullanmak gerektiğinde her bir kod satırını tek tek yazmak yerine kullanılan kod yapılarıdır. Blockly aracındaki döngü kod blokları Python diline uygun hale getirilmiş ve Şekil 8'de örnek olarak sunulmuştur.



Şekil 8. Döngü yapısı (Blockly kod blokları ve Python dili sözdizimi yapısı).

### Görsel programlama ile ilgili araştırmalar.

Blok tabanlı görsel programlama araçlarıyla ilgili alan yazında birçok çalışmanın olduğu göze çarpmaktadır (Ünal ve Topu, 2018). 2012-2019 yıllarında SSCI kapsamındaki dergilerin yanı sıra, YÖK Tez Merkezindeki ulusal tezlerde ve Türkiye kökenli dergilerde 53 çalışmaya ulaşılmıştır. Blok tabanlı programlama araçlarının tercih edilme oranları Tablo 2'de sunulmuştur.

Tablo 2. 2012-2019 Yıllarında Yaygın Olarak Kullanılan Blok Tabanlı Programlama Araçları

Programlama Aracı	Frekans	%
Scratch	37	69,91
Kodu Game Lab	4	7,5
Alice	3	5,66
App Inventor	2	3,77
Starlogo Tng	1	1,88
Blockly	1	1,88
Unity 3D	1	1,88
Logo	1	1,88
Robomind	1	1,88
GUI	1	1,88
Small Basic	1	1,88

Tablo 2 incelendiğinde en çok tercih edilen blok tabanlı programlama aracının Scratch olduğu (Chang, 2014; Çakiroğlu vd., 2018; Dohn, 2019; Perez-Marin, Hijon-Neira, Bacao & Pizarro, 2018; Topalli & Cagiltay, 2018), bunu Kodu Game Lab (Dinçer, 2018; Doğan & Kert, 2015; Erkoç, 2018; Fowler vd., 2012) ve Alice (Kaya, 2018; Kayabaşı, 2016; Solmaz, 2014; Zhong, Wang & Chen, 2016) araçlarının izlediği görülmektedir.

Çalışmalarda yaygın olarak incelenen değişkenlerin hangileri olduğu Tablo 3'te sunulmuştur.

Tablo 3. Yaygın Olarak İncelenen Değişkenler

Değişkenler	Frekans	%
Erişi/Başarı	16	27,23
Problem Çözme Becerileri	7	11,86
Motivasyon	6	10,16
Programlamaya Yönelik Tutum	4	6,77
Programlama Üzerine Görüşler	3	5,08
Öz-yeterlik Algısı	3	5,08
Kodlama Becerisi	3	5,08
Matematiksel Düşünme	2	3,38
Eleştirel Düşünme Becerileri	2	3,38
Mantıksal Düşünme Becerisi	1	1,69
Programlama Kaygısı	1	1,69
Düşünme Stilleri	1	1,69
Okuduğunu Anlama	1	1,69
Katılım	1	1,69
Kalıcılık	1	1,69
Yaratıcılık	1	1,69
Kodlama Hazır Bulunuşluk	1	1,69
Kodlama İlgisi	1	1,69
Matematik İlgisi	1	1,69
Diğer	3	5,08

Tablo 3'e göre programlama erişimi/başarısının en çok araştırılan değişken olduğu (Erol, 2015; Feng & Chen, 2014; Korkmaz, 2016; Park *vd.*, 2015; Tsai, 2019), bunu problem çözme becerisi (Çetin, 2012; Kalelioğlu, & Gülbahar, 2014; Shin, & Park, 2014; Solmaz, 2014; Vatansver, 2018), motivasyon (Erol & Kurt, 2017; Saez-Lopez, Roman-Gonzalez, & Vazquez-Cano, 2016; Saygıner, 2017; Kaya, 2018), tutum (Fesakis, & Serafeim, 2009; Yiğit, 2016; Yukselturk, & Altiok, 2017) ve programlama öğretimi üzerine belirtilen görüşlerin (Fesakis, & Serafeim, 2009; Gezgin *vd.*, 2017; Kaya, 2018; Saez-Lopez, Roman-Gonzalez, Vazquez-Cano, 2016; Zamin, Ab-Rahim, Savita, Bhattacharyya & Zaffar, 2018;) takip ettiği görülmektedir.

Çalışmada nicel çalışmaların, nitel ve karma yaklaşımlara göre daha fazla tercih edildiği (Boldbaatar, 2017; Erkoç, 2018; Tsai, 2019), çalışma yapılan öğretim seviyesinin ortaokul (Dzhenzher, 2014; Kasalak, 2017; Zhong, Wong & Chen, 2016) ve üniversite (Genç & Karakuş, 2012; Tsai, 2019) olduğu, buna karşın lise seviyesinde daha az çalışma yapıldığı (Çakiroğlu *vd.*, 2018; Koorssee, Cilliers, & Calitz, 2015), daha çok ölçek ve başarı testi veri toplama araçlarının kullanıldığı görülmüştür. Ayrıca %90'dan fazla çalışmanın BT alanında yapıldığı da belirlenmiştir.

Devam eden bölümde alan yazında blok tabanlı görsel programlama araçlarıyla yapılan çalışmalar özetlenmiştir.

Çetin (2012) Logo blok tabanlı programlama aracını kullandığı çalışmasında çocukların problem çözme becerileri üzerine etkisini incelemiştir. Çalışmada, blok tabanlı programlama eğitiminin öğrencilerin problem çözme becerileri üzerinde olumlu etki yaptığı, programlama eğitiminden memnun kaldıkları, mesleki açısından bilgi ve becerilerine olumlu yansıdığı ve ilerleyen yıllarda da bilgisayarla bu eğitime devam etmek istedikleri sonucuna ulaşmıştır.

Genç ve Karakuş (2012) çalışmalarında bilgisayar oyunları tasarımında Scratch kullanımına yönelik öğrenci görüşlerini almışlar ve öğrenmenin kalıcılığını incelemişlerdir. Nitel ve nicel verilerin kullanıldığı çalışmada programlama dersinden başarısız olan öğrencilerin bile Scratch programlama aracıyla başarılı oldukları ve Scratch aracına karşı olumlu görüş bildirdikleri görülmüştür.

Dzhenzher (2014) Scratch programlama aracının bilgisayar simülasyon dersinde ortaokul öğrencilerinin başarılarına yaptığı katkıyı incelemiş, öğrencilerin Scratch ile iyi vakit geçirdikleri, Scratch'in programlamaya başlamak için iyi bir araç olduğu, sadece eğlenceli bir ortam değil aynı zamanda programlama üzerine ciddi projelerde de kullanılabileceği, modelleme ve yaratıcı çalışma için uygun bir ortam olduğu sonucuna varmıştır.

Feng ve Chen (2014) 232 ortaokul öğrencileriyle yaptıkları araştırmada oyun tasarımında Scratch programlama aracını kullanarak öğrencilerin programlama performansını ve öz-düzenleme becerilerini incelemişlerdir. Çalışma sonucunda öğrencilerin öz-düzenleme becerilerini geliştirebildiklerini ve öğrenme başarılarını arttırabildiklerini belirlemişlerdir.

Gülbahar ve Kalelioğlu (2014) çalışmalarında Scratch programlama aracının 5. sınıf öğrencilerinin problem çözme becerilerine etkisini ve programlama hakkındaki görüşlerini incelemiş, çalışma sonucunda öğrencilerinin problem çözme becerilerinde anlamlı bir artış görülmediği fakat tüm öğrencilerin programlamayı sevdikleri ve ilerde geliştirmek istedikleri sonucuna ulaşmışlardır.

Solmaz (2014), Alice programlama aracının öğrencilerin eleştirel düşünme eğilimleri, problem çözme becerileri, üst bilişsel farkındalık düzeyleri ve ders başarıları üzerindeki etkisini ve ortama ilişkin öğrenci görüşlerini araştırdığı çalışmada Alice programının sadece eleştirel düşünme becerileri üzerinde etkili olduğunu, diğer değişkenlerde anlamlı bir etkisinin olmadığını belirlemiştir. Öğrencilerin Alice programlama aracını eğlenceli ve temel programlama kavramlarını öğrenmede faydalı bulduklarını fakat öğrenilen ileri seviye programlama dili ile bağlantı kuramadıklarını belirttikleri gözlenmiştir.

Demir (2015), blok tabanlı programlama dillerinin farklı kullanımının öğrencilerin programlama başarısı ve kaygısına etkisini araştırdığı çalışmada, blok tabanlı programlama dili entegre edilen grubun, bilişsel ve üst düzey bilişsel beceri puanlarının teori içerikli gruba göre anlamlı farklılık gösterdiğini belirlemiştir. Programlama kaygısı açısından ise uygulama ve teori aşaması blok tabanlı yapılan grupta anlamlı bir farklılık olduğu ancak sadece teori aşaması blok tabanlı yapılan grupta bir fark oluşmadığı gözlenmiştir. Çalışmada blok tabanlı programlama ortamının dersin teori ve uygulama kısmına entegre edilmesinin bilişsel ve üst düzey bilişsel beceriler çerçevesinde öğrencilerin programlama başarısını olumlu yönde etkilediği ve programlama kaygısını düşürdüğü belirtilmiştir.

Doğan ve Kert (2015), Kodu Game Lab blok tabanlı programlama aracı ile bilgisayar oyunu geliştirme sürecinin ortaokul öğrencilerinin eleştirel düşünme becerisi ve algoritma başarısına etkisini incelediği araştırmasında 6 haftalık bir eğitim süreci gerçekleştirmiştir. Bilişim Teknolojileri ve Yazılımı dersi algoritma tasarımı konusunda deney grubuna oyun tasarımı şeklinde verilen eğitim kontrol grubunda geleneksel yöntemlerle işlenmiştir. Çalışma sonuçlarına göre oyun tasarımı şeklinde işlenen algoritma eğitimi, geleneksel şekilde verilen eğitime göre öğrencilerin eleştirel düşünme becerilerini ve algoritma başarılarını olumlu etkilemiştir. Çalışma sonunda öğrenciler oyun tasarımı uygulaması ile bilgisayara olan ilgilerinin arttığını ifade etmişlerdir.



Erol (2015), üniversite düzeyinde 14 haftalık bir eğitimden oluşan çalışmada ilk 7 haftalık periyotta deney grubu öğrencilerine Scratch programlama ortamı ile oyun tasarımı eğitimi verirken kontrol grubu öğrencilerine klasik yöntemlerle akış şeması eğitimi vermiştir. İkinci 7 haftalık bölümde ise her iki gruba da ileri düzey bir programlama dili olan C# eğitimi vermiştir. Araştırma sonucunda öğrencilerin motivasyon puanlarının deney grubunda ön-test ve son-test bağlamında anlamlı bir farklılık gösterdiği, kontrol grubunda ise motivasyon puanlarının azaldığı gözlenmiştir. Programlama başarısında ise deney grubu lehine artış olduğu ortaya çıkmıştır. Ayrıca öğrenciler Scratch ortamının eğlenceli ve ilgi çekici olduğu, programlama öğretiminde motivasyonu artırıcı etkisi olduğu fakat programlama eğitiminde bazı temel yapıların öğretilmesinde bu aracın yetersiz kaldığı yönündeki görüşleriniiletmişlerdir.

Koorsse,Cilliers ve Calitz (2015) çalışmalarında lise öğrencilerinin blok tabanlı programlama aracının çeşitli konular üzerine etkileri incelemiş ve bazı programlama konuları üzerinde anlamayı kolaylaştırdığı ortaya çıkmıştır.

Kayabaşı (2016), Alice programlama aracının kullanımıyla öğretmen adaylarının programlama deneyimlerini ve öz-yeterlik algı düzeylerini belirlemiştir. Çalışma sonucunda öğretmen adaylarının öz-yeterlik algılarında anlamlı bir farklılık olduğu belirlenmiştir. Öğretmen adaylarının Alice programlama aracının eğlenerek ve uygulayarak öğrenme imkânı sunduğu yönünde görüş bildirdikleri belirlenmiştir.

Sáez-López, Román-González ve Vázquez-Cano (2016) İspanya’da 5 ortaokulda 2 yıl boyunca yürüttükleri çalışmalarında, Scratch programlama aracının, öğrencilerin programlamaya karşı tutum, motivasyon ve algılanan yararlık değişkenlerine etkisini incelemiştir. Süreç sonunda blok tabanlı programlama ortamının farklı alanlarda öğrencilerin tutum ve motivasyonunu artırdığı, algılanan yararlık ortalama puan değerlerinin yüksek çıktığı belirlenmiştir. Scratch’in eğlenceli bir içerik sunduğu, sanat ve tarih alanlarında içerik oluşturma, mantık yapılarını kavrama ve matematik gibi alanlarda katkıları olduğu sonucuna varılmıştır.

Yiğit (2016) üniversite öğrencilerinin programlamaya karşı tutumlarını ve programlama başarılarını araştırdığı çalışmada 42 üniversite öğrencisiyle birlikte Programlama Dilleri dersinde 8 haftadan oluşan ve deney grubu öğrencilerinin Blockly programlama aracını kullandığı, kontrol grubu öğrencilerinin ise geleneksel öğretimle eğitim gördüğü bir süreç ortaya koymuştur. Çalışma sonunda deney grubu öğrencilerinin kontrol grubu öğrencilerine göre programlama başarılarının daha yüksek olduğu, ancak

programlamaya karşı tutum puanları arasında anlamlı bir farkın ortaya çıkmadığı tespit edilmiştir.

Zhong, Wang ve Chen (2016) çalışmalarında cinsiyet ve arkadaşlık ilişkileri çerçevesinde ortaokul öğrencileriyle oluşturulan ikili gruplar arasında öğrencilerin programlama başarısını araştırmıştır. Çalışma sonucunda erkek grup, kız grup ve kız-erkek grupların arasında programlama başarısı açısından anlamlı bir farkın olmadığı ortaya çıkmıştır. Kız grupların çalışma sonunda daha üretken oldukları ve tüm gruplar arasında arkadaşlık ilişkilerinin geliştiği gözlenmiştir.

Boldbaatar (2017) çalışmasında üniversitesi son sınıf öğrencilerinin oyun tabanlı öğrenme dersi kapsamında görsel programlama aracı kullanarak daha önceki programlama tecrübelerini oyun tasarım ortamına nasıl aktardıkları incelenmiş, çalışan örnek, tamamlamalı örnek ve tam uygulamalı örnekler çalışma boyunca öğrencilere verilmiştir. Sonuçta ise metin tabanlı programlama deneyimi olan öğrencilerin daha yüksek final notları aldıkları, tamamlamalı örneğin tecrübeli öğrenciler açısından daha etkili olduğu, tam uygulamalı örneklerin diğer örneklerle göre öğrencileri daha çok zorladığı belirlenmiştir.

Erol ve Kurt (2017) Scratch programlama aracını kullandıkları çalışmada algoritma öğretiminde öğrencilerin motivasyon ve programlama başarısını incelemiştir. Çalışma sonucunda deney ve kontrol grubu öğrencilerinin programlama başarı puanlarının arttığı, motivasyon puanlarında ise deney grubu lehine anlamlı bir artışın olduğu fakat geleneksel yöntemin kullanıldığı kontrol grubunda motivasyon puanlarında bir azalma olduğu belirlenmiştir.

Kasalak (2017) ortaokul öğrencilerinin robotik kodlama eğitimi ve blok tabanlı programlama eğitimine ilişkin öz-yeterlik algılarını araştırdığı çalışmasında her iki grup için de 5 haftalık eğitim programı hazırlamıştır. Hem basit hem karmaşık blok tabanlı programlama eğitimine yönelik öz-yeterlik algılarında her grup için son test lehine anlamlı bir fark olduğu, cinsiyet, evde bilgisayar ve internet bağlantısı sahipliği ve ders dışı Scratch programlama ortamı çalışma olanağına göre anlamlı bir farkın olmadığı tespit edilmiştir. Robotik kodlama eğitimi sonunda öğrenci yaşantıları tespit edilmiş, öğrencilerin etkinlikleri ilgi çekici ve eğlenceli buldukları, kişisel gelişimlerine olumlu katkı sağladığını düşündükleri ve etkinliklere katılmaya istekli oldukları yönünde fikir ortaya koymuşlardır.

Saygıner (2017) 60 üniversite öğrencisinden oluşan çalışma grubuyla yaptığı araştırmada 10 hafta süren bir periyotta blok tabanlı ve metin tabanlı programlama öğretiminin öğrencilerin erişimi, mantıksal düşünme ve motivasyon seviyelerine etkisini araştırmıştır. Blok tabanlı programlama aracıyla eğitim yapılan grupta öğrencilerin erişimi,

motivasyon ve mantıksal düşünme puanlarının seviyesinde artış gözleendiği, metin tabanlı programlama aracıyla eğitim yapılan grupta, öğrencilerin mantıksal düşünme ve erişme seviyelerinde artış gözlenirken motivasyon seviyelerinde anlamlı bir artışın olmadığı ortaya çıkmıştır.

Yıldırım (2017) Scratch programlama aracının mobil cihazlar üzerinden eğitim aracı olarak kullanılmasının öğrencilerin programlama becerilerine etkisini araştırdığı çalışmasında 94 ortaokul öğrencisi araştırmacının çalışma grubunu oluşturmuş ve çalışma sonucunda mobil uygulama üzerinden yapılan uygulamanın öğrencilerin kodlama becerisi üzerinde olumlu bir etkiye sahip olduğu ortaya çıkmıştır.

Yükseltürk ve Altıok (2017) 151 üniversite öğrencisi ile yaptıkları araştırmalarında Scratch programlama aracına yönelik öğrencilerin öz-yeterlik algıları ve programlamaya karşı tutumlarını incelemiştir. Verilen tüm karmaşık görevlere rağmen öğrencilerin öz-yeterlik algılarında anlamlı düzeyde artış olduğu görülmüş, öğrencilerin bilgisayara karşı tutumlarında ise önemli ölçüde olumlu etkiler ortaya çıkmıştır.

Diğer (2018) Scratch ve Kodu Game Lab görsel programlama ortamlarını, öz-yeterlik algısı, programlama başarısı ve tutum bağlamında karşılaştırdıkları çalışmada, 20 ders saati boyunca oluşturulan 2 ayrı grupta Scratch ve Kodu Game Lab eğitimleri vermişlerdir. Çalışma sonucunda Scratch ile programlama eğitimi alan öğrencilerin Kodu Game Lab ile programlama eğitimi alan öğrencilere göre programlama başarısı açısından önde oldukları, gruplar arasında tutum ve öz-yeterlik algıları açısından anlamlı bir farkın olmadığı, her grubun kendi içerisinde tutum ve öz-yeterlik algılarının arttığı görülmüştür.

Erdem (2018) çalışmasında yüz yüze eğitim ve ters yüz sınıf modeli olan teknoloji destekli öğrenme ortamlarında Scratch aracıyla yapılan programlama öğretiminin öğrencilerin bilgi işlemsel düşünme becerilerine ve öz-yeterliklerine etkisini araştırmıştır. 79 ortaokul öğrencisiyle yapılan çalışmada hem nicel hem de nitel bulgulara ulaşılmıştır. Buna göre; her iki grupta da öğrencilerin bilgi işlemsel düşünme beceri puanları arasında anlamlı olmayan bir fark oluşmuştur. Ayrıca grup içi öz-yeterlik puan ortalamalarında anlamlı bir farklılık olduğu belirlenmiştir.

Erkoç (2018) çalışma grubunu 48 adet ortaokul öğrencisinin oluşturduğu araştırmada, deney grubuna iki farklı alt grupta bireysel oyun geliştirme ve işbirlikli oyun geliştirme yaklaşımı, kontrol grubunda ise ders programı kapsamında eğitimler vermiştir. Süreç sonucunda öğrencilerin eleştirel düşünme, problem çözme ve akademik başarı puanları incelenmiş, eleştirel düşünme becerisi açısından herhangi bir grup lehine anlamlı bir fark olmadığı, problem çözme becerisi açısından işbirlikli oyun geliştirme kullanılan grup lehine

anlamalı bir fark tespit edilmiştir. Akademik başarı puanlarında ise deney grubu lehine anlamlı bir fark olduğu saptanmıştır.

Kaya (2018) Alice görsel programlama ortamının 63 önlisans öğrencisinden oluşan çalışma grubunun akademik başarılarına, problem çözme beceris algılarına, motivasyonlarına ve programlama hazır bulunuşluk düzeylerine etkisini incelemiştir. Karma yöntemin temel alındığı çalışmada araştırmanın nicel bulgularına bakıldığında, Alice ortamının kullanıldığı deney grubunda öğrencilerin akademik başarıları, problem çözme becerisi algıları ve motivasyonları lehine anlamlı bir fark olmadığı ancak programlama hazır bulunuşlukları açısından anlamlı bir fark olduğu görülmüştür. Nitel bulgulara bakıldığında ise öğrenciler, Alice ortamının programlama mantığını anlamakta yardımcı olduğunu, programlama öğrenme isteklerini artırdığını ifade etmişlerdir.

Marcelino, Pessoa, Vieira, Salvador ve Jose Mendes (2018) 9 ilköğretim okulu öğretmenin yer aldığı çalışmalarında Scratch programlama ortamlarının, öğrencilerin bilgisayarca düşünme becerilerine etkisini incelemişler, öğrencilerin bilgisayarca düşünme becerilerinde ve teknoloji kullanım seviyelerinde artış olduğunu gözlemlemişlerdir.

Perez-Marin, Hijan-Neira, Bacelo ve Pizarro (2018) programlama öğretimi üzerine 9-12 yaş aralığında 132 ilköğretim öğrenci ile yaptıkları çalışmalarında programlama öğretimi için çeşitli metaforlar geliştirmişlerdir. Bu metaforlar eşliğinde Scratch aracı kullanılarak yapılan programlama eğitiminin öğrencilerin başarısına olumlu etki ettiği ve öğrencilerin bilgisayar kavramlarını kolay bir şekilde öğrendikleri sonucuna ulaşmışlardır.

Topallı ve Çağıltay (2018) 395 mühendislik öğrencisi ile gerçekleştirdikleri çalışmalarında hayatta karşılaşılan problemlere dayalı olarak geliştirilen oyunlarla, programlama derslerine girişin kolaylaştırılmasını ve öğrencilerin programlama becerileri ile motivasyonlarının artırılmasını amaçlamışlardır. Süreç sonunda öğrencilerin programlama performanslarının ciddi oranda arttığı gözlenmiştir.

Vatansever (2018) çalışma grubunu 226 ortaokul öğrencilerinin oluşturduğu çalışmada Scratch görsel programlama aracının öğrencilerin problem çözme becerisine etkisini incelemiştir. Çalışmada Scratch'in öğrencilerin problem çözme becerileri üzerinde anlamlı bir fark oluşturduğu, problem çözme becerisine cinsiyet ve sınıf seviyesinin etkisinin olmadığı belirlenmiştir. Ayrıca araştırmacı tüm hedeflerin gerçekleşmesi için tasarım odaklı yaklaşım uygulanması önerisini getirmiştir.

Zamin *vd.* (2018) Avustralya ve Malezya olmak üzere iki ayrı ülkede gerçekleştirdikleri çalışmalarında 8-12 yaş aralığındaki öğrencilere Scratch programlama

aracı kullanarak kurs aktiviteleri düzenlemiş ve öğrencilerin programlamaya yaklaşımlarını incelemiştir. Çalışma sonucunda öğrenciler kodlama sınıfında eğlendiklerini, kodlamayı eğlenceli bulduklarını, gelecekte kodlama becerilerini ilerletmeyi düşündüklerini ve kendilerine uygulanan öğretim şeklinin hoşlarına gittiğini ifade etmişlerdir.

Dohn (2019) araştırmasında Scratch programlama ortamının öğrencilerin matematik ilgisini etkileyip etkilemediğini incelemiştir. Çalışma grubunu 44 ortaokul öğrencisi oluştururken Scratch kullanımıyla birlikte kodlama ve matematik etkinliklerinin öğrencilerin ilgilerini azalttığı ancak oyun ve animasyon kullanımının öğrencilerin ilgisini arttırdığı tespit edilmiştir.

Tsai (2019) 180 üniversite öğrencisinden oluşan düşük, orta ve yüksek öz-yeterlik seviyesindeki gruplarda App Inventor kullanılan ve kullanılmayan uygulamalarda öğrencilerin programlama başarısını incelemiştir. Araştırma sonuçlarına göre App Inventor kullanılan uygulamalarda özellikle orta ve yüksek öz-yeterlik seviyesindeki gruplarda programlama başarısının daha çok arttığı görülmüştür.

İncelenen çalışmaların amacı, çalışmalarda kullanılan görsel programlama aracı, örneklem düzeyi ve sayısı, incelenen değişkenler, temel alınan araştırma yöntemleri ve elde edilen sonuçlar özetlenerek Ek-9'de gösterilmektedir.

Ek-9 incelendiğinde çalışmalarda çoğunlukla tercih edilen blok tabanlı programlama aracının Scratch olduğu görülmektedir (Dohn, 2019; Marcelino, Pessoa, Vieira, Salvador, & Mendes, 2018; Perez-Marin, Hijon-Neira, Bacelo, & Pizarro, 2018). Oysa Scratch blok tabanlı programlama aracının, algoritma mantığını öğrencilere kazandırırken gerçek bir programlama dili öğretimi gerçekleştirmediği alanyazında vurgulanmaktadır (Meerbaum-Salant, Armoni & Ben-Ari, 2011). Bu nedenle incelenen çalışmalarda çok az kullanılmasına rağmen bu çalışmada Blockly aracı tercih edilmiştir. Blockly aracı; programlama dilinin sözdizimini, blok yapısıyla pekiştirirken, o dilin yapısını da öğrencilere öğretme potansiyeline sahiptir. Bu nedenle MEB 2017 yılında, “öncü” ortaöğretim kurumları olarak nitelendirilen fen liseleri ve sosyal bilimler liselerinde, zorunlu diğer kurumlarda seçmeli olmak üzere bir programlama dili öğretimini kazanımlarıyla mümkün kılan Bilgisayar Bilimi dersini müfredata dâhil etmiştir (TTKB, 2017). Çalışmanın yapıldığı Sosyal Bilimler Lisesi ve dersin kazanımları göz önüne alındığında Blockly görsel programlama aracı farklı programlama dillerinin söz dizimlerini de görmeyi sağladığı için bu çalışmada tercih edilmiş ve öğrencilerin programlama başarıları tespit edilmeye çalışılmıştır.

İncelenen çalışmaların daha çok ortaokul (Dzhenzher, 2014; Kasalak, 2017; Zhong, Wong & Chen, 2016) ve üniversite (Genç & Karakuş, 2012; Tsai, 2019) öğrenim

seviyelerinde yapıldığı görülmüştür. Bu çalışmanın katılımcı grubunun sosyal bilimler lisesi öğrencileri olduğu düşünüldüğünde çalışmanın bu açıdan özgünlük taşıdığı söylenebilir. Diğer taraftan incelenen çalışmalarda nicel yöntemlerin daha çok tercih edildiği görülmüştür. Bu çalışmada ise karma araştırma desenlerinden çeşitleme yapılmış, nicel ve nitel veriler birlikte yorumlanarak daha kapsamlı ve bütüncül bir bakış açısıyla sonuçlar irdelenmiştir. Ayrıca çalışmalarda daha çok başarı, tutum, motivasyon değişkenlerinin incelendiği görülmüştür (Erol, 2015; Feng & Chen, 2014; Kaya, 2018; Saez-Lopez, Roman-Gonzalez, & Vazquez-Cano, 2016; Saygıner, 2017; Tsai, 2019; Yiğit, 2016). Bu çalışmada ise programlama başarısının yanı sıra programlama kaygısı ve bilişsel yük dışsal, içsel ve etkili bilişsel yük alt boyutlarıyla birlikte ele alınmıştır. Bu durum çalışmayı diğer araştırmalardan farklı kılmaktadır.

### **Programlama Öğretimi ve Kaygı**

Kaygı, insanın tepki vereceği bir uyarıcıyla karşılaştığında, bedensel, duygusal ve zihinsel olarak ortaya çıkan uyarılmışlık halidir (Mayer, 2008). Kaygı, bireyin kendisini tehdit eden bir ortamda huzursuz ve endişeli hissetme hali olarak da tanımlanabilir (Işık, 1996; Jiao, Q. G., & Onwuegbuzie, A. J., 2017) Çavuş ve Günbatar (2008) kaygıyı yaratıcı ve yapıcı davranışları teşvik eden ya da engelleyen bir duygu, dürtü olarak nitelendirmişlerdir. Kaygı çeşitli bağlamlarda (herhangi bir teknolojiyi kullanma, dersi başarma, uygun davranış sergileme vs.) ortaya çıkabilen bir duygu olduğundan Baltaş (2002)'ın da belirttiği gibi öğretim yapılan ortamlarda çoğunlukla kendisine yer bulan bir değişken olduğu söylenebilir.

Özellikle bilişim teknolojilerindeki gelişmeyle birlikte bilgisayar destekli uygulamaların artışı “bilgisayar kaygısı” kavramını da beraberinde getirmiştir. Bilgisayar kaygısı; bilgisayar teknolojisini kullanırken veya bilgisayar kullanma ihtimali ortaya çıktığı sırada hissettiği korku ve endişe olarak tanımlanabilir (Cock *vd.*, 2009). Alan yazına bakıldığında bilgisayar kaygısının cinsiyet, yaş, bilim dalı, bilgisayar kullanım oranı ve tecrübesi (Cazan, Cocoradă, & Maican, 2016; Cocorada, 2014; Mathew, 2012; Sam, Othman, & Nordin, 2005; Simsek, 2011) gibi değişkenlere göre farklılık gösterdiği görülmektedir. Diğer taraftan yüksek bilgisayar kaygısı yaşayan öğrencilerin bilgisayar temelli etkinliklerde gelişme gösteremedikleri de alan yazında belirtilmiştir (Gürcan, 2005).

Bilgisayar temelli bir öğrenme süreci gerektiren programlama öğretiminde de öğrencilerin bilgisayar kaygısıyla birlikte gelişen programlama kaygısı yaşamaları olasıdır. Bunun yanı sıra programlama kaygısı; programlama öğretimi konusunda karşılaşılan kişisel güçlüklerden biri olarak da karşımıza çıkmaktadır (Özmen & Altun, 2014). Rogerson ve Scott (2010) programlama öğretimi sürecinde öğrencilerin korkuları arasında programlama

kaygısının da yer aldığını belirtmiştir. Programlama kaygısı, öğrencilerin programlama becerilerini yanlış değerlendirmelerinden dolayı ortaya çıkabilmektedir (Connolly vd., 2007).

Programlama öğretimiyle ilgili çalışmalarda programlama kaygısına yönelik sonuçların da olduğu göze çarpmaktadır. Örneğin; bilgisayar kaygısının (Achim & Al Kassim, 2015), programlamaya karşı algının (Zainal vd., 2012), programlama dillerindeki karmaşık söz diziminin (Saeli, Perrenet, Jochems, & Zwaneveld, 2011), programlamaya karşı oluşan kalıplaşmış düşüncelerin (Connolly vd., 2007), programlama içeriklerini oluştururken yabancı bir dil kullanımının (Ozoran, Cagiltay, & Topalli, 2012), programlamada kullanılan yapıları anlamakta yaşanan zorlukların (Katai & Toth, 2010) ve program tasarımına bağlı sebeplerin (Lahtinen, Ala-Mutka & Jarvinen, 2005) programlama kaygısına neden olabileceği incelenen çalışmalarda ortaya çıkmıştır. Buna bağlı olarak Şekil 9'da programlama kaygısını tetikleyen etmenler sunulmuştur.



Şekil 9. Programlama kaygısını tetikleyen etmenler.

Programlama kaygısıyla ilgili incelenen çalışmaların çoğunda bilgisayar kaygısı programlama kaygısı değişkeni olarak ele alınmıştır (Achim, & Al Kassim, 2015; Chang, 2005; Owolabi, Olanipekun, & Iwerima, 2018). Ancak bu çalışmada bilgisayar kaygısı yerine programlama kaygısı değişkeni ele alınmıştır. Bu açıdan çalışmanın alana katkı sağlayacağı düşünülmektedir. Diğer taraftan çalışmalarda daha çok algoritma mantığının yerleştirilmesine yönelik olarak Scratch, Kodu Game Lab gibi programlama dili kod söz diziminin olmadığı blok tabanlı görsel programlama araçlarının kullanıldığı görülmektedir (Demir, 2015; Wu, Chang, & He, 2010). Bu nedenle hem programlamada yer alan kod söz diziminin gösterildiği ve hem de blok tabanlı olan Blockly görsel programlama aracıyla yapılan bu çalışmanın öğrencilerin programlama kaygılarını azaltıp azaltmayacağını belirlemenin önemli olduğu düşünülmektedir. Bunun yanında Blockly'de yer alan kod bloklarının öğrenim yapılan programlama diline göre etiketlerinin değiştirilebilmesi, yeni kod bloklarının

eklenebilmesi, tasarım öğeleri olan renk, boyut, stil gibi unsurların farklılaştırılabilmesinin özellikle acemi kullanıcılar için avantaj olabileceği öngörülmektedir. Blockly'nin farklı dillerde destek sağlamasının da öğrencilerin programlama kaygısına neden olan yabancı dil unsuruna alternatif olacağı söylenebilir.

### **Programlama Öğretimi ve Bilişsel Yük**

Öğrencilerin, bir öğrenme görevini yerine getirirken bilişsel olarak zorlanmaları bilişsel yükü oluşturur (Sweller, Van Merriënboer, & Paas, 1998). Diğer bir ifadeyle bilişsel yük; öğrencilerin öğrenme ortamlarında belirli görevleri yürütürken, bilgi işleme süreçlerindeki limit sebebiyle bilişsel yapılarındaki yüküdür (Kirschner, 2002). Chandler ve Sweller (1991) bilişsel yükü öğrenim faaliyeti sırasında çalışan bilişsel belleğe yüklenen bilişsel kaynaklar olarak tanımlar. Bilişsel yük, belirli bir görevi gerçekleştiren belirli bir öğrencinin bilişsel sistemi üzerine yükleyen yükü temsil eden çok boyutlu bir yapı olarak düşünülebilir (Paas, Renkl, & Sweller, 2003). Bilişsel yük kuramı ise; sınırlı kapasiteye sahip çalışan belleğin nasıl en verimli şekilde çalışabileceğiyle ilgilenir (Paas & Van Merriënboer, 1994; Salleh, Shukur, & Judi, 2018). Ayrıca kuram, uzun süreli bellekte oluşan ya da oluşması gereken şemaların otomatik olarak harekete geçmesi için, ona yardımcı olacak çalışan belleğin sağlıklı çalışabilmesi için, üzerindeki yükün farklı öğretim materyalleriyle hafifleyebileceğini savunur (Kirschner, 2002; Takır, 2011). Bilişsel yük kuramına göre üç çeşit bilişsel yükten söz edilmektedir. *İçsel Bilişsel Yük*, öğrenim görevlerinin içsel doğasından kaynaklanmaktadır; *Dışsal Bilişsel Yük*, görevlerin sunum şekliyle ilgilenmektedir; *Etkili Bilişsel Yük* ise, görevin bellekte oluşturacağı şemalarla ilgilenmektedir. Üç bilişsel yük türünün bir araya gelmesiyle çalışan bellekte oluşacak yük meydana gelebilir (Paas, Tuovinen, Tabbers, & Van Gerven, 2003).

#### **İçsel bilişsel yük.**

İçsel bilişsel yük, öğretim materyalinin kendi doğasından kaynaklanan, öğretici ya da materyal tasarımcılarının dışardan müdahalesi söz konusu olmayan, öğrenilmesi zor olarak nitelendirilen, çalışan bellek üzerinde oluşan yük ve öğrenme süreçlerini olumsuz etkileyen bir durum olarak belirtilmektedir (Caiping & Xiong, 2009). İçsel bilişsel yük, öğretilecek konunun içsel yapısıyla ve kişinin konuyu öğrenme yeteneğiyle ilgilidir (Sweller, 2010). Öğrenilecek konunun içeriğinin karmaşıklığı içsel bilişsel yüke neden olan bir unsurdur (Çakmak, 2007; Sweller, Ayres, & Kalyuga, 2011). Öge etkileşimi ve öğelerin birbiriyle ilişkisindeki yoğunluğun çalışan bilişsel yükün miktarını etkilediği, bu durumun da içsel bilişsel yüke neden olabileceği öngörülmüştür (Clark, Nguyen, & Sweller, 2011). İçsel



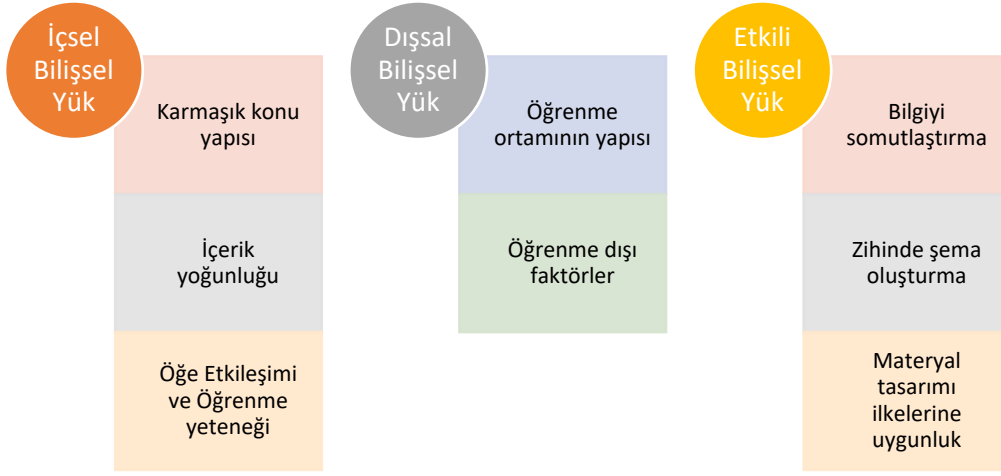
bilişsel yükü, içeriklerden ötürü doğrudan azaltmak mümkün değildir. Fakat öğrenciye verilecek görevi, sıraya koyarak, basit ve küçük adımlarla uygulayarak yeniden tasarımıyarak bu durum sağlanabilir (Doolittle, McNeill, Terry, & Scheer, 2005).

### **Dışsal bilişsel yük.**

Dışsal bilişsel yük, öğretim materyallerinin yapısından kaynaklanan, öğrenciye verilen görevle bağlantısı olmayan, gereksiz zihinsel çaba sarf etmeye neden olan bilişsel aktivitelerin olduğu ancak öğretim materyalini hazırlayan kişi tarafından kontrol altına alınabilen, çalışan belleğe baskı yapan yük ve öğrenme süreçlerini olumsuz etkileyen bir durum olarak ifade edilmektedir (Paas *vd.*, 2003; Sweller, 2010). Konu dışı bilişsel yükü azaltmak için çalışma belleğinin alt kanalları olan sözel ve görsel kanalların birlikte ve etkili bir biçimde kullanılması gerekir. Sadece tek bir kanalın kullanılması öğrencilerde odaklanma sorununu meydana getirecektir. Bu da konu dışı bilişsel yükün artmasına neden olacaktır (Moreno & Mayer, 1999).

### **Etkili bilişsel yük.**

Etkili bilişsel yük, öğrenme etkinliklerinin tümünün bilişsel yapı üzerine kurduğu baskıdır. Zihinde bilginin oluşmasını sağlayan olumlu bir durumdur (Cierniak, Scheiter & Gerjets, 2009). Etkili bilişsel yük de öğretici ya da öğretim materyali tasarımcısı tarafından kontrol altına alınabilir. İçeriğin karmaşıklığını azaltıp, görsel işitsel öğelerin kullanımı dengelenerek, ilişiksiz bilgiler azaltılarak, bilgiler parçalara bölünüp parça bütün ilişkisi korunarak, bilgilerdeki aşamalılık gözetilerek, içeriğin biçimsel özellikleri düzenlenerek, dikkatin bölünmesi önlenerek, öğrencilerin ön bilgi eksikliği giderilerek ve hazırbuluşlukları sağlanarak etkili bilişsel yük artırılabilir (Clark *vd.*, 2011; Paas & Van Merriënboer, 1994; Sweller *vd.*, 2011, Moreno & Mayer, 1999). Etkili bilişsel yükün artması şemaların oluşması ile gerçekleşir ve öğrenmenin somutlaştığını gösterir. Şemalar ilerde karşılaştığımız benzer durumlarda ne tepki vereceğimizi bize söyleyen yapılardır (Orru & Longo, 2018). Dışsal ve içsel bilişsel yükün azalması, yani materyal tasarımında öğrencilerin karmaşık yapıdan kurtulup aşama aşama öğrenim görevini yerine getirmesini sağlayıcı tasarımlar yapılması, çalışan bellek üzerinde etkili bilişsel yüke daha çok yer açılmasını sağlayacak ve bilişsel şemalar oluşabilecektir (Paas *vd.*, 2003; Sweller, 2010). Buna göre kalıcı öğrenmeler için dışsal bilişsel yükün azaltılması, içsel bilişsel yükün iyi yönetilmesi, etkili bilişsel yükün ise artırılması gerektiği söylenebilir. Şekil 10'da içsel, dışsal ve etkili bilişsel yükün etkileyen etmenler özetlenmiştir.



Şekil 10. Bilişsel yükü oluşturan alt boyutlar.

Programlama dillerinin doğası gereği karmaşık işlemlerin fazlalığı nedeniyle programlama öğretim süreci öğrencilerin zihinsel yapılarında zorlanmaya ve yüke neden olabilir. Alt boyut olarak içsel bilişsel yük; konu yapısıyla ve karmaşıklığıyla, içeriğin yoğunluğuyla alakalı olduğundan dışardan herhangi bir müdahale ile değiştirilemez (Sweller, J., Ayres, P., & Kalyuga, S. (2011). konunun parçalara bölünerek, parça bütün ilişkisi kurularak adım adım öğrenciye aktarılması içsel bilişsel yüklenmeyi azaltabilir (Sweller, J., Van Merriënboer, J. J., & Paas, F. G. (1998). Diğer bir alt boyut olan dışsal bilişsel yük, bireye verilen öğrenme ödevinin gerçekleştirildiği öğrenme ortamı ve öğrenme dışı faktörler ile ilgili olduğundan (Sweller, 2010) öğrenme materyalinin karmaşık olmayan basit yapısı ve tasarım ilkelerine uygunluğu dışsal bilişsel yüklenmeyi azaltabilir. Etkili bilişsel yük ise bilginin somutlaştırılmasına yönelik destekleyici materyallerle, öğrenme yöntemleriyle ilişkili olduğundan (Sweller, 2011) zihinde kalıcı ve etkili bir şema oluşumu için etkili bilişsel yükün bilişsel süreçlerde aktif rol alması gerekir . Dışsal bilişsel yükü etkileyen uygun renk kullanımı, yönlendirme, boyut, hareket, konumlandırma gibi özelliklerin etkin kullanımıyla çalışma belleğinde daha düşük bir yüke sebep olarak etkili bilişsel yük için çalışma alanı oluşturulabilir. Nitekim içsel ve dışsal bilişsel yükün artması belleğinin etkili çalışma alanını daraltacağından öğrenciler için programlama öğretiminde bir engel olarak karşılımları çıkabilir. Buna rağmen incelenen çalışmalarda bilişsel yük değişkenine değinen çok az araştırmaya rastlanmıştır (Abdul-Rahman & Du Boulay, 2014; Chang vd., 2011; Çakiroğlu vd., 2018; Salleh vd., 2018). Mevcut çalışmalarda bilişsel yük sadece tek bir boyut olarak ele alınmış alt boyutları irdelenmemiştir. Bu çalışmada ise; Blockly görsel programlama aracı kullanılarak karmaşık ve çok adımlı programlama etkinlikleri parçalara bölünmüş, adım adım uygulanmış, hem basit bir sürükle bırak işlemiyle görsel nesnelere temsil edilen kod blokları kullanılmış hem de birleştirilen blokların arka planında çalışan kod yapısı gösterilmiştir. Böylece içsel bilişsel yük kontrol altına alınmaya çalışılmıştır. Ayrıca ortamdaki konu dışı

yapı ve içerikler azaltılarak dışsal bilişsel yüklenmenin önüne geçilmeye çalışılmış ve belleğin etkili çalışması için alan açılarak kalıcı öğrenme sağlanmaya çalışılmıştır. Buna göre Blockly programlama aracının kullanıldığı ve kullanılmadığı gruplarda öğrencilerin bilişsel yüklenme durumları alt boyutlarıyla (içsel, dışsal ve etkili bilişsel yük) birlikte incelenmiştir. Çalışma bu yönüyle diğer araştırmalardan ayrılmaktadır. Diğer taraftan deneysel yöntemlerde bilişsel yükü ölçmek adına öznel, fizyolojik ve görev-performansa dayalı teknikler kullanıldığı bilinmektedir (Pass, 1992). Bu çalışmada bilişsel yükün 3 alt boyutu ayrı ayrı ele alınmış, her bir boyutta 9'lu derecelendirme bilişsel yük ölçeğinin kullanıldığı öznel ölçme tekniği temel alınmıştır. Haftalık olarak uygulanan ölçekte öğrencilerin kendi zihinsel süreçlerini değerlendirdikleri 5 soru yer almıştır. 10 hafta boyunca haftalık içsel, dışsal ve etkili bililsel yük ölçümlerinin yapıldığı bu çalışmadan elde edilen sonuçların ileriki çalışmalara zengin veri sağlaması açısından değerli olduğu düşünülmektedir.

## **Bölüm Özeti**

21. yüzyıl öğrenci becerileri olarak tanımlanan birey özellikleri, donanımlı öğrenci, dijital vatandaş, bilgi üreten, yenilikçi ve tasarımcı, bilgi işlemsel düşünebilen, yaratıcı ve küresel işbirliğine açık olan birey şeklinde tanımlanmaktadır. Birçok sosyal platform tarafından üst düzey düşünme becerisi olarak gösterilen ve 21. yüzyıl öğrenci becerilerinden biri olarak kabul edilen algoritmik düşünme becerisi, birçok ülke tarafından müfredata koyulan algoritma ve programlama konularını kapsayan dersler ve kurslar vasıtasıyla öğrencilere kazandırılmaya çalışılmaktadır. Mevcut olan bir problemin, seçilen bir programlama aracılığıyla kodlanarak bir çözüme kavuşturulması olarak ifade edilen programlama işleminin yapılabilmesi için çeşitli programlama dilleri geliştirilmiştir. Bu dillerin birbirinden farklı sözdizimi ve anlam bilgisi olsa da aynı amaç için tasarlanmışlardır. Bunlardan biri olan Python programlama dili dünyada ve ülkemizde en çok kullanılan programlama dilleri arasında bulunmakta ve MEB tarafından derslerde temel alınması önerilmektedir. Bu nedenle çalışmada Python programlama dili kullanılarak etkinlikler yapılmıştır.

Programlama dilleri çok işlevsel kütüphaneler olsada, programlama öğretiminde yaşanan güçlükler alanyazında vurgulanmaktadır. Bu durumun programlama dillerinin yapısından ve öğretim şeklinden kaynaklandığı düşünülmektedir. Bu nedenle programcılar programlama öğretimi için yeni arayışlara girmiş ve çeşitli blok tabanlı programlama araçları geliştirilmiştir. Bu araçlar, programlama dillerinin sözdizimi ve anlambilimini kullanıcılara aktarmak için birden fazla boyut kullanırlar. Blok tabanlı programlama araçlarından biri olan Blockly, Google firması tarafından 2012 yılında kullanılmaya başlanmış ve 2017 yılında aktif

gelişime açık hale getirilmiştir. Blockly diğer araçlar gibi blok, kod ve çalışma alanı olarak üç ana bölümden oluşmaktadır. Blockly’i diğer araçlardan ayıran en önemli özelliği, çalışılan programlama dilinin kendi sözdiziminde kullanılabilmesi ve blok yapılarının değiştirilebilmesidir. Blockly programlama aracının avantajlarından biri de Python programlama dilini desteklemesidir. Bu özellikleriyle MEB tarafından da programlama öğretiminde kullanılması önerilmektedir. Bu nedenle çalışmada Blockly programlama aracı tercih edilmiştir. Uygulama öncesinde Blockly içerisinde yer alan kod blokları Python programlama diline uygun hale getirilerek blokların Python programlama dili sözdizimi yapısıyla farklı olmasının önüne geçilmek istenmiştir. Alanyazında blok tabanlı programlama araçlarıyla yapılan çalışmalar incelendiğinde taşıdığı özelliklere rağmen Blockly aracının çok kullanılmadığı görülmüştür.

Bilgisayar kullanımının artmasıyla “bilgisayar kaygısı” kavramı ortaya çıkmıştır. Bilgisayar temelli bir öğrenme süreci gerektiren programlama öğretiminde de öğrencilerin programlama kaygısı yaşamaları olasıdır. Bilgisayar algısı, olumsuz algı, karmaşık sözdizimi, kalıplaşmış düşünceler, yabancı dil kullanımı, programlamada kullanılan yapılar ve program tasarımı gibi programlama kaygısını tetikleyen etmenler olduğu alanyazında vurgulanmaktadır. Buna rağmen incelenen araştırmalarda bu değişkene çok fazla yer verilmediği görülmüştür. Bu nedenle çalışmada programlama etkinliklerinin blok tabanlı olduğu ve olmadığı gruplarda programlama kaygısı incelenmiştir.

Programlama dillerinin doğası gereği karmaşık işlemlerin fazlalığı nedeniyle programlama öğretim süreci öğrencilerin zihinsel yapılarında zorlanmaya ve yüke neden olabilir. Ancak alanyazında bilişsel yük değişkenine değinen çok az araştırmaya rastlanmıştır. Mevcut çalışmalarda ise bilişsel yük sadece tek bir boyut olarak ele alınmış alt boyutları irdelenmemiştir. Bu nedenle bu çalışmada programlama etkinliklerinin blok tabanlı olduğu ve olmadığı gruplarda öğrencilerin bilişsel yüklenmeleri alt boyutlar (içsel, dışsal ve etkili bilişsel yük) da dikkate alınarak incelenmiştir.

## ÜÇÜNCÜ BÖLÜM

### Yöntem

Bu bölümde çalışmada yer alan araştırma deseni, çalışma grubunun seçimi ve veri toplama araçlarının tanıtımı, etkinliklerin tasarımı, uygulama ve veri toplama süreci, araştırma sorularının cevaplanmasında kaynak oluşturacak verilerin analizi, araştırmacının rolü ve araştırmanın geçerlilik ve güvenilirliği detaylı olarak açıklanmıştır.

#### Araştırma Deseni

Araştırmaların geçerli ve güvenilir bir şekilde tamamlanması için önemli unsurlardan bir de temel alınacak araştırma yöntemidir (Fraenkel, Wallen & Hyun, 2012). Temel alınan nitel ve nicel her bir araştırma yönteminin güçlü ve zayıf yönleri bulunmaktadır. Araştırmaların tasarım sürecinde bu durum göz önüne alınmalıdır. Bu nedenle her iki araştırma yönteminin bir arada ele alındığı karma araştırma yöntemleri de son zamanlarda sıklıkla kullanılmaktadır. Karma yöntemin en önemli özelliği, nitel ve nicel araştırma yöntemlerinin güçlü yönlerini birleştirerek, karmaşık süreçlerde zengin verinin elde edilmesini, verilerin karşılaştırılmasını, aralarındaki ilişkinin görülmesini kolaylaştırarak araştırmanın anlaşılabilirliğini artırmasıdır (Creswell & Creswell, 2017).

Bu çalışmada karma araştırma yöntemi kapsamındaki eşzamanlı çeşitleme deseni kullanılmıştır (Creswell, 2003). Çeşitleme deseninde aralarında herhangi bir öncelik sırası olmadan nitel ve nicel veriler toplanır, elde edilen zengin veriler analiz edilirken birbirleriyle ilişkili karmaşık süreçler birlikte yorumlanabilir. Bu çalışmada da araştırma sorularına yeterli düzeyde cevap bulmak, nitel ve nicel veri türleri kullanılarak yapılan analiz sonuçlarını bir bütün olarak değerlendirmek amacıyla çeşitleme deseni kullanılmıştır. Yapılan bu veri çeşitlemesiyle çalışmanın geçerlik ve güvenilirliği artırılmaya çalışılmıştır. Ancak karma araştırmalar zaman alıcı olduklarından araştırma süreci iyi planlanmalıdır (Johnson & Christensen, 2004). Bu çalışmanın nicel boyutunda 1. 2. ve 3. araştırma sorularına cevap bulmak amacıyla blok tabanlı uygulama yapılan ve yapılmayan grupların karşılaştırılması için ön test-sontest yarı deneysel desen, 4. araştırma sorusunda yer alan değişkenler arasında ilişki olup olmadığını belirlemek amacıyla ilişkisel desen kullanılmıştır (McMillan, & Schumacher, 2010). Çalışmanın nitel boyutunda ise 5. araştırma sorusunda belirtilen faktörleri ortaya çıkarmak için durum çalışması yapılmış (Yıldırım & Şimşek, 2006), nicel ve nitel verilerden

elde edilen sonuçlar birlikte yorumlanmıştır. Çalışmada araştırma sorularına göre başvuru alan araştırma desenleri, veri toplama araçları ve veri analiz teknikleri Tablo 4’de özetlenmiştir.

Tablo 4. *Araştırma Sorularına Göre Araştırma Desenleri, Veri Toplama Araçları ve Veri Analiz Teknikleri*

<b>Karma Yöntem (Eş Zamanlı Çeşitleme Deseni)</b>	<b>Araştırma Soruları</b>	<b>Veri Toplama Araçları</b>	<b>Veri Analiz Teknikleri</b>
<b>Yarı Deneysel Desen</b>	1	Kaygı Ölçeği	Betimsel Analiz/ Ortalama, Standart Sapma Kestirimsel Analiz/Bağımsız Gruplar t-Testi
	2	Bilişsel Yük Ölçeği	Betimsel Analiz/ Ortalama, Standart Sapma Kestirimsel Analiz/Tek Yönlü MANOVA
	3	Programlama Başarı Testi	Betimsel Analiz/ Ortalama, Standart Sapma Kestirimsel Analiz/Bağımsız Gruplar t-Testi
<b>İlişkisel Desen</b>	4	Kaygı Ölçeği + Bilişsel Yük Ölçeği + Programlama Başarı Testi	İlişkisel (Betimsel + Kestirimsel Analiz)/Pearson Çoklu Korelasyon Testi
<b>Durum Çalışması</b>	5	Yarı Yapılandırılmış Görüşme Rehberi	İçerik Analizi
<b>Durum Çalışması</b>	6	Yarı Yapılandırılmış Görüşme Rehberi	İçerik Analizi

### Çalışma Grubu

Çalışma grubu, Antalya ili Alanya ilçesi Türkler İMKB Sosyal Bilimler Lisesi’nde 2017-2018 eğitim öğretim yılı II. döneminde 4 farklı hazırlık sınıfında öğrenim gören 70 erkek, 20 kız öğrenci olmak üzere toplam 90 öğrenciden oluşmaktadır. Örneklem belirleme sürecinde seçkisiz olmayan örnekleme yöntemlerinden amaçlı örnekleme kullanılmıştır. Amaçlı örnekleme, araştırmanın amacı kapsamında inceleme yapabilmek için veri açısından zengin örneklem kitlelerinin seçilmesidir (Büyüköztürk, Çakmak, Akgün, Karadeniz & Demirel, 2017). Ayrıca amaçlı örnekleme zaman ve maliyet açısından sınırlı olan çalışma ortamlarında konuya özel olan kitlelerin varlığında, büyük oranda genelleme imkanı sunmasıyla alan yazında geniş yer bulmaktadır (McMillan & Schumacher, 2010). Çalışma grubunu oluşturan kitlenin eğitim seviyesi ve uygulamanın yapıldığı ders ortamının belirli ortaöğretim kurumlarında olması nedeniyle katılımcıların belirlenmesinde amaçlı örnekleme yöntemine başvurulmuştur. Amaca uyan lise hazırlık sınıfları herhangi bir kıstas gözetilmeksizin rastgele olarak Python tabanlı etkinlik sınıfları (kontrol) ve Blockly tabanlı etkinlik sınıfları (deney) olmak üzere iki gruba ayrılmıştır. Buna göre 2 sınıf kontrol grubunu

oluştururken, diğer 2 sınıf ise deney grubunu oluşturmuştur. Araştırmanın başlangıcında çalışma grubu 101 kişi olarak tasarlanmıştır. Ancak uygulama sürecinde ortaya çıkan öğrenci devamsızlıkları ve veri toplama araçlarının her öğrenciye uygulanamaması nedeniyle sayı 90 kişiye düşürülmüştür. Veri toplama araçlarına göre gruptaki öğrenci sayıları Tablo 5’de gösterilmiştir.

Tablo 5. *Veri Toplama Araçlarına Göre Öğrenci Sayıları*

	Blok Tabanlı Etkinlik Grubu			Python Tabanlı Etkinlik Grubu			Genel Toplam
	Kız	Erkek	Toplam	Kız	Erkek	Toplam	
<b>Ölçek ve Başarı Testi Uygulanan Tüm Örneklem</b>	36	8	44	34	12	46	90
<b>Görüşme Yapılan Örneklem</b>	7	3	10	7	2	9	19

Tablo 5’de görüldüğü üzere Blok tabanlı etkinlik grubunda (BTEG) 44, Python tabanlı etkinlik grubunda (PTEG) ise 46 olmak üzere toplamda 90 hazırlık sınıfı öğrencisine ölçek ve başarı testi uygulanmıştır. Uygulama sonrasında ise blok tabanlı etkinlik grubunda 10, Python tabanlı etkinlik grubunda ise 9 öğrenciyle görüşme yapılmıştır. Ayrıca görüşme rehberinin geliştirilmesi sürecinde 4 öğrenciye anlaşılabilirliğini test etmek için görüşme soruları incelenmiştir.

Çalışma gruplarına etki edecek herhangi bir sebep araştırmanın geçerli ve güvenilir olmaktan uzaklaşmasına neden olabilir (Fraenkel & Wallen, 2006). Bu çalışmada da grupların benzer olup olmadığı test edilmiştir. Öğrencilerin ortaokulda zorunlu bilişim teknolojileri ve yazılımı dersi müfredatında yer alan temel algoritma becerisine yönelik konuları işlemiş oldukları ancak 2017-2018 öğretim yılıyla birlikte ilk kez lise hazırlık sınıfına Bilgisayar Bilimi dersinin zorunlu olarak dahil edilmesi nedeniyle programlama konusunda herhangi bir ders almamış oldukları göz önüne alındığında grupların benzer özellik gösterdiği söylenebilir. Ayrıca çalışma gruplarının benzerliğini test etmek amacıyla, uygulanan başarı ön testi sonrasında bağımsız gruplar t-testi analizi yapılarak gruplar arasında programlama başarı durumları açısından farklılık olmadığı, grupların benzer özellik gösterdiği Tablo 6’da sunulmuştur.

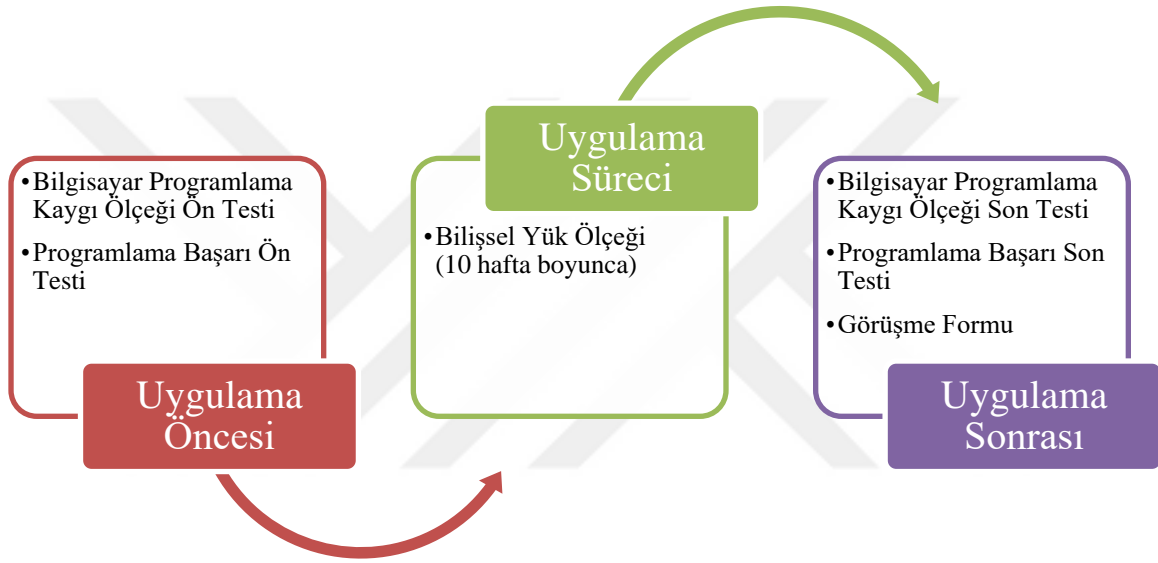
Tablo 6. *Gruplara Göre Ön-Test Programlama Başarı Düzeyleri Arasındaki Fark*

Grup	N	$\bar{X}$	S	sd	t	p
PTEG	46	30.08	13.11	88	-.723	.472
BTEG	44	32.09	13.18			

Tablo 6'ya göre gruplar arasında ön test programlama başarı düzeyleri arasında anlamlı bir fark ortaya çıkmamıştır ( $t(88) = .472, p < .05$ ). Bu sonuca göre her iki grubun da öğretimi yapılacak programlama dili açısından denk iki grup olduğu varsayılmıştır.

### Veri Toplama Araçları

Araştırmada veri toplama aracı olarak; bilgisayar programlama kaygı ölçeği (BPKÖ), bilişsel yük ölçeği (BYÖ), programlama başarı testi (PBT) ve görüşme formu kullanılmıştır. Devam eden bölümde bu araçlar hakkında bilgi verilmiş, her iki gruptaki veri toplama süreci ise Şekil 11'de özetlenmiştir.



Şekil 11. Veri toplama süreci.

#### Bilgisayar programlama kaygı ölçeği.

Öğrencilerin programlama kaygılarını belirlemek amacıyla Cheung (1990) tarafından geliştirilen Demir (2015) tarafından Türkçe'ye uyarlanan Bilgisayar Programlama Kaygı Ölçeği kullanılmıştır. Ölçek "Hiçbir zaman doğru değil", "Nadiren doğru", "Bazen doğru", "Sıklıkla doğru", "Her zaman doğru" şeklinde ifade edilen 5'li likert tipi ölçek 19 adet maddeden oluşmaktadır (Bkz. Ek-1). Demir (2015) ölçeğin Türkçe'ye uyarlanmasında 6 kişiden oluşan Türkçe ve İngilizce dil uzmanlarının gerekli çevirileri yaparak ölçeğe son şeklini verdiklerini, sonrasında ise geçerlilik ve güvenirlik testlerinin uygulandığını belirtmektedir. Güvenirlik madde analizinde alt ve üst grup %27 grup ortalamaları alınarak madde analizi gerçekleştirilmiştir. Demir (2015) tarafından, 116 kişilik pilot grubuna uygulanan 19 maddelik ölçeğin iç tutarlık katsayısı olan Cronbach Alpha ( $\alpha = .963$ ) olarak hesaplanmıştır. Ölçeğin geçerliliği için yaptığı pilot çalışmada Kaiser-Meyer-Olkin katsayısı .911 olarak hesaplanırken Bartlett'in Sphericity testi (Approx Chi-Square = 1095,848)  $p <$



0.001 düzeyinde anlamlı bulunmuştur. Bu incelemelerin ardından Demir (2015) ölçeği geçerli ve güvenilirlik olarak nitelendirmiştir. Bu çalışma kapsamında ise BPKÖ araştırmacı tarafından çalışma öncesinde 75 ortaöğretim öğrencisine pilot olarak uygulanmış ve Cronbach Alpha değeri  $\alpha = .963$  olarak ortaya çıkmıştır. Bu değer bilgisayar programlama kaygı ölçeğinin güvenilir bir ölçme aracı olduğunu göstermektedir.

### **Bilişsel yük ölçeği.**

Çalışmada Paas (1992) tarafından oluşturulan 9'lu derecelendirmeye sahip bilişsel yük ölçeği temel alınmıştır. Cronbach Alpha iç tutarlılık katsayısı tarafından  $\alpha = .90$  olarak hesaplanmıştır. Ölçeğin Türkçe'ye uyarlanması Kılıç ve Karadeniz (2004) tarafından yapılmış ve Cronbach Alpha iç tutarlılık katsayısı  $\alpha = 0.78$  olarak belirlenmiştir. Ölçek “çok çok az”, “çok az”, “az”, “kısmen az”, “ne az ne fazla”, “kısmen fazla”, “fazla”, “çok fazla”, “çok çok fazla” olmak üzere 9 derecelendirmeye sahiptir. Bir öğrencinin en fazla alabileceği ortalama puan 9 iken en az alabileceği ortalama puan 1'dir. Ortalama puanı 5'in altında olan puanlar “düşük bilişsel yüklenme”, ortalama puanı 5'in üzerinde olan puanlar ise “yüksek bilişsel yüklenme” durumuna sahip olarak kabul edilmektedir.

Bu çalışmada ise Paas (1992)'ın 9'lu derecelendirme ölçeği temel alınarak Gerjets, Scheiter, Opfermann, Hesse ve Eysink (2009) tarafından toplam bilişsel yük ölçümünün yanı sıra içsel, dışsal ve etkili bilişsel yük olarak 3 alt boyutla geliştirilen bilişsel yük ölçeği kullanılmıştır (Bkz. Ek-2). Her iki gruba farklı programlama araçlarıyla yapılan etkinliklerde öğrencilerin öğrenmek için sarf ettikleri çabayı içsel, dışsal ve etkili bilişsel yük açısından ayrı ayrı ölçmek amacıyla bu bilişsel yük ölçeği tercih edilmiştir. Ölçek, Yılmaz (2012) tarafından Türkçe'ye çevrilmiş, dil uzmanlarına gösterilmiş, kapsam geçerliliği için alan uzmanlarına inceletilmiş ve 34 kişilik üniversite öğrencisinden oluşan bir gruba 10 hafta boyunca uygulanarak, pilot uygulaması yapılmıştır. Sonrasında her bir alt boyut için güvenilirlik katsayısı ayrı ayrı hesaplanmıştır. İçsel bilişsel yük için katsayı  $\alpha = 0.73$ , dışsal bilişsel yük için katsayı  $\alpha = 0.80$  ve etkili bilişsel yük için katsayı  $\alpha = 0.72$  olarak elde edilmiştir. Son hali verilen bilişsel yük ölçeği 3 alt boyuttan ve toplam 5 sorudan oluşmakta olup her bir boyut farklı sorularla ölçek içerisinde ölçülmektedir. İçsel bilişsel yük için tek bir soruda konunun kendi doğasından kaynaklı değerlendirmeler yapılmış bunun sonucunda ortaya çıkan ortalama haftanın içsel bilişsel yük ortalamasını oluşturmuştur. Dışsal bilişsel yükü tespit edebilmek içinse 3 soru sorulmuş her bir sorunun grup ortalaması alınmış, sonrasında grup ortalamaları hafta ortalamasını oluşturmuştur. Etkili bilişsel yükte ise sorulan soru derste yapılan örnekleri ilgilendirdiğinden öğrencilere verilen her örnek için puanlama yapılmış, bu puanların ortalamaları her öğrenci için o haftanın etkili bilişsel yükünü

oluşturmuştur. Ölçekte yer alan sorular ve hangi alt boyuta ait oldukları Tablo 7’de gösterilmiştir.

Tablo 7. *Bilişsel Yük Ölçeğindeki Sorular ve Ait Oldukları Alt Boyutlar*

<b>Alt Boyut</b>	<b>İçerdiği Soru</b>
<b><i>İçsel Bilişsel Yük</i></b>	Aşağıdaki her bir konu şu anda sizin için ne kadar zor gelmektedir? Bu dersin öğrenme ortamında önemli ve önemsiz bilgileri ayırt etmek ne kadar zordur?
<b><i>Dışsal Bilişsel Yük</i></b>	Dersin öğrenme ortamı sizin için ne kadar zordur? Öğrenme ortamında ihtiyacınız olan tüm bilgileri toplamak sizin için ne kadar zordur?
<b><i>Etkili Bilişsel Yük</i></b>	Derste yapılan en son örneği takip etmek için sarf ettiğiniz çabayı aşağıda gösterilen ölçek üzerinde belirtiniz.

Bu ölçek asıl uygulamaya başlamadan önce, Bayburt Fen Lisesi’nde öğrenim gören çalışmanın yapılacağı grupla aynı müfredatı içeren aynı dersi almış aynı öğretim seviyesindeki 80 öğrenciye 1 haftalık müfredatın işlenmesi sonucu pilot olarak uygulanmıştır. Pilot uygulama yapıldıktan sonra her bir alt boyut ve tüm ölçek için Cronbach Alpha iç tutarlılık katsayısı hesaplanmıştır. Buna göre içsel bilişsel yük için  $\alpha=.929$ , dışsal bilişsel yük için  $\alpha=.942$ , etkili bilişsel yük için  $\alpha=.977$  ve ölçeğin tamamı için  $\alpha=.969$  olarak hesaplanmıştır. Bu sonuca göre veri toplama aracının yeterli düzeyde güvenilirliğe sahip olduğu söylenebilir.

Bilişsel yük ölçeği her haftaki uygulama sonunda o haftanın kazanımları ölçekte yer alacak şekilde uygulanmıştır. Uygulama 10 hafta boyunca devam etmiş olup her bir katılımcıya 10 kez bilişsel yük ölçeği uygulanmıştır. Her hafta uygulanan bilişsel yük ölçeğinin içsel, dışsal ve etkili bilişsel yük ortalamaları ayrı ayrı hesaplanmış 3 alt boyutun ortalaması alınarak ölçek ortalaması belirlenmiştir. Ayrıca uygulama sonunda hem 3 alt boyutun 10 haftalık ayrı ayrı ortalamaları alınmış, hem de tüm boyutların 10 haftalık ortalaması alınarak her katılımcının bilişsel yük ortalaması oluşturulmuştur.

#### **Programlama başarı testi.**

Çalışmada katılımcıların programlama dili kazanımlarını edinme düzeylerini ölçmek amacıyla araştırmacı tarafından Python programlama başarı testi oluşturulmuştur. MEB’in Bilgisayar Bilimi dersi yeni öğretim programındaki (TTKB, 2017) programlama dili hedef kazanımların testte temsil edilmesine özen gösterilmiştir. Başarı testindeki soruların kazanımlara göre dağılımı Tablo 8’de gösterilmiştir.

Tablo 8. Programlama Başarı Testi Sorularının Kazanımlara Göre Dağılımı

Kazanım	Soru
Belirlenen programlama dilinde değişkenleri ve sabitleri uygun şekilde kullanarak programlar geliştirir.	1,2,3,4
Belirlenen programlama dilinde girdilerin belirlendiği ve çıktılarının gözlemlendiği programlar geliştirir.	2,3,4,5,6,7,8
Belirlenen programlama dilinde kontrol yapılarını kullanarak programlar geliştirir.	9,10,11,12,13,16,19,20
Belirlenen programlama dilinde tekrarlı yapıları kullanarak programlar geliştirir.	13,14,15,16,21,22
Belirlenen programlama dilinde parametre almayan fonksiyon içeren programlar geliştirir.	16
Belirlenen programlama dilinde parametre alan fonksiyon içeren programlar geliştirir.	3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25
Belirlenen programlama dilinde değer döndüren fonksiyon içeren programlar geliştirir.	18,20
Tanımladığı dizgi tipindeki veriye ait temel fonksiyonların yer aldığı programlar geliştirir.	21
Belirlenen programlama dilinde dizi tipinde veri tanımlar.	21,22,23,24,25
Tanımladığı dizi tipindeki veriye ait temel fonksiyonların yer aldığı programlar geliştirir.	22,24,25

Başarı testi oluşturulurken alan yazındaki programlama başarı testleri de incelenmiş ve hazırlanan taslak form 25 maddeden oluşmuştur. 2 Bilişim Teknolojileri öğretmenine ve 1 Öğretim Teknolojileri uzmanına incelenmiştir. Soruların anlaşılabilirliğini denetlemek için örneklem dışında fakat örneklem düzeyiyle aynı öğretim seviyesinde 1 kız ve 1 erkek öğrenciye sorular okutulmuştur. Gerekli düzeltmeler yapılan test Türkçe dil uzmanına da gösterilmiştir.

Ölçeğin kapsam geçerliliğini artırmak için alt ve üst gruplar % 27 grup ortalamaları madde analizi ile madde güçlük ve madde ayırt edicilik indeksleri hesaplanmıştır (Büyüköztürk vd., 2017; Şencan, 2005, s. 746). Sonuçlar Tablo 9’da sunulmuştur.

Tablo 9. Programlama Başarı Testi Madde Analizleri

Madde Numarası	Madde Güçlük İndeksi (p)	Madde Ayırt Edicilik İndeksi (D)	Açıklama
1.Madde	0.09	0.19	Zor ve düşük düzeyde ayırt edici madde
2.Madde	0.87	0.26	Kolay fakat orta düzeyde ayırt edici madde
3.Madde	0.56	0.59	Orta güçlükte ve yüksek düzeyde ayırt edici madde
4.Madde	0.67	0.30	Kolay ve yüksek düzeyde ayırt edici madde
5.Madde	0.43	0.41	Orta güçlükte ve yüksek düzeyde ayırt edici madde
6.Madde	0.65	0.56	Kolay ve yüksek düzeyde ayırt edici madde
7.Madde	0.54	0.48	Orta güçlükte ve yüksek düzeyde ayırt edici madde

Tablo 9. (Devamı)

8.Madde	0.41	0.30	Orta güçlükte ve yüksek düzeyde ayırt edici madde
9.Madde	0.37	0.52	Zor ve yüksek düzeyde ayırt edici madde
10.Madde	0.33	0.37	Zor ve yüksek düzeyde ayırt edici madde
11.Madde	0.74	0.52	Kolay ve yüksek düzeyde ayırt edici madde
12.Madde	0.63	0.59	Kolay ve yüksek düzeyde ayırt edici madde
13.Madde	0.24	0.33	Zor ve yüksek düzeyde ayırt edici madde
14.Madde	0.81	0.37	Kolay ve yüksek düzeyde ayırt edici madde
15.Madde	0.33	0.44	Zor ve yüksek düzeyde ayırt edici madde
16.Madde	0.39	0.33	Zor ve yüksek düzeyde ayırt edici madde
17.Madde	0.57	0.41	Orta güçlükte ve yüksek düzeyde ayırt edici madde
18.Madde	0.78	0.22	Kolay fakat orta düzeyde ayırt edici madde
19.Madde	0.85	0.22	Kolay fakat orta düzeyde ayırt edici madde
20.Madde	0.67	0.37	Kolay ve yüksek düzeyde ayırt edici madde
21.Madde	0.81	0.37	Kolay ve yüksek düzeyde ayırt edici madde
22.Madde	0.48	0.44	Orta güçlükte ve yüksek düzeyde ayırt edici madde
23.Madde	0.87	0.26	Kolay fakat orta düzeyde ayırt edici madde
24.Madde	0.43	0.11	Orta güçlükte ve düşük düzeyde ayırt edici madde
25.Madde	0.37	0.44	Zor ve yüksek düzeyde ayırt edici madde

Tablo 9 incelendiğinde 1. ve 24. maddelerin ayırt edicilik indekslerinin 0,20 değerinin altında olduğundan düşük düzeyde ayırt edici madde olduğu görülmektedir (Başol vd., 2013; Büyüköztürk vd., 2017). Ancak 1. sorunun değişkenler konusunda, 24. sorunun ise dizi elemanı ekleme ve silme konularında önemli sorular olduğu düşünüldüğünden BT alan uzmanlarının da görüşü alınarak her iki soru da kapsam geçerliliği açısından araştırmacı tarafından başarı testinin içerisinde tutulmuştur.

Programlama başarı testinin güvenilirliğini tespit etmek amacıyla maddeler arası iç tutarlığı için KR-20 güvenilirlik katsayısı hesaplanmıştır (KR-20= .71 > .70). Böylece 25 maddeden oluşan programlama başarı testinin geçerli ve güvenilir bir ölçme aracı olduğu belirlenmiştir. Testteki her soru 4 puan üzerinden değerlendirilken testten alınabilecek en yüksek puan 100'dür. Her iki grup için uygulama öncesi ve sonrasında aynı test kullanılmıştır. Testte sadece soruların gösterimi gruplara göre farklılık göstermektedir. BTEG testinde Blockly görsel programlama aracıyla soru gösterimi yapılırken (Bkz. Ek-3), PTEG metin tabanlı Python programlama dili sözdizimine göre soru gösterimi yapılmıştır (Bkz. Ek-4).

### Görüşme formu.

Programlama dili öğretimi sürecinde kaygı, bilişsel yük ve akademik başarıyı etkileyen faktörleri tespit etmek ve değişkenler arasındaki ilişkileri detaylı inceleyebilmek

amacıyla yarı yapılandırılmış görüşme formu hazırlanmış, her iki gruptaki öğrencilerle görüşmeler gerçekleştirilmiştir (Bkz. EK-5). Görüşme tekniği, katılımcıların doğrudan gözlemlenemediği durumlarda, araştırmacı kontrolünde, daha önceden hazırlanmış, amaçlı soruların yol göstericiliğinde, katılımcının düşüncelerini ortaya çıkarmayı amaçlayan, araştırılan konu hakkında, tarihsel bilgi de sağlanabilen nitel araştırma tekniğidir (Creswell & Creswell, 2017). Ayrıca araştırmacı, sorularına anında yanıt alıp dönüt verebildiği için, katılımcı araştırmanın karmaşık yapısından sıyrılabilir (Büyüköztürk *vd.*, 2017). Çalışma amacına uygun alanyazından yararlanılarak hazırlanan taslak form 2 Bilişim Teknolojileri öğretmenine ve 1 Öğretim Teknolojileri uzmanına incelenmiştir. Soruların anlaşılabilirliğini denetlemek için örneklem dışındaki 1 kız ve 1 erkek öğrenciye sorular okutulmuş ve gerekli düzeltmeler yapılmıştır. Pilot uygulama yapılan grup içerisinde 2 kız ve 2 erkek öğrenciyle görüşme yapılmış, sorulardaki eksikliklerin tespit edilmesi amaçlanmıştır. Her soruda sadece bir odak noktasına bakılmış, derinlemesine cevaplar alabilmek amacıyla görüşme formu sonda sorularla zenginleştirilmiştir (Büyüköztürk *vd.*, 2017; Yıldırım, 2010, Creswell & Creswell, 2017). Türkçe dil uzmanının da görüşleri alınan forma son şekli verilmiştir. Python tabanlı etkinlik grubundan 7 kız ve 2 erkek olmak üzere 9 öğrenci, Blockly tabanlı etkinlik grubundan 7 kız ve 3 erkek olmak üzere 10 öğrenci olmak üzere toplamda 19 öğrenciyle görüşme yapılmıştır. Görüşme yapılan öğrencilerin kendilerini daha rahat hissederek sorulara samimi cevaplar verebilmeleri için görüşmeler ders saatleri dışında gerçekleştirilmiştir. Her iki gruptan rastgele seçilen öğrencilerle yapılan görüşmeler veri kaybının en az olması, geçerli ve güvenilir olması açısından katılımcıların da izni alınarak ses kayıt cihazı ile kaydedilmiş daha sonra ses kayıtları yazıya çevrilmiştir.

### **Veri toplama araçlarının geçerliliği ve güvenilirliği**

Araştırmalarda yapılan ölçümün temel amacı, ulaşılmaya çalışılan olguyu ve onu etkileyen etmenler arasındaki ilişkiyi somut olarak gösterebilmektir. Veri toplama aracının ölçümü doğru yapması, ölçümün tekrarlanabilirliği ve genellenebilirliği önemli ve gereklidir. Bu gereklilik için veri toplama araçları, niteliklerine göre, geçerlilik ve güvenilirlik aşamalarından geçmelidir (Fraenkel *vd.*, 2012; Şencan, 2005).

Geçerlilik, araştırmacının ölçmek istediği unsurların, hiçbir müdahale olmadan ne kadar doğru ölçtüğüyle ilgilidir. Araştırmacının amaçlanan ölçmeyi gerçekleştirebilme oranıdır (Büyüköztürk *vd.*, 2017; McMillan & Schumacher, 2010). Veri toplama araçlarını oluşturan maddelerin, ölçülmek istenen değişkenleri ne kadar temsil ettiği, örneklediği kapsam geçerliliğini oluşturmaktadır (Fraenkel *vd.*, 2012; Şencan, 2005). Güvenirlik, veri toplama aracının ölçme sonuçlarının hatalardan temizlenme derecesi, duyarlılığı, kararlılığı ve

tutarlığı olarak ifade edilmektedir (Büyüköztürk vd., 2017). Veri toplama araçlarının ne derece güvenilir olduğu mevcut bazı yöntemlerle belirlenebilmektedir.

Bu araştırmanın geçerliği ve güvenilirliğinin yüksek olması için veri toplama araçları üzerinde çeşitli istatistiksel yöntemlerle testler uygulanmış, maddelerin değişkenleri temsil etme oranı ve hata ortaya çıkma ihtimali en aza indirilmeye çalışılmıştır. Buna göre çalışmada, kullanılan hazır veri toplama araçlarının geçerlik ve güvenilirliği testinden geçmiş araçlar olmasına dikkat edilmiştir. Araştırmacı tarafından geliştirilen veri toplama araçlarının ise kapsam geçerliliğini ortaya çıkarmak için, ilgili alanyazın incelenmiş, 2 Bilişim Teknolojileri öğretmeni ve 1 Öğretim Teknolojileri uzmanından görüşler alınmış, gerekli düzeltmeler yapılmıştır. Veri toplama araçlarının niteliğine göre güvenilirliğini belirlemek için ise Kuder-Richardson (KR-20) ve Cronbach Alpha güvenirlik katsayıları hesaplanmıştır. Çalışmada kullanılan veri toplama araçlarıyla ilgili alınan geçerlik güvenirlik önlemleri Şekil 12’de özetlenmiştir.

<b>Bilgisayar Programlama Kaygı Ölçeği</b>	<ul style="list-style-type: none"><li>• <b>GEÇERLİLİK</b></li><li>• Programlama kaygısıyla ilgili alan yazındaki ölçek ve çalışmalar incelenmiştir.</li><li>• Geçerliliği ve güvenilirliği test edilmiş hazır ölçek kullanılmıştır.</li><li>• Hazır ölçeğin çalışmanın kapsamına ve örnekleme uygunluğu 2 BT öğretmeni ve 1 öğretim üyesi tarafından incelenmiştir.</li><li>• Ölçek maddelerinin anlaşılır olup olmadığı 1 Türkçe uzmanına kontrol ettirilmiştir.</li><li>• Örneklem düzeyiyle aynı seviyede 75 öğrenci ile pilot uygulama yapılmıştır.</li><li>• <b>GÜVENİRLİK</b></li><li>• Cronbach Alpha iç tutarlılık katsayısı hesaplanmıştır.</li><li>• Pilot uygulama yapılmıştır.</li></ul>
<b>Bilişsel Yük Ölçeği</b>	<ul style="list-style-type: none"><li>• <b>GEÇERLİLİK</b></li><li>• Bilişsel yük ve ölçekleri ile ilgili alan yazında inceleme yapılmıştır.</li><li>• Geçerliliği ve güvenilirliği test edilmiş hazır ölçek kullanılmıştır.</li><li>• Hazır ölçeğin çalışmanın kapsamına ve örnekleme uygunluğu 2 BT öğretmeni ve 1 öğretim üyesi tarafından incelenmiştir.</li><li>• Ölçek maddelerinin anlaşılır olup olmadığı 1 Türkçe uzmanına kontrol ettirilmiştir.</li><li>• Örneklem düzeyiyle aynı seviyede 80 öğrenci ile pilot uygulama yapılmıştır.</li><li>• Öğrencilerin maddeleri anlayıp anlayamadıkları erkek ve kız öğrenciler tarafından sınanmıştır.</li><li>• <b>GÜVENİRLİK</b></li><li>• Pilot uygulama yapılmıştır.</li><li>• Ölçeğin Cronbach Alpha iç tutarlılık katsayısı hesaplanmıştır.</li></ul>
<b>Programlama Başarı Testi</b>	<ul style="list-style-type: none"><li>• <b>GEÇERLİLİK</b></li><li>• Sorular öğretim yapılacak seviye ve öğretim programındaki kazanımlar dikkate alınarak geliştirilmiştir.</li><li>• Geliştirilen testin çalışmanın kapsamına ve örnekleme uygunluğu 1 Öğretim Teknolojileri uzmanı ve 2 BT öğretmeni tarafından incelenmiştir.</li><li>• Testi maddelerinin anlaşılır olup olmadığı 1 Türkçe uzmanına kontrol ettirilmiştir.</li><li>• Öğrencilerin maddeleri anlayıp anlayamadıkları 1 erkek ve 1 kız öğrenci tarafından sınanmıştır.</li><li>• Soruların madde güçlük ve madde ayırt edicilik indeksi hesaplanmıştır.</li><li>• <b>GÜVENİRLİK</b></li><li>• İç tutarlılık tespiti için KR-20 katsayısı hesaplanmıştır.</li></ul>
<b>Görüşme Formu</b>	<ul style="list-style-type: none"><li>• <b>GEÇERLİLİK</b></li><li>• Görüşme formu alan yazın çerçevesinde tüm değişkenleri kapsayacak şekilde hazırlanmıştır.</li><li>• Sorular öğretim yapılacak seviye dikkate alınarak hazırlanmıştır.</li><li>• Her bir sorunun yalnızca bir yargıyı içerip içermediği kontrol edilmiştir.</li><li>• Soruların çalışma değişkenler çerçevesinde olup olmadığı ve çalışma kapsamına uygunluğu 1 Öğretim Teknolojileri uzmanı ve 2 BT öğretmeni tarafında incelenmiştir.</li><li>• Görüşme formu sorularının anlaşılır olup olmadığı 1 Türkçe uzmanına kontrol ettirilmiştir.</li><li>• Pilot uygulama ile örneklem dışında 1 erkek ve 1 kız öğrenci ile görüşme yapılmıştır.</li><li>• <b>GÜVENİRLİK</b></li><li>• Görüşme sorularının değişkenler çerçevesinde olup olmadığı bir uzmanlar tarafından kontrol edilmiştir.</li><li>• Pilot uygulama ile görüşme sorularının çalışma amacına uygunluğu test edilmiştir.</li></ul>

Şekil 12. Veri toplama araçlarıyla ilgili geçerlilik ve güvenirlik önlemler

## Uygulama Süreci

Uygulama 2017-2018 eğitim-öğretim yılı bahar döneminde Antalya ili Alanya ilçesindeki Türkler İMKB Sosyal Bilimler Lisesi hazırlık sınıfları öğretim programında yer alan Bilgisayar Bilimi dersinde gerçekleştirilmiştir. Uygulama süreci aynı zamanda Bilişim Teknolojileri öğretmeni olan araştırmacı tarafından 2 hazırlık sınıfında 10 hafta boyunca bizzat yürütülmüştür. Bu süreçte deney grubu için öğretim aracı olarak blok tabanlı Blockly programlama aracı, kontrol grubu için ise Phyton programlama dili için tasarlanan metin tabanlı kod editörü kullanılmıştır.

Uygulama öncesinde araştırmacı tarafından Blockly’de yer alan bloklar Python programlama diline göre yeniden düzenlenmiştir. Bilgisayar bilimi dersi öğretim programındaki kazanımlar temel alınarak hazırlanan yıllık plana göre her iki grup için ayrı etkinlik içerikleri geliştirilmiştir (Bkz. EK-8, EK-9). BTEG ve PTEG gruplarına bir hafta boyunca süreç anlatılmış, öğretim süresince kullanılacak olan öğretim araçları hakkında bilgi verilmiştir. Her iki gruba da Programlama Dili Başarı Ön Testi ve Bilgisayar Programlama Kaygı Ölçeği Ön Testi yapılmıştır. Alanya Türkler İMKB Sosyal Bilimler Lisesi’ndeki mevcut bilişim teknolojileri sınıfının teknik altyapısı blok tabanlı öğretim aracı kullanımı için uygun olduğundan uygulama takvimi bilişim teknolojileri sınıfının haftalık ders programına göre düzenlenmiş ve Tablo 10’da gösterilmiştir.

Tablo 10.Uygulama Takvimi

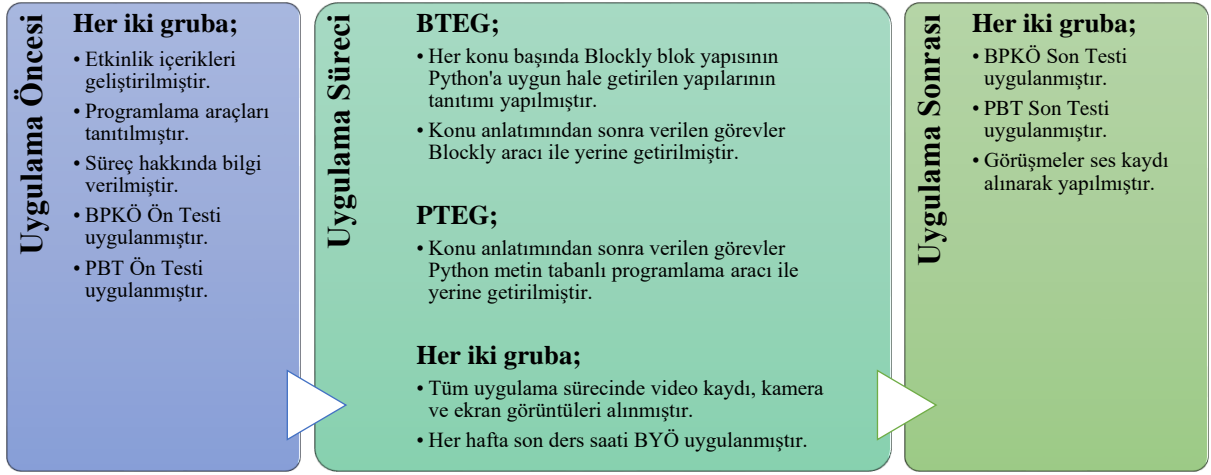
Aylar	Haftalar	Konular (Her iki Grup için )	Kazanımlar
Mart	1.hafta	<b>Değişkenler</b> Sabit Değerler Karışık Türlü İfadeler	Belirlenen programlama dilinde değişkenleri ve sabitleri uygun şekilde kullanarak programlar geliştirir.
	2.hafta	<b>Değişkenler</b> Aritmetik İfadeler Aritmetik Örnekler	Belirlenen programlama dilinde girdilerin belirlendiği ve çıktılarının gözlemlendiği programlar geliştirir.
	3.hafta	<b>Koşul İfadeleri</b> İlişkisel Operatörler İf ifadesi İf/Else ifadesi İç içe koşul yapıları	Belirlenen programlama dilinde kontrol yapılarını kullanarak programlar geliştirir.
	4.hafta	<b>Karar Yapıları</b> Çok yönlü karar ifadeleri Zincirleme durum ifadeleri Koşullu ifadeler	Belirlenen programlama dilinde kontrol yapılarını kullanarak programlar geliştirir.
Nisan	5.hafta	<b>Döngüler</b> Döngü yapıları For döngüsü İç içe döngüler	Belirlenen programlama dilinde tekrarlı yapıları kullanarak programlar geliştirir.
	6.hafta	<b>Döngüler</b> While döngüsü Belirli ve belirsiz döngüler Döngüden çıkma komutları	Belirlenen programlama dilinde tekrarlı yapıları kullanarak programlar geliştirir.

Tablo 10. (Devamı)

Aylar	Haftalar	Konular (Her iki Grup için )	Kazanımlar
<b>Nisan</b>	7.hafta	<b>Fonksiyonlar</b> Fonksiyonlara giriş Fonksiyona parametre tanımlama Standart matematik fonksiyonları Fonksiyon tanımlama	Belirlenen programlama dilinde parametre almayan fonksiyon içeren programlar geliştirir.
	8.hafta	<b>Fonksiyonlar</b> Global değişkenler Fonksiyon sıralaması belirleme Fonksiyon örnekleri	Belirlenen programlama dilinde parametre alan değer döndüren fonksiyon içeren programlar geliştirir.
<b>Mayıs</b>	9.hafta	<b>Listeler</b> Liste kavramı Değerler arası gezinme Liste üyeliği Listeye değer atama ve eşitleme	Belirlenen programlama dilinde dizi tipinde veri tanımlar.
	10.hafta	<b>Listeler</b> Dilimleme Listeden eleman çıkarma Listede kullanılan yöntemler	Tanımladığı dizi tipindeki veriye ait temel fonksiyonların yer aldığı programları geliştirir.

BTEG ve PTEG gruplarının ders programına göre haftanın 2 günü (günde 2 ders saati) olmak üzere 10 hafta süren uygulama süreci her bir grup için toplam 40 saattir. Ders saatlerinde önce öğretmen tarafından haftanın konusu hakkında temel bilgiler verilmiş, sonra konuyla ilgili kazanıma yönelik önceden hazırlanan etkinlikler öğrenciler tarafından yapılmıştır. Her grup kendi programlama öğretim aracını kullanarak etkinlikleri tamamlamıştır. Ayrıca uygulama süresince bilişim teknolojileri sınıfında oluşabilecek bilgisayar arızalarına karşı yedek bilgisayarlar bulundurulmuştur. Her iki grupta da dersler aynı haftalarda, aynı konu ve kazanımlar dikkate alınarak işlenmiştir. Ayrıca her haftanın 4. ders saatinin sonunda her iki gruptaki katılımcılara Bilişsel Yük Ölçeği uygulanmıştır. Tüm uygulama sürecinin bittiği 10. haftanın sonunda her iki gruba da Programlama Başarı Son Testi ve Bilgisayar Programlama Kaygı Ölçeği Son Testi yapılmıştır. Ayrıca BTEG ve PTEG gruplarından rastgele seçilen 19 katılımcı ile görüşmeler gerçekleştirilmiş, bu süreçte ses kayıtları alınmıştır. Uygulama süreciyle ilgili her iki gruptaki öğrenci etkinliklerinden ekran kayıtları ve kamera görüntüleri Ek-6 ve Ek-7’de sunulmuştur. Uygulama sürecinde yapılanlar Şekil 13’te özetlenmiştir.





Şekil 13. Uygulama süreci.

## Veri Analizi

Çalışmada, nicel ve nitel verilerin analiz sürecinde farklı analiz teknikleri takip edilmiştir (Akbulut, 2010; Field, 2009).

### Nicel veri analizi.

SPSS 21 paket programı kullanılarak yapılan nicel veri analizi süreci Şekil 14'te özetlenmiştir.

#### Nicel Verilerin Analize Hazırlanması

- Veri analizinden önce çalışma grubundaki her öğrenciye ID numarası olarak okullarında kullandıkları öğrenci numarası verilmiştir.
- Eksik veri analizi yapılmış, ön test ve son testlerin tümünde yer almayan katılımcıların verileri analize dâhil edilmemiştir.

#### Betimsel Analizler ve Normallik Testi

- Gruplardaki her veri seti için öncelikle betimsel analiz yapılmış aritmetik ortalama ve standart sapmalar hesaplanmıştır.
- Gruplardaki her veri setine normallik analizi yapılmıştır.
- Verilerin normal dağılımının kontrol edilmesi için çarpıklık basıklık değerleri, histogram grafiği ve Shapiro-Wilk normallik testi yapılmış her iki grupta da verilerin normal dağıldığı gözlenmiştir.

#### BTEG ve PTEG Grupları Kestirimsel Analizler

- (1. Araştırma Sorusu) BPKÖ gruplar arası fark için Bağımsız Gruplar T-Testi
- (2. Araştırma Sorusu) BYÖ alt boyutları gruplar arası fark için Tek yönlü MANOVA
- (3. Araştırma Sorusu) PBT gruplar arası fark için Bağımsız Gruplar T-Testi

#### İlişkisel Analizler

- (4. Araştırma Sorusu) Gruplara göre kaygı, başarı, bilişsel yük arasındaki ilişki için Pearson çoklu korelasyon testi

Şekil 14. Nicel veri analiz süreci.

Şekil 14'te de belirtildiği üzere Bilişsel Yük Ölçeği 3 alt boyutunun gruplara göre istatistiksel olarak karşılaştırılmasında tek yönlü MANOVA analizi kullanılabilmesi için birtakım varsayımların veri seti tarafından karşılanabilmesi gerekmektedir (Akbulut, 2010; Field, 2009). MANOVA varsayımları sınanmış ve maddeler halinde açıklanmıştır:

- Örneklem sayısı şartına göre her bir hücrede en az bağımlı değişken sayısı kadar katılımcı olması gerekmektedir. Bu çalışmadaki PTEG grubunda 34 kız, 12 erkek öğrenci yer alırken BTEG grubunda 36 kız 8 erkek öğrenci yer almaktadır. Verilen değerler bağımlı değişken sayısından büyük olduğu için veri seti bu varsayımı karşılamıştır.
- Normallik, MANOVA testlerinde aranan diğer varsayımlardan bir tanesidir. Normallik varsayımı, tek değişkenli normallik ve çok değişkenli normallik olmak üzere 2 alt boyutta incelenmektedir. Tek değişkenli normallik için her bir bağımlı değişkene ait verileri çarpıklık-basıklık katsayıları, histogram grafikleri incelenmiş ve verilerin normal dağılım sergilediği görülmüştür. Çok değişkenli normallik analizi için tüm alt boyutlarda, Mahalonobis uzaklıkları hesaplanmış ve herhangi bir uç değere rastlanmamıştır.
- Varyans-Kovaryans matrislerinin homojenliği için Box'ın Varyans-Kovaryans matrisleri homojenlik testi uygulanmıştır. Değerlere bakıldığında varyans-kovaryans matrisinin homojen olduğu, varsayımın sağlandığı görülmüştür ( $p=.970$ ,  $p>.05$ ).
- Hata varyanslarının homojenliğinin incelenmesi içinse veri setine her bir bağımlı değişken için Levene F testi uygulanmıştır. Analiz sonucu içsel bilişsel yük ( $p=.683$ ), dışsal bilişsel yük ( $p=.435$ ), etkili bilişsel yük ( $p=.647$ ) bağımlı değişkenlerinin hata varyanslarının homojen olduğu görülmüştür ( $p>.05$ ).
- Varsayımlardan birisi de bağımlı değişkenler arasında çoklu bağıntı ve tekilliğin olmamasıdır. Bağımlı değişkenler arası ilişkinin .90 düzeyi ve üstünün olmaması varsayımın sağlanması anlamına gelmektedir. Tablo 11'e bakıldığında bu varsayımın da sağlandığı ortaya çıkmaktadır.

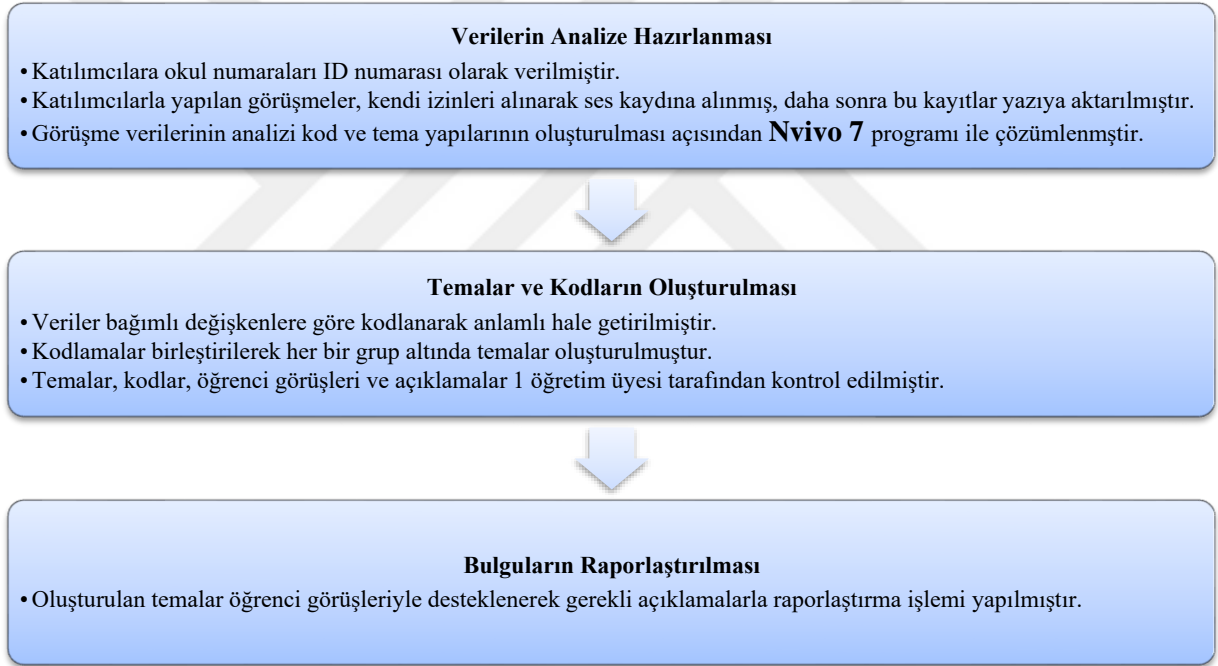
Tablo 11.*Hata Varyanslarının Homojenliği*

	Bağımlı	İçsel bilişsel yük	Dışsal bilişsel yük	Etkili bilişsel yük
<b>Tüm örneklem</b>	İçsel bilişsel yük	-	.825*	.780*
	Dışsal bilişsel yük	.825*	-	.886*
	Etkili bilişsel yük	.780*	.886*	-
<b>Deney grubu</b>	İçsel bilişsel yük	-	.787*	.741*
	Dışsal bilişsel yük	.787*	-	.879*
	Etkili bilişsel yük	.741*	.879*	-
<b>Kontrol grubu</b>	İçsel bilişsel yük	-	.851*	.805*
	Dışsal bilişsel yük	.851*	-	.890*
	Etkili bilişsel yük	.805*	.890*	-

\*  $p < .01$

### **Nitel veri analizi.**

Bu çalışmada her iki gruptaki programlama dili öğretim sürecini etkileyen faktörlerin tespiti için görüşmelerden elde edilen veriler nitel veri analizi tekniklerinden İçerik Analiziyle derinlemesine incelenmiştir. İçerik analizi, belirli bir çerçeve içerisinde yürütülen araştırma ışığında, genel yönelimleri ve sonuçları kimlikleştirmeyi ve tanımlamayı amaçlayan, uygulama süreci uzun süren araştırma gruplarında tercih edilen ve gerektiğinde yapılan incelemenin tekrar edilmesine olanak sağlayan bir derlemedir (Büyüköztürk *vd.*, 2017; Creswell & Creswell, 2017;). Özellikle betimsel analizde farkedilmeyen tema ve kodların ortaya çıkarılması, bu kodların birbiriyle ilişkilerinin detaylı incelenerek araştırma değişkenleri çerçevesinde bir bütün olarak ele alınması amacıyla bu çalışmada içerik analizi yapılmıştır (Fraenkel, Wallen & Hyun, 2012 ; Yıldırım & Şimşek, 2006). Nitel veri analizi süreci Şekil 15’te gösterilmiştir.



Şekil 15. Nitel veri analiz süreci.

### **Araştırmacının Rolü**

Araştırma sürecinin sağlıklı yürütülebilmesi açısından bu çalışmada araştırmacı uygulama öncesi, sırası ve sonrasında önemli görevler üstlenmiştir. Uygulama öncesinde öğretim materyallerini ve etkinlikleri hazırlamış, hazır veri toplama araçlarını elde etmiş, bazı veri toplama araçlarını geliştirmiş, pilot uygulamaları yapmıştır. Uygulama sırasında; ders öğretmeni olarak sürecin ve öğretim materyalinin tanıtımını yapmış, her iki gruptaki öğretim sürecini 10 hafta boyunca bizzat yönetmiştir. Uygulama sonrasında; veri toplama sürecini gerçekleştirmiş, nicel ve nitel analizlerini yapmıştır. Süreç içerisinde karşılaşılan problemlere

araştırmacı anında müdahale etmiştir. Araştırmacı müfredatta yer alan içeriklerin gruplara aktarılmasında her iki grup için de aynı süre harcanmasına özen göstermiştir. Blok tabanlı Blockly programlama aracının Python programlama diline uygun hale getirilmesi de araştırmacı tarafından yapılmıştır.

### Çalışmanın Geçerlik ve Güvenirliği

Eğitim araştırmalarında, çalışmaların alan yazına katkı sağlaması, inandırıcı olması ve doğru sonuçlar vermesi önemli görülmekte ve araştırmanın niteliğinin artırılması için çeşitli önlemler alınmasının gerekliliği belirtilmektedir (Büyüköztürk vd., 2017; McMillan & Schumacher, 2010). Çalışmalarda geçerlik ve güvenirlilik kavramları birbirini etkilediğinden alınan önlemler kesin çizgilerle ayrılmamaktadır (Topu, Baydaş, Turan, & Göktaş, 2012). Karma yöntemin temel alındığı çalışmalarda ise hem nicel hem nitel yöntemdeki önlemlere başvurulmaktadır. Bu çalışmada karma araştırma desenlerinden çeşitlemeye başvurulmuş hem nicel hem nitel araştırmalarda geçerlik güvenirlilik için alınan önlemler birlikte kullanılmış, böylece çalışmanın inandırıcılığı, genellenebilirliği, tutarlılığı artırılmaya çalışılmıştır (Fraenkel, Wallen, & Hyun, 2012) Çalışmada alınan geçerlik ve güvenirlilik önlemleri Tablo 12’de gösterilmiştir.

Tablo 12.Çalışmada Alınan Geçerlik ve Güvenirlilik Önlemleri

Strateji	Önlem
Geçerlik	Çalışma sürecinde araştırmacı rolü belirtilmiştir.
	Kullanılacak blok tabanlı programlama aracı 2 yüksek lisans ve 2 BT öğretmeni tarafından kontrol edilmiştir.
	Araştırma sorularının tüm unsurlarıyla cevaplanabilmesi için karma araştırma yöntemi tercih edilmiştir.
	Hazır veri toplama araçlarına alanyazın araması sonrasında uzman görüşü de alınarak karar verilmiştir.
Dış Geçerlik	Pilot uygulamalar yapılmıştır.
	Örnekleme seçimi açıklanırken detaylı anlatılmış, sayısı mümkün olduğunca yüksek tutulmuştur.
	Veri toplama araçlarının geçerlik ve güvenirlilik önlemleri detaylı şekilde açıklanmıştır.
	Veri toplama süreci açıklanmıştır.
Güvenirlilik	Çalışmada kullanılan araştırma yöntemi detaylı şekilde açıklanmıştır.
	Uygulama süreci, öncesi ve sonrasıyla açıklanmıştır.
	Python’a uyumlu hale getirilen bloklar pilot çalışmayla gözden geçirilmiştir.
	Çalışma gruplarının denkliği kontrol edilmiştir.
İç Güvenirlilik	Çalışmaya başlamadan önce katılımcı gruplara uygulama süreci ve veri toplama araçları tanıtılmıştır.
	Veri toplama araçları yönergelerle açıklanmış, güvenirliliği olumsuz etkilememesi için madde sayısına dikkat edilmiştir.
	Veri toplama araçlarının, tüm gruplar için benzer ortam ve aynı zamanda uygulanmasına özen gösterilmiştir.
	Ses ve video kayıtları yazı diline aktarılırken tekrar tekrar incelenmiştir.
Dış Güvenirlilik	Çalışma verilerinin sentezi için uzman, akran ve danışman görüşüne başvurulmuştur.

## Bölüm Özeti

Blok tabanlı programlama öğretiminin öğrencilerin programlama kaygısı, bilişsel yükü ve programlama başarısına etkisinin incelendiği bu çalışmada, karma araştırma yöntemi kullanılmış olup çeşitleme deseninden yararlanılmıştır. Araştırmanın nicel boyutunda blok tabanlı Blockly programlama aracı kullanılan ve kullanılmayan gruplarda programlama kaygısı, bilişsel yük ve programlama başarısı arasında fark ve ilişki olup olmadığı incelenmiştir. Nitel boyutta ise programlama öğretim sürecini ele alınan değişkenler açısından etkileyen faktörler detaylı olarak araştırılmıştır. Çalışma grubu; amaçlı örnekleme yöntemiyle seçilmiştir ve Alanya’da lise hazırlık sınıfında öğrenim gören 70 kız ve 20 erkek olmak üzere her iki gruptaki toplam 90 öğrenciden oluşmaktadır. Çalışmada öğrencilerin programlama kaygılarını belirlemek amacıyla programlama kaygı ölçeği, bilişsel yük ölçeği, Python programlama başarı testi ve yarı yapılandırılmış görüşme formu, veri toplama aracı olarak kullanılmıştır. Çalışma, 1 hafta uygulama öncesi, 10 hafta uygulama süreci ve 1 hafta uygulama sonrası olmak üzere 12 haftalık bir süreçten oluşmaktadır. Uygulama öncesinde programlama kaygı ölçeği ve başarı testi ön-test olarak uygulanmıştır. Uygulama sürecinde ise Bilgisayar Bilimi dersi müfredatına göre hazırlanmış 10 haftalık öğretim programı takip edilmiş (her gruba haftada 4 saat) ve her haftanın sonunda o haftaya ait konuyu kapsayan bilişsel yük ölçeği çalışma grubuna uygulanmıştır. Son hafta ise son-test olarak programlama kaygı ölçeği, başarı testi ve görüşme formu uygulanmıştır. Veri toplama süreci sonunda veriler araştırma soruları ışığında değerlendirilmiştir. Nicel veri analizinde, betimsel analiz, kestirimsel analiz ve ilişkisel analizden yararlanılmış, bağımsız örneklem t-testi, bağımlı örneklem t-testi, tek yönlü MANOVA ve Pearson çoklu korelasyon testi yapılmıştır. Nitel veri analizinde ise içerik analiziyle tema ve kodlar ortaya çıkarılmıştır. Çalışmada, araştırma kapsamından çıkılmaması, doğru sonuçlar vererek alan yazına katkı sağlaması için çeşitli geçerlik ve güvenirlik önlemleri alınmıştır.

## DÖRDÜNCÜ BÖLÜM

### Bulgular

Bu bölümde çalışmanın veri analizi sonucu elde edilen bulgular raporlaştırılmıştır. Bulgular araştırma soruları çerçevesinde sunulmuştur.

#### Öğrencilerin Programlamaya Yönelik Kaygı Seviyelerinin Gruplara Göre Farkı

Çalışmada Blockly tabanlı programlama eğitimi yapılan grup (BG) ile Python tabanlı programlama eğitimi yapılan grup (PG)'un programlamaya yönelik kaygıları arasında anlamlı farklılık olup olmadığını belirlemeden önce grupların ön-test ve son-test kaygı düzeyleri betimsel istatistik yöntemi ile analiz edilmiştir. Bulgular Tablo 13'de gösterilmiştir.

Tablo 13. *Gruplara Göre Ön-Test ve Son-Test Programlama Kaygı Seviyeleri*

	Çalışma Grubu	N	$\bar{X}$	SS
Ön - Test	BG	44	1.35	.756
	PG	46	.89	.783
Son - Test	BG	44	1.51	.841
	PG	46	1.55	.964

Tablo 13 incelendiğinde, BG ön-test programlama kaygı seviyesinin PG'nin kaygı seviyesinden daha yüksek olduğu görülmektedir. Uygulama sonrasında ise BG programlama kaygı seviyesinin PG'nin kaygı seviyesinden düşük olduğu görülmüştür.

BG ve PG son-test programlama kaygı seviyeleri arasında anlamlı bir farklılık olup olmadığını saptamak için Bağımsız Örneklem t-Testi yapılmıştır. Bulgular Tablo 14'te sunulmuştur.

Tablo 14. *Öğrencilerin Son-Test Programlama Kaygı Seviyelerinin Gruplara Göre Farkı*

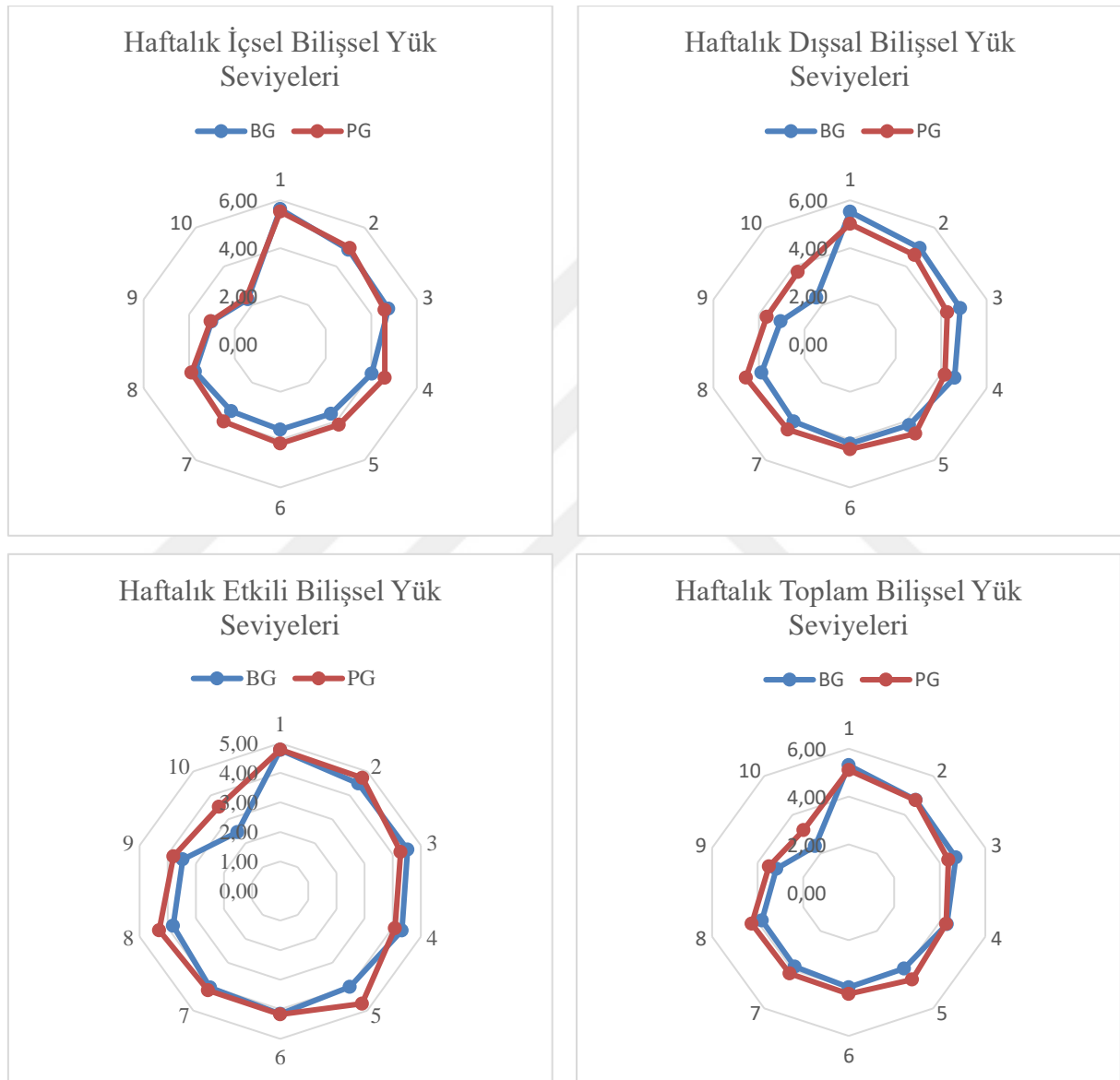
	N	$\bar{X}$	SS	sd	t	p	R <sup>2</sup>
BG	44	1.51	.96	88	.187	.852	-.044
PG	46	1.55	.84				

Tablo 14 incelendiğinde blok tabanlı programlama aracıyla öğretim yapılan grubun programlama kaygı ortalaması ile metin tabanlı programlama aracıyla öğretim yapılan grubun programlama kaygı ortalaması arasında anlamlı bir farklılık olmadığı saptanmıştır ( $t_{(88)} = .187$ ,  $p > .05$ ,  $R^2 = -.044$ ). Ancak BG'nin kaygı seviyesinin PG'den düşük olduğu görülmüştür.

Bu bulgu yapılan blok tabanlı programlama etkinliklerinin öğrencilerin programlama kaygılarını azalttığını göstermektedir.

### Öğrencilerin Bilişsel Yük Seviyelerinin Gruplara Göre Farkı

Çalışmada öncelikle 10 haftalık uygulama sürecinde BG ve PG gruplarındaki öğrencilerin haftalık olarak içsel, dışsal, etkili ve genel bilişsel yük düzeyleri belirlenmiş ve Şekil 16'da sunulmuştur.



Şekil 16. Gruplara göre haftalık bilişsel yük seviyeleri.

Şekil 16 incelendiğinde her iki gruptaki öğrencilerin ilk haftaki toplam bilişsel yük seviyelerinin son haftadan yüksek olduğu görülmüştür. Bu durum öğrencilerin programlama eğitimi ve bu yöndeki konularla ilk kez karşılaşmış olmalarından kaynaklanmış olabilir. İçsel bilişsel yüklenmeye bakıldığında ise her iki grubun ilk 3 hafta bilişsel yüklenmelerinin yakın olduğu ve ilk hafta yüksek bilişsel yüklendikleri görülmüştür. Bu durum konu yapısının zor

olduğunu ve programlama aracına bakılmaksızın öğrencilerin algılarının benzer olduğunu ortaya koymaktadır. Öğrencilerin dışsal bilişsel yüklenmeleri incelendiğinde ise ilk 5 hafta Blockly kullanılan BG'nun PG grubuna göre daha yüksek bilişsel yüklandikleri belirlenmiştir. 5. haftadan sonra tersine döndüğü saptanan bu durumun sebebi olarak öğrencilerin öğretim aracına alışma süreci gösterilebilir. Öğrencilerin etkili bilişsel yüklenmelerinin ise dışsal bilişsel yüklenmeye paralel şekilde hareket ettiği ve 5. haftadan sonra BG lehine yön değiştirdiği görülmüştür. Bu durum bilişsel yük kuramını desteklemektedir. Dışsal bilişsel yüklenmeleri azalan öğrencilerin etkili bilişsel yüklenmeleri düşmektedir.

Uygulama süreci boyunca toplam bilişsel yükün yanı sıra içsel, dışsal ve etkili olmak üzere 3 alt boyutta incelenen bilişsel yük, her hafta son ders saati verilen ölçeklerle ölçülerek 10 hafta boyunca takip edilmiş, BG ve PG gruplarının bilişsel yük ortalamaları betimsel istatistik yöntemiyle ortaya çıkarılmıştır. Bulgular Tablo 15'de gösterilmiştir.

Tablo 15. *Gruplara Göre Öğrencilerin Bilişsel Yük Seviyeleri*

	İçsel Bilişsel Yük		Dışsal Bilişsel Yük		Etkili Bilişsel Yük		Toplam Bilişsel Yük	
	$\bar{X}$	SS	$\bar{X}$	SS	$\bar{X}$	SS	$\bar{X}$	SS
<b>BG</b>	3.90	1.11	4.16	.98	4.01	1.15	4.02	1.00
<b>PG</b>	4.13	1.27	4.34	1.12	4.25	1.30	4.24	1.16

Tablo 15'e göre BG toplam, içsel, dışsal ve etkili bilişsel yük ortalamalarının PG toplam, içsel, dışsal ve etkili bilişsel yük ortalamalarından daha düşük olduğu görülmektedir ( $\bar{X}_{BG} = 4.02$ ,  $\bar{X}_{PG} = 4.24$ ). Bilişsel yük ortalamasının 5 ve üzeri olmasının öğrencilerin bilişsel yüklandiklerini gösterdiği düşünüldüğünde her iki grup için de hem toplam hem alt boyutlardaki bilişsel yüklenmenin düşük çıktığı görülmektedir. Bu bulgu her iki gruptaki etkinliklerin de öğrencilerde yüksek bilişsel yük oluşturmadığını göstermektedir.

Çalışmada öğrencilere uygulanan haftalık bilişsel yük ölçeğinin 10 haftalık ortalaması olan toplam bilişsel yük açısından gruplar arası farklılık olup olmadığını belirlemek için bağımsız gruplar t-testi yapılmıştır. Bulgular Tablo 16'da sunulmuştur.

Tablo 16. *Öğrencilerin Toplam Bilişsel Yük Seviyelerinin Gruplara Göre Farkı*

	N	$\bar{X}$	SS	sd	t	p	R <sup>2</sup>
<b>BG</b>	44	4.02	1.00	88	.947	.346	-.203
<b>PG</b>	46	4.24	1.16				



Tablo 16'ya göre BG ile PG toplam bilişsel yük seviyeleri arasında anlamlı bir farklılık bulunmamaktadır ( $t_{(88)} = .947$ ,  $p > .05$ ,  $R^2 = -.203$ ). Ancak BG bilişsel yük ortalamasına bakıldığında PG bilişsel yük ortalamasından daha düşük olduğu görülmektedir. Bu durum Blocky programlama aracının öğrenciler üzerinde daha az bilişsel yüklenmeye sebebiyet verdiği şeklinde yorumlanabilir.

Çalışmada uygulamanın ilk haftası ile son haftası arasında bilişsel yük seviyesi açısından gruplar arası farklılık olup olmadığını belirlemek amacıyla bağımsız gruplar t testi uygulanmıştır. Sonuçlar Tablo 17'de sunulmuştur.

Tablo 17. Öğrencilerin 1. ve 10. Hafta Toplam Bilişsel Yük Seviyelerinin Gruplara Göre Farkı

		N	$\bar{X}$	SS	sd	t	p	R <sup>2</sup>
<b>1.hafta</b>	<b>BG</b>	44	5.31	1.47	88	-.714	.477	.150
	<b>PG</b>	46	5.11	1.18				
		N	$\bar{X}$	SS	sd	t	p	R <sup>2</sup>
<b>10. hafta</b>	<b>BG</b>	44	2.38	.92	86.04	3.80	.000	-.809
	<b>PG</b>	46	3.21	1.12				

Tablo 17 incelendiğinde 1. hafta BG ve PG arasında bilişsel yük puan ortalamaları arasında anlamlı bir fark bulunmazken 10. hafta BG ve PG arasında bilişsel yük puan ortalamaları arasında anlamlı bir fark olduğu tespit edilmiştir ( $t_{(88)} = .947$ ,  $p > .05$ ,  $R^2 = -.203$ ). Bu durum ilk hafta her iki grupta görülen yüksek bilişsel yük seviyesinin ilk kez verilen programlama eğitimine ve programlama araçlarına alışma sürecinden kaynaklanmış olabilir. Ayrıca son hafta BG lehine belirlenen düşük bilişsel yük seviyesi blok tabanlı programlama aracının uygulama sürecine olumlu katkısını göstermektedir.

Çalışmada ayrıca grup içi 1. ve 10. haftaki bilişsel yük seviyeleri arasında farklılık olup olmadığını belirlemek adına bağımlı gruplar t testi yapılmış, sonuçlar Tablo 18'de sunulmuştur.

Tablo 18. Grup İçi 1. ve 10. Hafta Toplam Bilişsel Yük Seviyeleri Farkı

Grup	$\bar{X}$	SS	sd	t	p	R <sup>2</sup>
<b>PG 1. ve 10. hafta</b>	1.89	1.45	45	8.851	.000	1.303
<b>BG 1. ve 10. hafta</b>	2.92	1.50	43	12.921	.000	1.946

Tablo 18'e göre PG öğrencilerinin 1. ve 10. hafta toplam bilişsel yüklenme seviyelerinin anlamlı bir farklılık gösterdiği ( $t_{(45)} = .000$ ,  $p > .05$ ,  $R^2 = 1.303$ ) ve bu farklılığın 10. hafta bilişsel yük seviyesi lehine olduğu tespit edilmiştir (1. Hafta  $\bar{X}_{PG} = 5.11$ , 10. Hafta  $\bar{X}_{PG} = 3.21$ ). Benzer şekilde BG öğrencilerinin 1. ve 10. hafta toplam bilişsel yüklenme seviyelerinin de anlamlı bir farklılık gösterdiği ( $t_{(43)} = .000$ ,  $p > .05$ ,  $R^2 = 1.946$ ) ve bu farklılığın 10. hafta bilişsel yük seviyesi lehine olduğu tespit edilmiştir (1. Hafta  $\bar{X}_{PG} = 5.31$ , 10. Hafta  $\bar{X}_{PG} = 2.38$ ). Bu bulgular her iki grubun da programlama öğretim sürecinin uzunluğuna bağlı olarak öğrenme ortamına ve öğrenme araçlarına alıştıklarını kanıtlar niteliktedir.

Blockly görsel programlama aracının kullanıldığı ve kullanılmadığı öğretim ortamlarında öğrencilerin içsel, dışsal ve etkili bilişsel yük ortalamaları arasında anlamlı bir fark olup olmadığını saptamak için tek yönlü MANOVA testi uygulanmıştır. Testin uygulanmasında bağımlı değişkenler olarak birbiriyle ilişkili içsel, dışsal ve etkili bilişsel yük alt boyutları, bağımsız değişken olarak ise gruplar dikkate alınmıştır. Elde edilen bulgular Tablo 19'da sunulmuştur.

Tablo 19. *Gruplara Göre İçsel, Dışsal, Etkili Bilişsel Yük Arasındaki Farka Yönelik MANOVA Sonuçları*

	<b><math>\Lambda</math></b>	<b>F</b>	<b>p</b>	<b>R<sup>2</sup></b>
<b>Intercept</b>	.056	479.901	<b>.000</b>	.944
<b>Grup</b>	.988	.349	<b>.790</b>	.012

Tablo 19 incelendiğinde Blockly görsel programlama aracının kullanıldığı ve kullanılmadığı öğretim ortamlarında öğrencilerin içsel, dışsal ve etkili bilişsel yük ortalamaları arasında anlamlı bir farklılığın olmadığı görülmektedir ( $Wilks' Lambda = .988$ ,  $F_{(2,88)} = .349$ ,  $p > .017$ ). Bu bulgu blok tabanlı programlama aracının öğrencilerin içsel, dışsal ve etkili bilişsel yüklerine etkisi olmadığını göstermektedir.

### **Öğrencilerin Programlama Başarı Puanlarının Gruplara Göre Farkı**

Blok tabanlı görsel programlama aracının kullanıldığı ve kullanılmadığı öğretim ortamlarında öğrencilerin programlama başarı seviyelerinin belirlenebilmesi için betimsel istatistik yöntemiyle analiz yapılmıştır. Elde edilen bulgular Tablo 20'de sunulmuştur.

Tablo 20. *Gruplara Göre Ön Test ve Son Test Programlama Başarı Seviyeleri*

<b>Grup</b>	<b>Ön-Test</b>		<b>Son-Test</b>	
	<b><math>\bar{X}</math></b>	<b>SS</b>	<b><math>\bar{X}</math></b>	<b>SS</b>
<b>BG</b>	32.09	13.11	61.00	12.08
<b>PG</b>	30.08	13.18	54.86	15.35

Tablo 20 incelendiğinde BG hem ön test hem son test başarı ortalamalarının PG ön test ve son test başarı ortalamalarından yüksek olduğu belirlenmiştir.

Çalışmada BG son test başarı ortalamalarıyla PG son test başarı ortalamaları arasında anlamlı bir farklılık olup olmadığını belirlemek için bağımsız örneklem t-testi yapılmıştır. Elde edilen veriler Tablo 21’de sunulmuştur.

Tablo 21. Öğrencilerin Son-Test Programlama Başarı Seviyelerinin Gruplara Göre Farkı

	N	$\bar{X}$	SS	sd	t	p	R <sup>2</sup>
<b>BG</b>	44	61.00	12.08	88	-2.09	.039	.444
<b>PG</b>	46	54.86	15.35				

Tablo 21 incelendiğinde BG programlama başarı ortalaması ile PG programlama başarı ortalaması arasında anlamlı bir farklılık tespit edilmiştir ( $t_{(88)} = -2.09$ ,  $p < .05$ ,  $R^2 = .044$ ). Bu bulgu blok tabanlı programlama etkinliklerinin metin tabanlı programlama etkinliklerine göre öğrencilerin başarılarına daha fazla katkı sağladığını göstermektedir.

Diğer taraftan blok tabanlı etkinlik yapılan grubun ön-test ve son-test programlama başarıları arasında anlamlı bir farklılık olup olmadığını tespit etmek için bağımlı örneklem t-testi yapılmıştır. Elde edilen bulgular Tablo 22’de gösterilmiştir.

Tablo 22. BG Öğrencilerinin Ön-Test ve Son-Test Programlama Başarı Seviyeleri Arasındaki Fark

Grup	Test	N	$\bar{X}$	S	sd	t	p	R <sup>2</sup>
<b>BG</b>	Ön-Test	44	32.09	13.18	43	-12.33	.000	.336
	Son-Test	44	61.00	12.08				

Tablo 22 incelendiğinde BG öğrencilerinin ön-test ve son-test başarı seviyeleri arasında anlamlı bir farklılık olduğu ( $t_{(43)} = -12.33$ ,  $p < .05$ ,  $R^2 = .336$ ), bu farklılığın son test lehine olduğu belirlenmiştir ( $\bar{X}_{(ön-test)} = 32.09$ ,  $\bar{X}_{(son-test)} = 61.00$ ).

Metin tabanlı etkinliklerin yapıldığı grubun ön-test ve son-test programlama başarıları arasında anlamlı bir farklılık olup olmadığını tespit etmek için bağımlı örneklem t-testi yapılmıştır. Elde edilen bulgular Tablo 23’te gösterilmiştir.

Tablo 23. PG Öğrencilerinin Ön-Test ve Son-Test Programlama Başarı Seviyeleri Arasındaki Fark

Grup	Test	N	$\bar{X}$	S	sd	t	p	R <sup>2</sup>
PG	Ön-Test	46	30.08	13.11	45	-10.63	.000	.481
	Son-Test	46	54.86	15.35				

Tablo 23 incelendiğinde PG öğrencilerinin ön-test ve son-test programlama başarıları arasında anlamlı bir farklılık olduğu ( $t_{(45)} = -10.63$ ,  $p < .05$ ,  $R^2 = .481$ ), bu farklılığın son test lehine olduğu belirlenmiştir ( $\bar{X}_{(ön-test)} = 30.08$ ,  $\bar{X}_{(son-test)} = 54.86$ ).

Bu bulgular hem blok tabanlı hem de metin tabanlı yapılan programlama öğretiminin öğrencilerin öğrenmesine katkı sağladığını göstermektedir.

#### Gruplara Göre Öğrencilerin Programlamaya Yönelik Kaygıları, Bilişsel Yükleri ile Akademik Başarıları Arasındaki İlişki

Blok tabanlı programlama aracının kullanıldığı ve kullanılmadığı öğretim ortamlarında öğrencilerin programlamaya yönelik kaygıları, bilişsel yükleri ve programlama başarıları arasındaki ilişkinin tespit edilebilmesi için Pearson çoklu korelasyon testi uygulanmıştır. Elde edilen bulgular Tablo 24’te sunulmuştur.

Tablo 24. Gruplara Göre Kaygı, Bilişsel Yük ve Akademik Başarı Arasındaki İlişki

	İBY	DBY	EBY	GBY	Başarı	Kaygı
PG	İBY	1				
	DBY	.851*	1			
	EBY	.805*	.890*	1		
	GBY	.935*	.960*	.949*	1	
	Başarı	-.126	-.128	-.063	-.110	1
	Kaygı	-.080	-.038	-.002	-.042	-.147
BG	İBY	1				
	DBY	.787*	1			
	EBY	.741*	.879*	1		
	GBY	.906*	.950*	.939*	1	
	Başarı	.061	.086	.116	.095	1
	Kaygı	.236	.132	.056	.151	-.176

İBY: İçsel Bilişsel Yük DBY: Dışsal Bilişsel Yük EBY: Etkili Bilişsel Yük; GBY: Genel Bilişsel Yük; \*  $p < .01$

Tablo 24 incelendiğinde, blok tabanlı programlama aracının kullanıldığı BG’de, toplam bilişsel yük ve akademik başarısı arasında pozitif yönde çok zayıf düzeyde anlamlı olmayan bir ilişki ( $r = .095$ ,  $p < .01$ ), toplam bilişsel yük ve programlama kaygısı arasında pozitif yönde çok zayıf düzeyde anlamlı olmayan bir ilişki ( $r = .151$ ,  $p < .01$ ), akademik başarı ve programlama kaygısı arasında negatif yönde çok zayıf düzeyde anlamlı olmayan bir ilişki ( $r = -.176$ ,  $p < .01$ ) olduğu belirlenmiştir. Bilişsel yükün alt boyutları arasındaki ilişkiler incelendiğinde ise içsel bilişsel yük ile dışsal bilişsel arasında pozitif yönde, güçlü ve anlamlı bir ilişki ( $r = .787$ ,  $p < .01$ ), içsel bilişsel yük ile etkili bilişsel yük arasında pozitif yönde, güçlü ve anlamlı bir ilişki ( $r = .741$ ,  $p < .01$ ), etkili bilişsel yük ile dışsal bilişsel yük arasında ise pozitif yönde, çok güçlü ve anlamlı bir ilişki ( $r = .879$ ,  $p < .01$ ) olduğu tespit edilmiştir. Fakat bilişsel yüklenmenin alt boyutları olan içsel, dışsal ve etkili bilişsel yüklenme ile diğer değişkenler olan akademik başarı ve programlama kaygısı arasında negatif yönde anlamlı olmayan çok zayıf bir ilişki tespit edilmiştir.

Metin tabanlı programlama araçlarının kullanıldığı PG’de toplam bilişsel yüklenme ve akademik başarı arasında negatif yönde çok zayıf düzeyde anlamlı olmayan bir ilişki ( $r = -.110$ ,  $p < .01$ ), toplam bilişsel yüklenme ve programlama kaygısı arasında negatif yönde çok zayıf düzeyde anlamlı olmayan bir ilişki ( $r = -.042$ ,  $p < .01$ ), akademik başarı ve programlama kaygısı arasında negatif yönde çok zayıf düzeyde anlamlı olmayan bir ilişki ( $r = -.147$ ,  $p < .01$ ) olduğu belirlenmiştir. Bilişsel yükün alt boyutları olan içsel dışsal ve etkili bilişsel yük arasındaki ilişkiler incelendiğinde ise içsel bilişsel yük ile dışsal bilişsel yük arasında pozitif yönde, anlamlı ve çok güçlü bir ilişki ( $r = .851$ ,  $p < .01$ ), içsel bilişsel yük ile etkili bilişsel yük arasında pozitif yönde, anlamlı ve çok güçlü bir ilişki ( $r = .805$ ,  $p < .01$ ), etkili bilişsel yük ile dışsal bilişsel yük arasında pozitif yönde, anlamlı ve çok güçlü bir ilişki ( $r = .890$ ,  $p < .01$ ) olduğu belirlenmiştir. Bunun yanında içsel, dışsal ve etkili bilişsel yükün akademik başarı ve programlama kaygısı ile negatif yönde anlamlı olmayan ve çok zayıf bir ilişki olduğu görülmüştür. Bu bulgular 10 hafta gibi uzun bir sürece yayılan programlama öğretiminde her iki gruptaki öğrencilerin bilişsel yük seviyelerinin düşük çıkmasından kaynaklanmış olabilir. Ayrıca her iki grupta da programlama başarısı ile programlama kaygısı ve toplam bilişsel yük arasında anlamlı ilişki çıkmaması alanyazınla zıtlık göstermektedir .

### **BG ve PG Öğrencilerinin Programlama Öğretim Sürecinde Kaygılarını, Bilişsel Yüklerini ve Programlama Başarılarını Etkileyen Faktörler**

Her iki gruptaki öğrencilerin programlama dili öğrenmeye yönelik kaygılarını, bilişsel yüklerini ve başarılarını etkileyen faktörleri belirlemek için içerik analizi yoluyla birçok tema ve kod ortaya çıkarılmıştır. Bazı temalar altında ortak faktörler olduğu, faktörlerin öğrencileri etkileme durumlarının gruplara göre değişiklik gösterdiği de saptanmıştır. Devam eden

bölümde bulgular programlama kaygısı, bilişsel yük ve programlama başarısı alt başlıklarında açıklamalı olarak sunulmuştur. Her iki gruptaki öğrencilerin (ÖX) görüşlerinden alıntılarla bulgular desteklenmiştir.

### Programlama kaygısını etkileyen faktörler

BG ve PG öğrencilerinin programlama kaygılarını etkileyen faktörler temalar şeklinde Tablo 25’te gösterilmiştir. Bildirilen görüşlere yönelik açıklamalar ve temaları oluşturan doğrudan alıntılar ilgili başlık altında sunulmuştur.

Tablo 25. Programlama Kaygısını Etkileyen Faktörler

Temalar	Etkileyen Faktörler			
	Kaygı Var	f	Kaygı Yok	f
<i>Ders hakkında bilgi sahibi olma/olmama</i>	▪ Programlama dersini ilk kez alma	1	❖ Daha önce algoritma oluşturma ile ilgili ders alma	3
<i>Derse ilgi/ilgisizlik</i>	• Bilgisayar kullanmayı sevmeme	1	❖ Bilgisayar kullanmayı (oyunlarını) sevmeme	3
			❖ Bu derse gerek yok/dersi ciddiye almama	2
<i>Hata yapmaktan korkma/korkmama</i>	❖ Öğretmenin gözünden düşmekten korkma	4	❖ Herkesin hata yapabileceğini düşünme	9
	❖ Kodları karıştırmaktan/ayıramamaktan korkma	2	❖ Öğretmenen yardım alma / çekinmeme durumu	5
	▪ Arkadaşın başarılı olması durumu	1	❖ Öğrenmek için hata yapmaktan çekinmeme	4
	▪ Arkadaşın dalga geçmesinden/tepkisinden çekinme	1		
<i>Kendine güvenme/güvenmeme</i>	❖ Yapamayacağını düşünme	5	▪ Yapabileceğine/başaracağına inanma	1

- BG’yi Etkileyen Faktörler
- PG’yi Etkileyen Faktörler
- ❖ Hem BG hem PG’yi Etkileyen Faktörler

➤ **Ders hakkında bilgi sahibi olma/olmama:** Her iki grupta da ortaokuldaki BTY dersi kapsamına blok tabanlı algoritma yazmaya yönelik ders alan ve almayan öğrenciler olduğu tespit edilmiştir. Dersi alan öğrencilerin bu derse yönelik kaygıları olmadığı ancak dersi daha önce almamış öğrencilerin bu dersin içeriğinin nasıl olacağına yönelik kaygı duydukları tespit edilmiştir. Bu yöndeki öğrenci görüşleri şöyledir:

#### **Programlama dersini ilk kez alma**

*“Başta biraz endişelendim. Endişelenmemin sebebi daha önce hiç görmediğim bir dil öğrenecektim, Python öğrenecektim. Yapamam diye korkuyordum baştan ama sonra zor olmadığını gördüm ve endişem yok oldu (BG, Ö1, Kız).”*

#### **Daha önce algoritma yazma ile ilgili ders alma**

*“Biraz endişelendim ama fazla değil yani. Çünkü biz önceden görmüştük bu dersi ortaokulda yani. Hoca giriyordu çıkıyordu giriyordu çıkıyordu onun gibi bir şey olabilir diye düşündüm ama burada ders işledik. O yüzden girdikten sonra endişelendim. Başında değil (BG, Ö4, Kız).”*

*“Hayır, endişelenmedim çünkü daha öncede bu dersi gördüm. O yüzden bir sıkıntı görmüyorum. Ortaokuldayken de gördüm uygulamalı olarak. Biz hatta*

*daha çok görüyorduk. Yani o yüzden pek bir şey olmadı. Endişem olmadı (BG, Ö2, Kız). ”*

*“Fazla endişelenmedim çünkü ortaokulda da görmüştük ama böyle olacağını bilmiyordum (PG, Ö16, Kız). ”*

➤ **Derse ilgi / ilgisizlik:** BG ve PG içerisinde bilgisayar kullanmayı seven ve sevmeyen öğrencilerin olduğu, bilgisayar kullanmayı seven öğrencilerin herhangi bir kaygı beslemedikleri fakat sevmeyen öğrencilerin kaygılandıkları görülmüştür. Ayrıca böyle bir derse gerek olmadığı düşüncesine sahip ve dersi ciddiye almadığını ifade eden öğrencilerin kaygılanmadıkları belirlenmiştir. Bu yöndeki öğrenci görüşleri şöyledir:

#### ***Bu derse gerek yok/dersi ciddiye almama***

*“Endişelenmedim. Ciddiye almadığım için endişelenmedim. Çünkü doğru kelime herhalde saçma yani ne bileyim. Bize neden programlama dili öğretiyorlar. Neden biz? Sosyal bilimlerde ne işi var? Bize neden böyle bir ders veriyorlar (BG, Ö3, Kız)? ”*

#### ***Bilgisayar kullanmayı (oyunlarını) sevmeme***

*“Hayır endişelenmedim. Çünkü 6-7 yaşından beri bilgisayarla aram çok iyi. Bilgisayar oyunları oynuyorum. Oynadığım oyunlar arasında minik de olsa kodlama içeren oyunlar var. Minik de olsa sadece birkaç kelime de olsa İngilizce gerektiren bir programlama dili. O yüzden de herhangi bir endişe duymadım. Rahattım hala da rahatım (BG, Ö9, Erkek). ”*

*“...Endişelenmedim. Bilgisayarla ilgili şeyler yapmak güzel yani. Bilişim dersi kafa dinliyoruz hem. (PG, Ö14, Kız). ”*

*“Endişelenmedim. Çünkü buda her zamanki normal derslerimizden birisi. Matematik neyse bilişim de bana göre öyle. Matematikte nasıl öğreniyorsak bilişimde de öyle öğreniyoruz. Benim için fark eden yine bir şey olmadı (PG, Ö15, Erkek). ”*

#### ***Bilgisayar kullanmayı sevmeme***

*“...Ben zaten bilgisayarı fazla kullanmayı sevmiyorum. Bi de kod yazdığımızı öğrenince biraz zorlandım (PG, Ö17, Erkek). ”*

➤ **Kendine güvenme / güvenmeme:** Öğrencilerin kendilerine duydukları güvenin dersin kazanımlarını başarabileceği inancını artırdığı, güven duymayan öğrencilerin ise bu inanca sahip olmadığı ortaya çıkmıştır. Bu yöndeki öğrenci görüşleri şöyledir:

#### ***Yapamayacağını düşünme***

*“Biraz endişelendim. Zor falan yapamam ben bunu sınavlarda falan dedim de. Ama şimdi yapabildiğimi düşünüyorum (PG, Ö13, Kız). ”*

*“Endişeleniyorum çünkü yapamayacağımı düşünüyorum. Öyle de oluyor. Aslında anlıyorum hatta siz 1. Örneği yazıyorsunuz anlıyorum bir daha kendim yapacağım zaman zorlanıyorum. Herhâlde bende bir problem var (BG, Ö4, Kız). ”*

*“...Evet, biraz endişeleniyorum. Çünkü yapamama korkusu geliyor. Ama üzerine çaba sarf ettiğim sürece ‘yapacağım yapacağım’ dediğim sürece korku gidiyor (BG, Ö5, Erkek). ”*

*“Başta biraz endişeleniyorum bu beni zorlar mı zorlamaz mı diye endişeleniyorum. Gerçekten beni zorlayacak bir şeyse daha da çözmeye odaklanıyorum. Eğer çok zorlamayacak bir şeyse arkadaşlarımla birlikte çözmeyi daha çok tercih ediyorum (BG, Ö1, Kız).”*

### **Yapabileceğine/başaracağına inanma**

*“İlk başlarda endişeleniyordum. Çok umursamıyordum. Ama sonradan hani biraz kendimi verince çok endişelenmedim yapabildiğimi görünce (BG, Ö7, Kız).”*

*“Hayır. Sonuçta bilinçaltımla düşünüyorum. Siz bahsediyorsunuz bunu yapboz haline getiriyorum, öncelikle parçalara bölüp bir parçasını nereden oluşturacağımı düşünüyorum. Bu da yapbozu tamamlamak benim için çok kolay olduğundan direk yapıyorum. O esnada zaten kimseyle konuşmuyorum. Kendi kendimeyim. Endişelenmiyorum (BG, Ö8, Kız).”*

➤**Hata yapmaktan korkma / korkmama:** Her iki gruptaki öğrencilerin program kodlarını karıştırmaktan ve öğretmenin gözünden düşme korkusundan dolayı kaygı duydukları, BG’de ise arkadaş tepkisinden dolayı öğrencilerin kaygı duydukları görülmektedir. Ayrıca yine her iki grupta herkesin hata yapabileceği, öğrenmek için hata yapmaktan çekinmeyeceği ve gerekirse öğretmenden yardım alabilme durumundan dolayı kaygı duyulmadığı ortaya çıkmıştır. Bu yöndeki öğrenci görüşleri şöyledir:

### **Kodları karıştırmaktan/ayıramamaktan korkma**

*“Tabi biraz endişelendim. Dediğim gibi zaten karışık şeyleri sevmiyorum. Bunda da bir sürü şeyi aklımda tutmam gerekecek. Yanlış şeyleri yanlış yerlere yerleştirdince hata oluşacak ve o hatalar beni strese sokacak. O yüzden hoşlanmadım (PG, Ö18, Kız).”*

*“Birbirine karıştırabileceğimi düşünüyorum kodlamaları. O yüzden minik ufak bir endişe, telaş oluyor içimde. Sonra yapmaya başladıkça doğru gittiğimi gördükçe o endişe de ortadan kayboluyor. Konuları karıştırmaktan korkuyorum. Yerlerini sıralarını. Kodların sırasını ve bu konuda bunu mu kullanacaktım yoksa bunu mu? Bu ikisi yüzünden telaşa düşüyorum (BG, Ö9, Erkek).”*

*“Endişelenmiyorum. Çünkü yapabileceğime inanıyorum. Yapamam bile yaptığım yanlışlarla doğruları öğreniyorum. O yüzden hiçbir endişem olmuyor (PG, Ö18, Kız).”*

*“Panik olmam. Öğrendiğim gibi yazmaya başladım. Mesela siz bana ne öğrettiyseniz ona göre devam ederim. Endişelenecek bir şey olmadığını düşündüğüm için sıkıntı olmaz benim için (BG, Ö10, Erkek).”*

### **Öğretmenin gözünden düşmekten korkma**

*“Biraz endişeleniyorum. Yapamazsam mesela hocanın bakış açısı değişir diye (PG, Ö17, Erkek).”*

*“Öğretmenin bir şey demeyeceğini, hem yeni yeni öğrendiğimiz için tepki göstermeyeceğini düşündüğüm için endişelenmiyorum (BG, Ö5, Erkek).”*

*“Çekinmem. Öğretmenimizin vereceği tepkiyi biliyorum. Yapamıyor olsam da birçok öğrenci hata yapıyor programlamada. Ben de hata yapabilirim. O*



yüzden bir endişem olmuyor ya da korkum olmuyor. Rahat oluyorum (PG, Ö18, Kız).”

“Bilmiyorum. Soru eğer kolaysa ve ben bunu yapabilecek kapasitedeysem çekinirim. Ama bu soru gerçekten zorsa çekinmem çünkü yanlış yapılabilir (BG, Ö4, Kız).”

“Öğretmeniminkinden çekinebilirim. Sonuçta öğretiyor ve ben onu yapamadığımda ben öğretememişim gibisinden düşünebilir öğretmenim. Bu yüzden endişelenebilirim (BG, Ö9, Erkek).”

### **Arkadaşın dalga geçmesinden/tepkisinden çekinme**

“Evet, ilk başta hani korktum. Çünkü bir şey bilmiyordum hani diğer arkadaşların bazıları bildiği için yani utanıyordum yapamayacağım diye. O yüzden çekindim biraz. Ama sonradan hocanın anlatımından sonra yapabildiğime inandım (BG, Ö6, Kız).”

“Hayır. Sonuçta profesyonel değilim. Öğrenme aşamasındayım. Sınıfta benim gibi daha fazla öğrenmeye çalışan insan var. Onlar da bunu yapamayabilir. Ben bunun neyinden çekineyim ki. Yapamam profesyonel değilim bunda daha ilk senem (BG, Ö8, Kız).”

### **Bilişsel yükü etkileyen faktörler**

BG ve PG öğrencilerinin bilişsel yüklenmelerini etkileyen faktörler Tablo 26’da temalar şeklinde gösterilmiş, temalara yönelik açıklamalar ve doğrudan alıntılar ilgili başlıklar altında sunulmuştur.

Tablo 26. *Bilişsel Yükü Etkileyen Faktörler*

Temalar	Etkileyen Faktörler	f
<i>Bazı programlama dili konularının mantığını anlamakta zorlanma</i>	❖ Döngüleri anlamakta zorlanma	7
	❖ Liste elemanlarını oluşturmakta zorlanma	5
	❖ Fonksiyonları anlamakta zorlanma	3
	● Karar yapılarını anlamakta zorlanma	2
	● Akış şemalarını/algorithmaları anlamakta zorlanma	2
<i>Bilgisayar kullanımında tecrübesiz olma</i>	● Bilgisayar kullanmayı sevmeme	1
	● Bilgisayar kullanmama	1
<i>Derse motive olma/olmama</i>	❖ Dersin sıkıcı/gereksiz olduğunu düşünme	6
	❖ Dersi dinlememe	2
	❖ Sınav kaygısı nedeniyle dersi geçecek kadar çaba sarf etme	2
	■ Derse ilgi duymama	1
	● Kod yazmaktan keyif alma	1
	● Önemsiz ders olarak görmekten dolayı zorlanma	1
	❖ Nerede hata yaptığını anlamaya çalışma/hırsızlanma	10
<i>Hatanın kaynağını bulmakta zorlanma</i>	❖ Tekrar tekrar deneme	6
	❖ Hatalı kod yazmaya yönelik olumsuz duygulara kapılma (sinirlenme, sıkılma)	5
	❖ Pes etme	4
	● Baştan yapmaya çalışma	1
	■ Blockly’de açıklayıcı/anlaşılır görsel öğeler kullanma	4
<i>Programlama aracının kolaylık sağladığını düşünme</i>	● Bilgisayarda kod yazmanın kolaylık sağladığını düşünme	2
	■ Blockly’de uygulamaya hazır sürükle bırak kod blokları kullanma	1
	■ Blockly’de düzenlenebilir kod blokları kullanma	1
	❖ Yeni bir programlama aracına alışmada zorlanma	3
<i>Programlama aracını kullanmada zorlanma</i>	● Python’da kodlama yapmakta zorlanma	2
	■ Programlama dilinde İngilizce kullanmada zorlanma	1
	❖ Örnek etkinlikler yaparak tekrar etme	3
<i>Programlama tecrübesine bağlı olarak kolay anlama</i>	❖ Programlama diline alışma	2
	❖ Programlama aracını kullanırken öğretmenden yardım alma	3
<i>Rehberliğe/yardıma ihtiyaç duyma</i>	● Kod yazma sürecinde yetiştiremeyince/yapamayınca arkadaşından yardım isteme	1

■ BG Etkileyen Faktörler

● PG Etkileyen Faktörler

❖ Hem BG hem PG Etkileyen Faktörler

➤ **Bazı programlama dili konularının mantığını anlamakta zorlanma:** Her iki grup öğrencilerinin de döngü yapılarında, listeleme öğelerinde, fonksiyon, karar yapıları, akış şemaları ve algoritmalar konularının karmaşık oluşundan ve matematikle bağlantılı oluşundan dolayı anlamada zorlandıkları ortaya çıkmıştır. Bu yöndeki öğrenci görüşleri şöyledir:

#### ***Döngüleri anlamakta zorlanma***

“...sonsuz döngüye girdiği zaman sinirleniyorum neden hata yaptım gibi ve baştan yapmak zorunda kalıyorum. Bu uzun bir kodsda daha da sinirleniyorum. Bu beni tedirgin yapıyor. Birkaç kere yaptıktan sonra 2-3 kez sonsuz döngüye girebiliyor. Farklı farklı hatalar olabiliyor. Bu hatalar git gide çoğalabiliyor. O yüzden bundan korkuyorum (BG, Ö1, Kız).”

“Bu süreç boyunca liste konusunda ve döngü konusunda zorlandığımı hissediyorum. Mesela bir yanlış yaptığımızda direk döngü devam ediyor sonsuza dek. İşte o can sıkıcı oluyor biraz. O yüzden liste ve döngü konusunda biraz zorlandım (BG, Ö10, Erkek).”

“Döngülerde zorlandım. Bilmiyorum karışık geliyordu. Kod haline getirmek zordu (PG, Ö11, Kız).”

“Son işlediğimiz konularda mesela zorlandım. Karmaşık geliyor iç içe geçmiş birbirleriyle. Hangisi hangisiyle falan bunu aklımda tutamıyorum (PG, Ö18, Kız).”

#### ***Liste elemanlarını oluşturmakta zorlanma***

“Listeleme konusunda zorlandım. Nerden ne ekleyeceğimi o zaman anlamamıştım. Onun için zorlanmıştım. Mantığında baya zorlanmıştım (BG, Ö5, Erkek).”

“.....Yani bilmiyorum ama bide liste şeyleri var. Hem yazarken çok hata yaptım. Hem de bana çok ağır geldiği için (PG, Ö15, Erkek).”

“Listeler çok karışık karmaşık geldi bana. Hocam direk mantığını anlamadım yani (BG, Ö7, Kız).”

#### ***Fonksiyonları anlamakta zorlanma***

“Fonksiyonlarda falan biraz daha işin içine matematik giriyor. Matematik zaten hepsinde var ama biraz daha farklı daha lisedeyiz ve görmediğimiz konular var. Bunlarda bilmediğimiz şeyler olunca zorlanıyorum. Fonksiyonun mantığını anlamak zorladı beni. Onun için ders dışında da çalıştığım oldu yani (BG, Ö8, Erkek).”

“Birazcık fonksiyonlarda zorlanmış olabilirim. Mantığı karışık falan o yüzden. Yazarken zorlanıyorum (PG, Ö13, Kız).”

“Fonksiyonlarda zorlandım. Mantığını anlamadım (PG, Ö17, Erkek).”

#### ***Karar yapılarını anlamakta zorlanma***

“Pythondaki if-else (karar) yapıları. Nasıl nerede kullanacağımı pek bilmiyordum. Kafam karışıyordu. Mantığını anlayamıyordum. Yazmada zorlanmıyordum ama mantığını anlayamıyordum (PG, Ö19, Kız).”

#### ***Akış şemalarını/algoritmaları anlamakta zorlanma***

“Biraz zorlandım. Algoritmada çok zorlandım karışık geldi. Özellikle o şekil olanlar onlarda çok zorlanmıştım. Hiçbir şekilde aklımda tutamıyordum. Zor geliyordu sonra tabi yavaş yavaş öğrendim (PG, Ö19, Kız).”

➤ **Bilgisayar kullanımındaki tecrübesizlik:** Öğrenciler bilgisayar kullanımına duydukları ilgisizliğin veya tecrübesizliğin ders sürecinde bir dezavantaj olduğunu belirtmişlerdir. Bu yöndeki öğrenci görüşleri şöyledir:

#### ***Bilgisayar kullanmayı sevmeme***

“Evet. Ben zaten bilgisayarı fazla kullanmayı sevmiyorum. Bi de kod yazdığımızı öğrenince biraz zorlandım. Dersin içeriği çok karışık (PG, Ö17, Erkek).”

#### ***Bilgisayar kullanmama***

“Nasıl desem bilgisayarla ilgili şeyleri pek anlayamıyorum, pek bilgisayar kullanmadım. 1 yıl falan kullanmışımdır. Sonra bıraktım bilgisayarı. Hele böyle yaptığımız şeylerle hiç alakam yoktu. Bundan dolayı genel olarak hepsinde zorluk çektim (PG, Ö16, Kız).”

➤ **Derse motive olma / olmama:** Her iki gruptaki öğrencilerin farklı nedenlerle derse motive oldukları ya da olamadıkları tespit edilmiştir. Dersi dinlememe, sıkıcı ve gereksiz olduğunu düşünme, derse yönelik ilgi eksikliği gibi faktörlerin öğrencilerin derse motive olma durumunu olumsuz etkilediği tespit edilirken, kod yazmaktan keyif alma ve sınav kaygısı nedeniyle dersi geçecek kadar çaba sarf etme gibi faktörlerden dolayı motive oldukları saptanmıştır. Buna göre öğrenci görüşleri şu şekildedir:

#### ***Dersin sıkıcı/gereksiz olduğunu düşünme***

“Anlamaya çalışmak bana sıkıcı geldiği için dersi hiç anlamadım (BG, Ö4, Erkek).”

“Genel olarak sıkıcı bulduğum için, hiç çaba sarf etmedim. Çünkü sıkıcı, saçma geliyor böyle bir programı yazacağız sonra bana hiçbir şey kazandıracağını düşünmüyorum (BG, Ö4, Kız).”

“Yani biraz ister istemez canım sıkılıyor hatasız yapmak istiyorum çünkü. Çaba gösteriyorum devam ediyorum. İlk denemede olduysa devam ederim ama 3. ve 4. de olduysa ister istemez hevesim kaçar (PG, Ö16, Kız).”

#### ***Dersi dinlememe***

“Dinlememezlik. Bir boşa vermişlik (PG, Ö15, Erkek).”

“2. dönem zorlandığımı düşünüyorum. Kod yazmamızla alakası var. İlk defa gördüğüm bir şey. Anlamak istemediğim ve anlamadığım bir şey. Devam ettirmek istemedim. En başta ciddiye almıyordum dediğim gibi. Değişir diye düşünmüştüm ama devam etti (BG, Ö3, Kız).”

#### ***Sınav kaygısı nedeniyle dersi geçecek kadar çaba sarf etme***

“Yani ne kadar çaba sarf ettim size onu söyleyeyim. Sınav dönemi geldiğinde o zaman çok çalışmaya çalıştım. O zaman sarf ettim ama dersteyken pek çaba sarf etmedim. Dersi boş vermek yani. Umursamamak. Derslere bakış açımı alakalı. Mesela bizim bu sene geçmemiz gereken dersler Türkçe ve İngilizce bi

*de Matematik. Yani onlara daha da fazla ilgi duydum. Ve bilgisayar dersini bilişim dersini biraz önemsiz buldum (PG, Ö15, Erkek)."*

*"Sınavda öğrendiğimi kanıtlayacak şekilde ilerde işime yarayabilecek kıvamda çaba sarf ettim. Oranlarsak yüzde 80 çaba sarf ettim. Bilgisayar üzerinden yaptığımız için bilgisayarı ne olursa olsun her türlü şartta kullanmayı severim. İşin içinde bilgisayar olduğu için bu yüzden bir çaba sarf etme ihtiyacı duydum (BG, Ö9, Erkek)."*

### **Derse ilgi duymama**

*"Derse ilgim olmadığı için, birazcık ilgim olsaydı sonuna kadar giderdim ama olmadığı için, aslında karneyi etkilediği için, birazcık çaba sarf ettim (BG, Ö3, Kız)."*

### **Kod yazmaktan keyif alma**

*"...hoşuma gitmeye başladı programlama yapmak. Kod yapmak falan. (PG, Ö14, Kız)."*

### **Önemsiz ders olarak görmekten dolayı zorlanma**

*"Gereksiz gördüğüm için hiç çaba sarf etmedim (PG, Ö12, Kız)."*

➤ **Hatalı kod yazmaya yönelik olumsuz duygulara kapılma:** BG görüşlerine bakıldığında kodlama sırasında aldıkları bir hata karşısında sinirlendikleri, bu durumun öğrencileri olumsuz etkilediği ortaya çıkmıştır. Her iki grup öğrencilerinin de kodlama sırasında hata yaptıklarında denemekten vazgeçmedikleri fakat doğru sonuca ulaşana kadar çaba sarf etmedikleri belirlenmiştir. Bu yöndeki öğrenci görüşleri şöyledir:

### **Nerede hata yaptığını anlamaya çalışma/hırslanma**

*"Hatamın olduğu kısma yönelirim. Yanlış yaptığımda canım sıkılır yani. Bir gerginlik olur üzerimde. Kendime sinirlenirim (BG, Ö9, Erkek)."*

*"İnceliyorum nerede hata yaptım diye bakıyorum tek tek. Anlamadıysam hocaya soruyorum. Anladıysam düzeltip tekrar deniyorum. Hırslanıyorum doğrusunu yapmaya çalışmak için (PG, Ö14, Kız)."*

*"Yapmışım ve hata veriyor. Düzeltmeye çalışırım önce bir nerede hata yaptığımı anlamaya çalışırım (PG, Ö18, Kız)."*

### **Tekrar tekrar deneme**

*"Hata verdiğinde çok sinirleniyorum. Bilmiyorum sonuçta orada hoca önceden yapıyor. Ben onu yapınca yanlış çıkıyor. Sinirleniyorum. Programı baştan kapatıp baştan açıyorum (PG, Ö13, Kız)."*

*"Yani biraz ister istemez canım sıkılıyor hatasız yapmak istiyorum çünkü. Çaba gösteriyorum devam ediyorum. İlk denemede olduysa devam ederim ama 3. ve 4. de olduysa ister istemez hevesim kaçar (PG, Ö16, Kız)."*

*"O anda beynim bir anda duruyor. Düşünüyorum. En baştan başlıyorum. Olmuyor. Yeni bir sekme açıyorum programı tekrar kuruyorum. Programda herhangi bir hata varsa o anda pes ediyorum korkuyorum sonuçta. Ellerim titremeye başlıyor (BG, Ö8, Kız)."*

### **Hatalı kod yazmaya yönelik olumsuz duygulara kapılma (sinirlenme, sıkılma)**

*"İlk önce bir gerilirim yani neden böyle olduğuna bakarım (PG, Ö19, Kız)."*

“Eğer döngüye en baştan yapmak zorunda kalıyoruz ve bu beni sinirlendiriyor. Sonsuz döngüye neden girdi şimdi bu gibisinden oluyor (BG, Ö1, Kız).”

“Kendime sinirlenirim. Çünkü çok basit bir hata olabiliyor. Tamam, çok zor bir hata olsa bir şey demem ama çoğunlukla hep basit hatalar oluyor. Bu yüzden kendime çok kızıyorum (BG, Ö2, Kız).”

### **Pes etme**

“O anda beynim bir anda duruyor. Düşünüyorum. En baştan başlıyorum. Olmuyor. Yeni bir sekme açıyorum programı tekrar kuruyorum. Programda herhangi bir hata varsa o anda pes ediyorum korkuyorum sonuçta. Ellerim titremeye başlıyor (BG, Ö8, Kız).”

“Bilmiyorum hocayı çağırırım. Sorun nerdeyse. Önce bir bakarım acaba nerede yanlış yaptım diye. Bulamayınca biraz endişelenirim. Sonra hocayı çağırırım. (PG, Ö11, Kız).”

### **Baştan yapmaya çalışma**

“Yapmışım ve hata veriyor. Düzeltmeye çalışırım önce bir nerede hata yaptığımı anlamaya çalışırım (PG, Ö18, Kız).”

➤ **Programlama aracının kolaylık sağladığını düşünme:** BG öğrencileri Blockly programının açıklayıcı ve anlaşılır görsel öğelerden oluştuğunu, hazır kod yapılarının sürüklenip bırak yöntemiyle kolay kullanıldığını, kod bloklarının kolay düzenlenebilir olduğunu ifade ederek programlama aracının kendilerine kolaylık sağladığını ifade etmişlerdir. PG öğrencileri ise metin tabanlı Python programlama aracını kullanırken öğretmen rehberliği sayesinde zorlukları aştıklarını ve kâğıt üzerinde bulunan kodların bilgisayar ortamına aktarmanın kodların anlaşılmasını kolaylaştırdığını belirtmişlerdir. Bu yöndeki öğrenci görüşleri şöyledir:

### **Blockly’de açıklayıcı/anlaşılır görsel öğeler kullanma**

“Çok iyi işime yarıyordu. Görsel olduğu için. Sadece yazı olsaydı bence daha zor olurdu (BG, Ö4, Kız).”

“Pek zorlanmadım. Basit geldi. Daha görsel olması benim açımdan daha iyi oldu. Çünkü görsel kalıcı oluyor biraz bende. Öyle olunca zorlanmadım (BG, Ö6, Kız).”

### **Bilgisayarda kod yazmanın kolaylık sağladığını düşünme**

“Hayır. Kâğıt üzerinde olduğunda çok zorlanıyordum ama bir program üzerinde yapmak daha kolay geliyor en azından bir öğrenim oluyor. Neler olduğunu görüyorum (PG, Ö19, Kız).”

“Programlama yaptığımız ortam iyi yani. Hemen öğrenebiliyoruz. O yüzden de zorlanmıyorum (PG, Ö10, Erkek).”

### **Blockly’de uygulamaya hazır sürüklenip bırak kod blokları kullanma**

“Blockly kullanırken zorlanmadım. Çünkü blokları çıkartıp ekleyebiliyoruz. O yüzden daha rahat oluyor. Şekiller daha açıklayıcı kodları hemen unutmamız için ayırmışlar (BG, Ö1, Kız).”

### **Blockly’de düzenlenebilir kod blokları kullanma**

“Genellikle hep Mouse kullandığımız için sürükleme birleştirme olduğu için herhangi bir zorlanma duymadım. Üstünde de yazılar yazdığı için biz sadece boşlukları dolduruyoruz. Bu yüzden herhangi bir zorlanma olmadığını düşünüyorum (BG, Ö9, Erkek).”

➤ **Programlama aracını kullanmada zorlanma:** Çalışma süresince gerek Blockly aracını gerekse de Python metin tabanlı kodlama editörünü kullanan öğrencilerin programlamada yabancı dil kullanımını, yeni bir programlama aracına alışmanın, kodlama yapmanın zorluğu nedeniyle sorun yaşadıkları görülmüştür. Buna göre öğrenci görüşleri şöyledir:

#### **Yeni bir programlama aracına alışmada zorlanma**

“Zorlandım çünkü birinci dönemde veya geçen senelerde farklı bir uygulama kullandık. Bu beni biraz zorladı. Hani onun bir düzeni var ya alışmak gerekiyor o düzene, biraz alışmakta zorlandım (BG, Ö8, Erkek).”

“Böyle bir şeyle ilk defa karşılaştığımız için tabii ki zorlandım. Bir de ders saatinde boş saldıığımız için kişisel olarak söylüyorum, önemsiz bir ders olarak gördüğümüz için o yüzden zorlandım çok (PG, Ö15, Erkek).”

“Zorlandım. Mesela 2. Dönem biraz daha farklı, yeni bir uygulamaya giriş yaptık. Uygulamada da kod yazmaya başladık. O kodlarda farklı bilmediğimiz yerlere değindik. Farklı bir uygulama öğrendik. Bu biraz beni tereddütte bıraktı, zorlandım (BG, Ö8, Erkek)”

#### **Python’da kodlama yapmakta zorlanma**

“İlk zamanlarda evet zorlandım ama sonra alıştım. Dediğim gibi bilgisayar pek kullanmam. Onun için bilmiyordum böyle şeyleri. O zaman zorlanmıştım da sonra öğrendim (PG, Ö16, Erkek).”

“Biraz zorlandım. Anlayıp yazmak (PG, Ö12, Kız).”

#### **Programlama dilinde İngilizce kullanmada zorlanma**

“İlk başta zorlandım. Ama sonra kelimeleri falan gördükçe bilmeye başladım ve o yüzden zorlanmadım. Yabancı bir dil olması ve listeler arası karışık geldi bana biraz. Ama sonra neyin nerde olduğunu öğrendim (BG, Ö7, Erkek).”

➤ **Programlama tecrübesine bağlı olarak anlamada kolaylık:** Her iki gruptaki öğrenciler karşılaştıkları güçlükleri programlama diline aşına olmaya başladıkça ve örnek etkinliklerle tekrar yaptıkça aştıklarını ifade etmişlerdir. Bu yöndeki öğrenci görüşleri şöyledir:

#### **Örnek etkinlikler yaparak tekrar etme**

“Baya örnek yaptığımız için her örneği yapmaya çalışıyorum. Ne çok zorlandım ne de zorlanmadım. Bazı yerler anlamadığım için zor gelir anladığım için de hemen yapabileceğim bir şeymiş gibi geliyor. Kolay gibi geliyor yani (BG, Ö10, Erkek)”

“Çalışıyorum, o gün yaptığımız şeyleri tekrar edince diğer günlerde aynı şeylerin tekrarı olduğu için, zaten hoca önce bir örnek gösteriyor, sonra bizden

ona benzer örnekler yapmamızı istiyor. Ona bakarak o örnekten yola çıkarak yapıyoruz (PG, Ö11, Kız).”

### **Programlama diline alışma**

“Başlarda zorlandım. Sonradan alışınca, biraz da sizin yaptıklarınızı not alarak zorlanmadım ondan sonra. Hocam hani görünce alışırız ya yapa yapa daha da aşına oldum konuya. Mantiğini anladım (BG, Ö7, Kız).”

“İlk başlarda zorlanıyordum. Kod yazmaya yeni başlamıştık. Sonradan zorlanmadım (PG, Ö13, Kız).”

➤ **Rehberliğe/yardıma ihtiyaç duyma:** Öğrencilerin ders sürecinde zorlandıkları ve çıkmaza girdikleri bir zaman öğretmen rehberliğine veya arkadaş yardımına ihtiyaç duydukları görülmüştür. Bu yöndeki öğrenci görüşleri şu şekildedir:

### **Programlama aracını kullanırken öğretmenden yardım alma**

“İlk denememde olmuyor çünkü hemen ben biraz kendimi üzebiliyorum. İkinci denememde hem sizin yardımınızla yapabiliyorum (BG ,Ö6, Kız).”

“Hoca gösteriyor. Kolay geliyor. Yapıyoruz. (PG, Ö11,Kız).”

“Hoca zaten örnekleri anlatıyordu bana bende yazıyordum. Sonuç çıkıyordu. Herhangi bir sorun yaşamadım (PG, Ö17, Erkek).”

### **Kod yazma sürecinde yetiştiremeyince/yapamayınca arkadaşından yardım isteme**

“Yazmak için baya uğraşıyorum mesela geride kaldığımda arkadaşlarımdan yardım alıyorum (PG, Ö17, Kız).”

## **BG ve PG öğrencilerinin programlama öğretim sürecine yönelik görüşleri**

BG ve PG öğrencilerinin programlama öğretim süreciyle ve bu dersin verilmesiyle ilgili görüşleri Tablo 27’de temalar şeklinde verilmiştir. Bu görüşler açıklamalar ve doğrudan alıntılarla desteklenmiştir.

Tablo 27. BG ve PG Öğrencilerinin Sürece Yönelik Görüşleri

Temalar	Etkileyen Faktörler	f
Çeşitli kazanımlar elde etme	❖Kodlama mantığını anlama / programlama becerisi edinme	4
	❖Bir yazılım oluşturma	3
	●Bir problemin nasıl adım adım çözüleceğini öğrenme	1
	●Hatalardan ders çıkarmayı öğrenme	1
	■Hiçbir kazanım elde etmediğini düşünme	1
Dersi almayı isteme/istememe	❖İleriki yıllarda bu bilgilerin lazım olacağını düşündüğü için bu dersi almayı isteme	5
	❖Meslek seçimini olumlu etkileme	5
	❖Ders içeriğini saçma ve zor görme	4
	❖Bilişim alanında bilgi sahibi olmak gerektiğini düşünme	4
	■Bilgisayarla ilgili bir şeyler öğrenildiği için bu dersi almayı isteme	3
	●Diğer dersler gibi yorucu olmadığı için bu dersi almayı isteme	2
	❖Bu derse ilgisi olmadığı için istememe	2
	■Faydasız bir ders olarak görme	2
	■Eğlenceli olduğu için bu dersi almayı isteme	1
	■Bu dersteği bilgileri ileride kullanmayacağını düşünme	1
	■Derse ilgisi olmayanlara verilme zorunluluğundan hoşlanmama	1
●Seçmeli ders olursa bu dersi almayı istememe	1	

■BG Etkileyen Faktörler

● PG Etkileyen Faktörler

❖ Hem BG hem PG Etkileyen Faktörler

➤**Çeşitli kazanımlar elde etme:** Programlama öğretim sürecinde her iki grup öğrencilerinin de çeşitli kazanımlar elde ettikleri ortaya çıkmıştır. Bir problemin çözümü için gereken adımları, hatalardan ders çıkarmayı ve programlama mantığını kavramayı öğrendikleri belirlenmiştir. Bu yöndeki öğrenci görüşleri şöyledir:

#### ***Kodlama mantığını anlama / programlama becerisi edinme***

*“Nasıl program yapacağımızı öğrendik. Küçükken benim de evde bilgisayarım vardı. Acaba buradan ne yapabilirim diye düşünürdüm, şimdi daha iyi öğrendim neler yapabileceğimi, nasıl program kurabileceğimi daha iyi öğrendim (BG, Ö5, Erkek).”*

*“Kazanım olarak hocam yani şimdi sonuçta biz normalde önceden görmüyorduk programlama tarzında. Bazı arkadaşlarımıza yardımcı oluyor mesleklerinde ama şu anda bana yardımcı olmadığı için sadece öğrendim konuyu. Kazanımı oldu ama ne olduğunu bilmiyorum (BG, Ö7, Kız).”*

*“Ben bir program kodu yazmayı öğrendim. Bunun gerçek hayatta yani kendim yapsam yapabileceğimi düşünmüyorum. Çünkü karışık geliyor yine. İlerisi için parlak değil diyorum. Bunu devam ettiremem (PG, Ö18, Kız).”*

*“Programlama ve yazılımı az-çok öğrendik hani. Hepsini öğrenmiş olmasak da temelini öğrendik. Yaz aylarında benim yapacağım planlarım var. Bu planlarımda oldukça katkısı olacak. Bu benim için yeterli bir kazanım diye düşünüyorum (BG, Ö8, Kız).”*

#### ***Bir yazılım oluşturma***

*“Günlük hayatımızda saniyelerde dakikalarda olsa kolaylaştırabileceğimiz kodlamalarla ufak programlar yapabileceğim bir kazanım elde ettim. İşimizi pratikleştirecek programlar (BG, Ö9, Erkek).”*

*“Kendime kısa bir uygulama yapabilirim. İlerde eğer devam edersem öğrenmeye kendi mesleğime ait bir program da yaratabilirim. Böyle düşünüyorum (BG, Ö1, Kız).”*

*“Bir programlama dilinde bir sürü şey öğrendim. Yapabileceğim bir sürü şey var. Bir programda bir kişiden bir sayı alabilmeyi öğrendim ya da herhangi bir şey. Karar yapılarını öğrendim. Bir program yapabilirim mesela öğrendiğim ders sayesinde (PG, Ö19, Kız).”*

#### ***Bir problemin nasıl adım adım çözüleceğini öğrenme***

*“Bir şeyin nasıl adım adım yapılması gerektiğini, sorunla karşılaştığımda ne yapılması gerektiğinin adımlarını öğrendim (PG, Ö11, Kız).”*

#### ***Hatalardan ders çıkarmayı öğrenme***

*“Hatalarımdan ders çıkarmayı öğrendim. Bilgisayar kodlama dersinden çok az da olsa, bir şeyler öğrenmeye çalıştım ve öğrendiğimi de düşünüyorum (PG, Ö15, Erkek).”*

#### ***Hiçbir kazanım elde etmediğini düşünme***

*“Bence bir şey öğrenmedim. Bir şey kattığını düşünmüyorum (BG, Ö4, Kız).”*



➤ **Bu dersi almayı isteme / istememe:** Katılımcıların çeşitli nedenlerle Bilgisayar Bilimi dersini tekrar seçme ya da seçmeme durumları olduğu belirlenmiştir. Her iki grup öğrencilerinden de bir programlama dili öğrenmelerinin niçin gerekli olduğuna anlam veremedikleri, ilk kez bir programlama diliyle karşılaşmaları nedeniyle öğrenmede zorluk yaşadıkları, bu nedenle dersin saçma, zor ve faydasız olduğunu, kazanımlarını ilerde kullanmayacaklarını düşündükleri yönünde ifadeler saptanmıştır. Diğer taraftan BG ve PG öğrencilerinden bazıları gelecekteki meslek seçimini olumlu etkileyeceğinden ve bilişim alanında bilgi sahibi olmak gerektiği düşüncesiyle bu desin gerekli olduğunu ifade etmişlerdir. Bu görüşlerin öğrencilerin başarılarını olumlu ya da olumsuz etkileme noktasında önemli ve değerli olduğu düşünülmektedir. Buna yönelik öğrenci görüşleri şöyledir:

***İleriki yıllarda bu bilgilerin lazım olacağını düşündüğü için bu dersi almayı isteme***

*“Yani işe yaradı biraz. Bilmediklerimi öğrendim ama öğrenmediğim kısım var onu da yaz tatilinde telafi etmeyi düşünüyorum. Mesela evde kendi kendime öğrendiklerimi yazabilirim. İlerde bu bölümü seçersem aklımda kalanlarla biraz daha ilerde başlayabilirim (PG, Ö17, Kız).”*

*“Öğrenmişimdir elbet bir şeyler. Mesela bir dahaki sene bu derse girersek gördüğümüz konularda zorlanmam diye düşünüyorum. Çünkü öğrendiğime inanıyorum yani (BG, Ö3, Kız).”*

*“İlerde böyle bir şeyle karşılaştığım zaman ilk başlarda çektiğim zorluğu çekmeyeceğim. Bir bilğim olacak. O yazılımla alakalı. İlerde karşılaşacağımı düşünmüyorum ama olurda karşılaşırsam (PG, Ö16, Kız).”*

*“Bu uygulamadan önce tabi ilk ortaokulda bu kadar bilmediğim için burada programlama az çok öğrendim hepsinden. İlerde öğretmen ya da başka bir şey olursam köy okuluna gittiğimde eğer orada bilgisayar dersi varsa ben de bildiklerimi anlatmak isterim. O yüzden çok kazanım oldu benim için. Programlama açısından düşünürsek ilerde inşallah ben de olmak isterim bilgisayar öğretmeni, bildiklerimi aktarmak benim de hoşuma gider (BG, Ö6, Kız).”*

*“Evet, şahsen bu ders seçmeli olsaydı da seçerdim ben. İyi bir ders olduğu için. Yani ilerde lazım olacak bana (BG, Ö10, Erkek).”*

***Meslek seçimini olumlu etkileme***

*“Programlama dersi bence gerekli bir ders. Çünkü ilerde gerçekten lazım olabilecek konular olabilir. Herkes kendi meslek seçimini yapıyor. Bu meslek seçimini yaparken belki kararlarımız değişebilir. Buradan daha da değişik mesleklere yönelebiliriz ya da bir hobi olarak uygulama geliştirebiliriz. En azından kendi mesleğinle ilgili bir uygulama yapıp onu kullanabilirsin (BG, Ö1, Kız)”*

*“Bence ilerde üniversite seçimimizde veya ileriki mesleğimizde baya işimize yarayacak programlama. Firmalar var isim vermeyeyim. Firmaların internet siteleri var. Tabi biz de burada programlama kodlama öğreniyoruz. Kendimizi kanıtlayabilirsek firmaların sitelerini daha uygun herkesin rahat bir şekilde*

*kullanabileceği hale getirebiliriz. İşimize baya yarar. Gerekli bir ders (BG, Ö9, Erkek)”*

*“Tabi bir zamandan sonra bunları öğrenmemiz şart. Programlama dilleri gerekiyor mu gerekiyor. İlerdeki işimizin üniversiteye gireceğimiz bölümler ve hayatımızdaki işimiz bilişimse, bilgisayar üzerineyse ona hazırlık olması için gerekli (PG, Ö15, Kız)”*

### **Ders içeriğini saçma ve zor görme**

*“Yapıyoruz diğerine geçiyoruz, ben unutuyorum yani. Ya benim zekâmda bir şey var, ne bileyim bana saçma geldi. Bir anda böyle çıkınca zor geldi (BG, Ö3, Kız).”*

*“Yani gösterilerek yapılması bence iyi bir şey. Yazmaktan daha iyi. Almamız bir yönden bakıldığında iyi bir şey. Çünkü bilgisayar hayatımız boyunca kullanacağımız bir şey. Bir yönden iyi bir şey ama diğer taraftan da zorlandığımız için hoşumuza gitmiyor dersler. Zorlandığım için çok gerekli görmüyorum (PG, Ö16, Kız).”*

### **Bilişim alanında bilgi sahibi olmak gerektiğini düşünme**

*“Şöyle düşünüyorum; bilgisayar konusunda yatkın insanlar var öğrenci olarak. Bunun lise ortamında verilmesi onlara gelecekte büyük bir katkı sağlar. Mesela ben reklamcılık istiyorum. Bu benim ilerde çok işime yarar. Grafik tasarım da istiyorum. Sonuçta bunu şimdiden öğrenmemin, önceden başlamış olmamın ilerdeki iş hayatıma önemli katkı sağlayacağını düşünüyorum (BG, Ö8, Kız).”*

*“Hocam bu derste gördüğümüz etkinlikleri ilerdeki zamanlarda mesela bir bilişimci bir bilişim merkezinde giriş yaptığımızda orada kullanabileceğimiz için önemli bir derstir. O yüzden bizim için işe yarayacak bir şey. İlerde programlama yaparsak mesela örneğin facebook gibi onun gibi sosyal medya uygulaması gibi bir şey üretirsek bu dilden yardım alabiliriz. İlerde işime yarayacağını düşünüyorum (BG, Ö10, Erkek).”*

*“Yani iyi bir şey bize öğretiyorlar. İlerde bize yardımcı olabilir. Şimdi bilişim dersine giriyoruz. Hocamız bilişim öğretmeni olursanız da işe yarar diyor. Ön hazırlık gibi bir şey. (PG, Ö17, Erkek)”*

### **Bilgisayarla ilgili bir şeyler öğrenildiği için bu dersi almayı isteme**

*“Bilmiyorum ilk kez gördüğüm bir ders. Ortaokulda bilgisayar dersi gördük ama programlama falan görmemiştik. Hocamız gelirdi serbest bırakırdı. Bir şey yapmazdı. Ama şimdi merak sarmaya başladım. Onun için mantıklı bence yani. Yeni şeyler öğrenmeyi seviyorum. Programlamada yeni yeni şeyler öğrendiğim için mantıklı geliyor (BG, Ö5, Erkek).”*

*“Bence çok iyi oldu çünkü hazırlık okuyoruz. Bir şeyler kazanmak için bilgisayarda. Çünkü ben geçen sene hiçbir şey bilmiyordum. Giriyordu hocamız ama o kadar detaylı anlatmadı. Bana yararı olduğunu sanmıyorum. Ama bu yıl ilk dönemde ve ikinci dönemde bilgisayar dersinin bana yararlı olduğunu anladım çünkü programların hepsini öğrendim. Gayet te iyi oldu benim için (BG, Ö6, Kız).”*

*“Arkadaşlar arasında böyle şeyler konuşuluyor biliyor musun bilmiyor musun diye. Bence bu konuda yardımcı oldu. Öğrenmek kazanç oldu biliyorum diyebiliyorsun (BG, Ö7, Kız).”*

### ***Diğer dersler gibi yorucu olmadığı için bu dersi almayı isteme***

*“Bu ders seçmeli olsaydı da evet seçerdim. kafa dinleniyor çünkü. Böyle ağır dersler seçmektense (PG, Ö14, Kız).”*

### ***Bu derse ilgisi olmadığı için istememe***

*“Bu dersi almayı istemezdim ama biraz ilgim olsaydı isterdim. Ama hiç ilgim yok (BG, Ö4, Kız).”*

*“Saçma buluyorum. Çünkü benim ilerde kullanacağım bir şey değil. Çünkü zaten bilgisayar üzerine olan şeylere karşı hiçbir ilgim yok ve karışık geliyor. “Karışık olan şeylerden de hoşlanmıyorum (PG, Ö18, Kız).”*

### ***Faydasız bir ders olarak görme***

*“Aslında hocam bu ders verilmesi gerekiyorsa veriliyordur. Bazı şeyleri saçma yani mesela. Bu derslerden kalma olasılığımız var mesela. Bundan daha öncelikli dersler de var (BG, Ö2, Kız).”*

*“Gereksiz bence. Bir faydası olacağını düşünmüyorum ( BG, Ö12, Kız).”*

### ***Eğlenceli olduğu için bu dersi almayı isteme***

*“Bu ders seçmeli olsaydı da seçerdim. Çünkü hem eğlenceli geçiyor hem de yeni şeyler öğreniyoruz bilgisayar hakkında (BG, Ö7, Kız).”*

### ***Bu dersteği bilgileri ileride kullanmayacağını düşünme***

*“Güzel bir ders ama bence gerek yok. Çünkü bilgisayarda bunlara ihtiyacımız olacağını düşünmüyorum. Hayatımın tüm süreçlerinde ihtiyacım olmayacağını düşünüyorum ( BG, Ö14, Kız).”*

### ***Derse ilgisi olmayanlara verilme zorunluluğundan hoşlanmama***

*“Gereksiz bence. Bence sadece ilgisi olan insanlara verilmeliydi böyle bir anda önümüze ders diye getirilmesi saçma olmuş. Sadece sevenler için yani merak edenler için olmalı (BG, Ö4, Kız).”*

### ***Seçmeli ders olursa bu dersi almayı istememe***

*“Öyle bir hakkımız olsaydı(bu ders seçmeli olsaydı) aslında ilk geldiğimiz kafayla almazdım. Ve yahut da alırdım. Çünkü daha yeni bir liseye başlıyoruz, daha yeni adım atıyoruz. Umursamamak var önemli derslerden kaçayım da bilgisayar dersini seçebilirdim yani (PG, Ö15, Erkek).”*

Çalışmanın nitel verilerinden elde edilen tüm bu bulgular programlama başarısı ve kaygısı ile bilişsel yük açısından her iki gruptaki öğrencileri etkileyen çok sayıda faktör olduğunu, Sosyal Bilimler Lisesi öğretim programında ilk kez yer alan bu derse yönelik öğrencilerin çeşitli açılardan hem olumlu hem olumsuz görüşlere sahip olduklarını ortaya koymaktadır. Bu sonuçların hem tasarımcılara hem öğreticilere hem de araştırmalara ileriki çalışmalarına yön vermeleri açısından zengin veri kaynağı sağladığı düşünülmektedir.

## **Bölüm Özeti**

Çalışmada, BG ve PG son test programlama kaygı seviyeleri arasında anlamlı bir fark olmadığı görülmüştür. Ancak PG kaygı seviyesinin BG kaygı seviyesinden daha yüksek olduğu tespit edilmiştir. İçsel, dışsal ve etkili bilişsel yükleri açısından BG ve PG arasında anlamlı bir farklılık olmadığı belirlenmiştir. Ancak BG içsel, dışsal, etkili ve toplam bilişsel yük seviyesinin PG'den yüksek olduğu görülmüştür. BG programlama başarısı ile PG arasında BG lehine anlamlı bir farklılık tespit edilmiştir. Bu bulgu blok tabanlı programlama aracıyla yapılan programlama öğretiminin öğrencilerin programlama başarılarına olumlu olarak yansıdığını göstermektedir. Programlama kaygısı, bilişsel yük ve akademik başarı arasındaki ilişki incelendiğinde ise değişkenler arasında, anlamlı olmayan çok zayıf düzeyde bir ilişki tespit edilmiştir. Bunun yanında bilişsel yükün alt boyutları olan içsel, dışsal ve etkili bilişsel yükün, programlama kaygısı ve akademik başarı ile anlamlı olmayan çok zayıf düzeyde bir ilişki olduğu görülmüştür. Bilişsel yükün alt boyutları arasında ise anlamlı ve çok güçlü bir ilişki tespit edilmiştir. BG ve PG öğrencilerin programlama kaygılarını, bilişsel yüklerini, akademik başarılarını etkileyen birçok faktör olduğu tespit edilmiş ve bu faktörler farklı temalar altında toplanmıştır. Programlama kaygısını etkileyen faktörler arasında öğretmenin gözünden düşmekten korkma, yapamacağını düşünme, kodları karıştırmaktan/ayıramamaktan korkma, herkesin hata yapabileceğini düşünme, öğretmenden yardım alma/çekinmeme durumu, öğrenmek için hata yapmaktan çekinmeme olduğu görülmektedir. Bilişsel yüklerini etkileyen faktörler arasında döngüleri anlamakta zorlanma, liste elemanlarını oluşturmakta zorlanma, fonksiyonları anlamakta zorlanma, dersin sıkıcı ve gereksiz olduğunu düşünme, nerede hata yaptığını anlamaya çalışma/hırslanma, tekrar tekrar deneme, hatalı kod yazmaya yönelik olumsuz duygulara kapılma (sinirlenme, sıkılma), pes etme, Blockly'de açıklayıcı/anlaşılır görsel öğeler kullanma, yeni bir programlama aracına alışmada zorlanma, örnek etkinlikler yaparak tekrar etme, programlama aracını kullanırken öğretmenden yardım alma olduğu görülmüştür. Ayrıca programlama öğretim süreciyle ilgili BG ve PG öğrencilerinin görüşleri de alınmıştır. Kodlama mantığını anlama/programlama becerisi edinme, bir yazılım oluşturma, ileriki yıllarda bu bilgilerin lazım olacağını düşündüğü için bu dersi almayı isteme, meslek seçimini olumlu etkileme, ders içeriğini saçma ve zor görme, bilişim alanında bilgi sahibi olmak gerektiğini düşünme gibi faktörlerin öne çıktığı görülmüştür.

## BEŞİNCİ BÖLÜM

### Tartışma ve Sonuç

Bu çalışmada sosyal bilimler lisesi öğrencilerine blok tabanlı programlama öğretiminin kaygı, bilişsel yük ve başarıya etkisi olup olmadığını ve değişkenler arası ilişkileri incelemek, bunun yanı sıra gruptaki öğrencileri bu değişkenler açısından etkileyen faktörleri ortaya çıkarmak ve uygulama sürecine yönelik öğrenci görüşlerini belirlemek amaçlanmıştır. Bu bölümde araştırma soruları çerçevesindeki bulgular alan yazına dayalı olarak yorumlanarak çıkarımlar ve gelecek çalışmalar için öneriler yapılmıştır. Çalışma toplam 90 öğrenciyle gerçekleştirildiği için sonuçlar bu bağlamda değerlendirilmelidir.

#### **Blok Tabanlı Görsel Programlama Aracının Kullanıldığı (BG) ve Kullanılmadığı (PG) Öğretim Ortamlarında Öğrencilerin Programlama Kaygıları**

Çalışmada elde edilen bulgularda BG ön test kaygı seviyesinin PG'den yüksek olduğu belirlenmesine rağmen BG son test kaygı seviyesinin PG'den düşük olduğu saptanmıştır. Diğer bir ifadeyle grup içi ön test ve son test kaygı seviyeleri incelendiğinde PG programlama kaygısının BG'ye göre daha çok arttığı görülmüştür. Bu durum Blockly ile yapılan blok tabanlı görsel programlama etkinliklerinin daha az kaygıya sebep olduğunu göstermektedir. Nitekim Wu, Chang ve He (2010) da benzer sonuçlara ulaşmışlar ve Scratch blok tabanlı programlama aracını kullandıkları çalışmada Scratch'ın programlama kaygısını azalttığını belirtmişlerdir. Her ne kadar Blockly bloklarla algoritma oluşturmanın yanı sıra bu algoritmanın arka planında çalışan kod yapısını da verme özelliğiyle Scratch'tan ayrılrsa da iki programlama aracının da blok tabanlı oluşu göz önünde bulundurulduğunda sonuçların benzer olduğu söylenebilir. Diğer taraftan PG grubunun kaygı düzeyinin daha fazla oluşu ise Python ile yapılan programlama etkinliklerinin metin tabanlı ve özel bir programlama yazım dili bilmeyi gerektirmesine, öğrencilerin bu dili henüz öğrenmeye başlamış olmalarına bağlanabilir. Nitekim alan yazında metin tabanlı programlama etkinliklerinin kaygıyı artırdığını vurgulayan çalışmaların olduğu bilinmektedir (Chang, 2014; Chuo, Tsai, Lan, & Tsai, 2011; Idemudia, Dasuki, & Ogedebe, 2016)

Diğer taraftan bu çalışmada farklı programlama öğretim araçları kullanılmasına rağmen gruplar arası son test kaygı seviyelerinde anlamlı bir farklılık olmadığı ancak her iki

grupta da programlama kaygısının arttığı ortaya çıkmıştır. Bu durum; nitel bulgularda da ortaya çıkan, programlama öğretimi yapılan kurum türünün sosyal bilimler temelli oluşu nedeniyle öğrencilerin ilk kez programlama dersi almalarına bağlı olarak öğrencilerde yarattığı olumsuz algıdan kaynaklanmış olabilir. Alanyazında ilk kez programlama dersi alan öğrencilerin kaygı duyduğu yönünde bulguların olduğu çalışmalar görülmektedir (Bosch & D’Mello, 2017; Bosch, D’Mello, & Mills, 2013; Freeman, Jaeger, & Brougham, 2004). Bununla beraber yine nitel bulgularda ortaya çıkan öğrencilerin bu dersi kendilerince gerekli bir ders olarak görmemelerinin de programlama kaygısı artmasına rağmen kaygı seviyelerinin düşük çıkmasında etkisi olabileceği düşünülmektedir. Ayrıca nitel bulgularda öğrencilerin kaygılarını etkileyen faktörler olarak belirlenen; programlama konusunda kendine güvenmeme ve hata yapmaktan korkma durumu programlama kaygısına neden olduğunu kanıtlamaktadır.

### **Blok Tabanlı Görsel Programlama Aracının Kullanıldığı (BG) ve Kullanılmadığı (PG) Öğretim Ortamlarında Öğrencilerin Bilişsel Yükleri**

Çalışmada her iki gruptaki öğrencilerin 10 hafta boyunca her hafta ölçülen bilişsel yük seviyelerinin hafta hafta değişkenlik gösterdiği ortaya çıkmıştır. Özellikle ilk haftalarda BG’nin bilişsel yük seviyesinin PG’den daha yüksek olduğu görülmüştür. Ayrıca her hafta işlenen konunun karmaşıklığı ve zorluk düzeyine göre öğrencilerde farklı bilişsel yüklenmelere sebep olduğu söylenebilir. Örneğin ilk 3 hafta konu yapısından dolayı her iki grup öğrencileri benzer içsel bilişsel yüke sahipken, dışsal bilişsel yüklenmelerine bakıldığında PG’nin daha az bilişsel yüklendiği görülmektedir. Bilgisayar temelli farklı bir öğretim aracı (Blockly) kullanmanın öğrencilerde “yenilik etkisi” oluşturması bu durumun sebebi olarak gösterilebilir. Bunun yanında, öğrenciler yapılan görüşmelerde en çok zorlanılan konuların döngüler ve listeler gibi karmaşık ve çok adımlı yapıların olduğu göze çarpmaktadır. Bu durum öğrencilerin listeler ve döngüler konularının işlendiği 5., 8. ve 10. hafta dışsal ve etkili bilişsel yük ortalamalarına yansımaktadır. Nitekim bu konuların yer aldığı belirtilen haftalarda BG grubunun dışsal ve etkili bilişsel yükü PG’den düşüktür. Buna göre Blockly’nin öğrencilerin zihinsel süreçlerine olumlu katkı sağladığı söylenebilir. Nitekim Çakıroğlu vd. (2018) da Scratch programlama aracını kullandıkları çalışmada öğrencilerin bilişsel yük seviyelerinin haftalara ve işlenen konuya göre değişkenlik gösterdiğini ifade etmişlerdir.

Ayrıca her iki grubun da ilk hafta yüksek olan bilişsel yük seviyelerinin özellikle 10 hafta sonunda oldukça düştüğü saptanmıştır. Bu durum öğrenim sürecinin uzunluğuna bağlı olarak öğrencilerin programlama becerisi edinmeye başlamalarından kaynaklanmış olabilir.

Alanyazında programlama öğretim sürecinde öğretim aracına aşına olunmasının bilişsel yükün düşmesi adına olumlu bir etkiye sahip olduğu ifade edilmektedir (Moons & De Backer, 2013; Sweller, 2010).

Kılıç ve Karadeniz (2004)'in belirttiği bilişsel yüklenme aralıkları dikkate alındığında her iki grubun da 10 haftalık bilişsel yük ortalamalarının düşük olduğu görülmüştür. Bu durum öğrenme sürecinin uzunluğuna bağlı olarak öğrencilerin programlama konularına aşinalık kazanmaya ve programlama becerisi edinmeye başlamış olmalarıyla açıklanabilir. Ayrıca her iki gruptaki öğrencilerin de nitel bulgularından elde edilen sınav kaygısına bağlı olarak dersi geçecek kadar çaba sarf etme ve sınav dönemlerinde derse daha çok motive olma faktörleri bilişsel yüklerinin düşük çıkmasına neden olmuş olabilir. Bu durum öğrencilerin programlama dersinde zihinsel olarak zorlandıklarını ancak farklı amaçlar çerçevesinde derse motive olduklarını göstermektedir (Tuomi, Multisilta, Saarikoski, & Suominen, 2018). Nitekim Gomes ve Mendes (2007) çalışmalarında öğrencilerin derse karşı olan tavırlarının psikolojik etkilerle ve içinde bulunulan dönemle değişebileceğini vurgulamaktadırlar.

Diğer taraftan BG ve PG öğrencilerinin 10 haftalık bilişsel yük ortalamaları arasında anlamlı bir fark bulunmazken BG'nin bilişsel yükünün PG'den daha düşük olduğu belirlenmiştir. Bu durum Yılmaz (2012)'in de belirttiği üzere programlama öğretiminde görsel olarak zenginleştirilmiş blok tabanlı öğretim araçlarının öğrencilerin daha az bilişsel yüklenmelerine neden olduğunu göstermektedir. Bunun yanı sıra ileriki çalışmalarda öznel, fizyolojik ve görev-performansa dayalı farklı bilişsel yük ölçüm teknikleri kullanılarak (Pass, 1992) gruplar arası bilişsel yük seviyesinde farklılık olu olmadığı belirlenebilir ve elde edilecek bulgularla bu çalışma sonuçları ile karşılaştırılabilir.

Bilişsel yükün alt boyutları ele alındığında içsel, dışsal ve etkili bilişsel yük açısından BG ve PG öğrencileri arasında anlamlı farklılık olmadığı belirlenmiştir. Buna göre Blockly blok tabanlı görsel programlama aracının öğrencilerin içsel, dışsal ve etkili bilişsel yüklerine etkisi olmadığı söylenebilir. Ancak Chang, Hsu ve Yu (2011) farklı öğrenme ortamlarında gerçekleştirdikleri programlama öğretiminde gruplar arasında bilişsel yük açısından anlamlı bir farklılık tespit etmişlerdir. Bu çalışmada ortaya çıkan zıt durum programlama eğitiminin 10 hafta gibi uzun bir sürece yayılmasına bağlı olarak hem Blockly ile etkinlikler yapan gruptaki hem de Python ile etkinlikler yapan gruptaki öğrencilerin bu süre boyunca programlama ile ilgili kavramlara ve konulara aşına olmaya ve programlama becerisi edinmeye başlamalarından kaynaklanmış olabilir.

Diğer taraftan BG'nin içsel, dışsal ve etkili bilişsel yükleri PG'den düşük çıkmıştır. Bu sonuç Mayer (2008)'in de vurguladığı gibi görsel olarak zenginleştirilmiş materyallerin ve

araçların bilişsel yükü azalttığına yönelik ifadelerle örtüşmektedir. Kanemune *vd.* (2004) da içsel bilişsel yükün aşırı ve uyarıcı olmadığı bir eğitim programında, öğrencilerin etkili bilişsel yükü düşürülerek programlama işlemi sırasında uzun vadeli bir şema oluşturulabileceğini belirtmişlerdir. Bu çalışmada BG'nin içsel bilişsel yükünün yanı sıra etkili bilişsel yükünün de PG'ye göre düşük çıkması BG'de kullanılan Blockly aracının yapısı gereği programlama öğretimindeki karmaşıklığı azaltabilecek potansiyele sahip oluşundan da kaynaklanmış olabilir (Culic, Radovici, & Vasilescu, 2015). Ayrıca Python metin tabanlı programlama öğretim aracıyla yapılan etkinliklerde programlama dillerinin doğasından kaynaklanan karmaşık yapısı (Weintrop & Wilensky, 2015) PG öğrencilerinin daha çok zihinsel çaba göstermelerine neden olmuş olabilir. Alan yazında da öğrenilecek konunun karmaşıklığının içsel bilişsel yüklenmeyi artırdığı belirtilmiştir (Çakmak, 2007; Mason, Cooper, & Wilks, 2015; Sweller, 2010). Ayrıca nitel bulgulardan elde edilen PG öğrencilerin hatalı kod yazdıklarında sinirlenmeleri ve kod yazmaktan vazgeçmeleri de içsel bilişsel yüklenmelerine neden olmuş olabilir. Bu duruma benzer sonuçlar alanyazında görülmektedir (Mason *vd.*, 2015; Moons & De Backer, 2013).

Bunun yanı sıra Python programlama aracının sadece metin tabanlı bir programlama arayüzüne sahip oluşu, programlama dilinde geçen kavramların yabancı bir dil oluşu ve öğrencilerin bu araçla ilk kez karşılaşmış oluşu PG'nin BG'ye göre daha çok dışsal bilişsel yüklenmelerine neden olmuş olabilir. Moreno ve Mayer (1999) da bu sonucu destekler şekilde bir öğrenme ortamının tek bir boyuttan oluşmasının öğrencilerin zihinsel süreçlerinde zorlanmalarına neden olabileceğini belirtmişlerdir. Kölling, Brown ve Altadmri (2015) çalışmalarında metin tabanlı programlama araçlarında, İngilizce'de varolan anahtar kelimeleri kullanmalarının acemi programcılar için güçlük meydana getirdiğini, blok tabanlı araçlarda ise bu kavramların birçok dile çevrilebildiğini ifade etmişlerdir. BG'de ise düzenlenebilir görsel hazır kod bloklarıyla desteklenmiş programlama ortamının kod yazmayı somutlaştırmaya yardımcı oluşu bu gruptaki öğrencilerin daha az dışsal bilişsel yüklenmelerini açıklar niteliktedir (Harms, 2013). Ancak her iki grupta da dışsal bilişsel yükün içsel ve etkili bilişsel yükten yüksek çıktığı belirlenmiştir. Buna göre her ne kadar grupların birinde blok tabanlı ve görsel bir programlama aracı kullanılsa da programlama aracından kaynaklanan bir dışsal bilişsel yük oluşumundan söz edilebilir. Bu durum göz önünde bulundurularak ileriki çalışmalarda öğrencilerin dışsal bilişsel yüklenmelerinin azaltılması amacıyla farklı nesne tabanlı görsel programlama araçlarının kullanıldığı araştırmalar yapılabilir.



Alanyazındaki dıřsal biliřsel ykn alıřma belleğinde azaltılarak ynetilmesinin etkili biliřsel yke yer aacađı (Dntolittle, McNeill, Terry, & Scheer, 2005) ifadesini destekler nitelikte bu alıřmada BG'nin dıřsal biliřsel yknn yanı sıra etkili biliřsel yk de PG'ye gre daha dřk ıkmıřtır. Benze řekilde Zheng, McAlack, Wilmes, Kohler–Evans ve Williamson (2009) alıřmalarında ynetilebilir đrenme ortamı ve srelerinin biliřsel yklenmeyi azaltarak problem zme seviyesini arttırdıđını ifade etmiřlerdir. Bu noktada etkinliklerin iyi planlanması ve đrenme ortamındaki gereksiz bilgilerin azaltılmasının bu durumu kontrol altına almada kolaylık sađlayacađı dřnlmektedir. Shaffer *vd.*, (2010) da đretim ortamlarında kullanılan grselleřtirme aralarının đrencilerin đrenme faaliyetlerine yardımcı olabileceđini ifade edilmiřtir. Bu durum Blockly grsel programlama aracı kullanılan grubun etkili biliřsel yknn metin tabanlı programlama aracı kullanan đrencilerinkinden dřk ıkmasını sađlamıř olabilir.

### **Blok Tabanlı Grsel Programlama Aracının Kullanıldıđı (BG) ve Kullanılmadıđı (PG) đretim Ortamlarında đrencilerin Programlama Bařarıları**

Her iki grubun n test programlama bařarısının dřk olduđu ve gruplar arasında anlamlı bir farkın olmadıđı belirlenmiřtir. Bu sonu grupların benzerliđini kanıtlamaktadır. Gruplar arası son test programlama bařarıları arasında ise BG grubu lehine anlamlı bir fark olduđu grlmřtr. Buna gre blok tabanlı programlama aracıyla yapılan đretimin metin tabanlı programlama aracıyla yapılandan daha etkili olduđu sylenebilir. Alanyazında farklı đrenim seviyesinde đrencilerle ve eřitli grsel programlama aralarıyla yapılsa da bu bulguyu destekler nitelikte ok sayıda alıřma olduđu belirlenmiřtir (Chang, 2014; Diner, 2018; Korkmaz, 2016; Park, Hyun, Jin, & Jeong, 2015; Su, Huang, Yang, Ding & Hsieh, 2015; Sykes, 2007; Wang *vd.*, 2009; Yiđit, 2016; Yksel & Gndođdu, 2018; Zhong, Wang & Chen, 2016; Wang, Huang, & Hwang, 2014). Blok tabanlı programlama aralarının metin tabanlı programlama aralarına gre daha renkli ve eđlenceli bir đrenme ortamı sunması, programlama dilinin szdizimsel hatalarının blok tabanlı ortamlarda olmaması (Demir, 2015; Kaleliođlu & Glbahar, 2014; Kayabařı, 2014; Koorsse, Cilliers, & Calitz, 2015) ve Blockly'nin oluřturulması gereken algoritmayı hem grsel kod bloklarıyla hem de seilen programlama dilinin szdizimsel yapısıyla grmeye imkan tanınması bilgiyi somutlařtırmaya yardımcı olduđundan (Culic, Radovici, & Vasilescu, 2015) BG grubunun programlama bařarısını artmıř olabilir.

Grupların kendi iindeki n test ve son test programlama bařarıları arasında her iki grupta da anlamlı farklılık olduđu grlmřtr. Buna gre ister metin tabanlı ister blok tabanlı olsun yapılan programlama etkinliklerinin đrencilerin đrenmesine katkı sađladıđı

söylenbilir. Nitekim Erol (2017) da yaptığı çalışmada bu sonuçları destekleyici bulgulara ulaşmıştır. Bu çalışmanın nitel bulguları arasında yer alan her iki gruptaki öğrencilerin öğrendikleri bilgilerle kendi tasarladıkları kodları yazabilmeyi isteme, ileriki yıllarda bilişimle ilgili diğer derslerde kullanmak için hazırlık yapmayı isteme gibi faktörler de öğrencilerin başarılı olmalarını olumlu etkilemiş olabilir. Bu bulguların öğrencilerde programlamaya karşı motivasyon kaynağı oluşturarak onların programlama başarısını olumlu yönde etkilediği söylenebilir.

### **BG ve PG Öğrencilerinin Programlamaya Yönelik Kaygıları, Bilişsel Yükleri ile Akademik Başarıları Arasındaki İlişki**

Çalışmada her iki grupta da içsel, dışsal ve etkili bilişsel yükün pozitif yönde yüksek düzeyde birbiriyle ilişkili olduğu saptanmıştır. Bu çalışmada programlama öğretiminin 10 haftalık uzun bir sürece yayılması öğrencilerin programlama kavramlarına aşina olmalarına ve programlama becerilerini pratiğe dökmelerine imkan sağlamış olabilir. Bu durum bilişsel yükü olumlu etkileyerek hem içsel, hem dışsal hem de etkili bilişsel yükün birbirleriyle ilişkili olarak düşük çıktığını göstermektedir. İleride programlama öğretiminin optimum düzeyde uzun bir sürece yayılması göz önüne alınarak çalışmalar yapılabilir.

Çalışmada her iki grupta da programlama kaygısı, bilişsel yük ve programlama başarısı arasında anlamlı ilişki olmadığı saptanmıştır. Ancak bilişsel yük zihinsel süreçleri zorlayan bir unsur olduğu için programlama kaygısını tetikleyebileceği düşünülmektedir, Rogerson ve Scott (2010) da programlama korkusunun öğrenci kazanımına etkisi olduğunu belirtmesine rağmen bu çalışmadan elde edilen bulgu şaşırtıcıdır. Fakat çok zayıf da olsa pozitif yönde bir ilişki olmasının beklenen bir durum olduğu düşünülmektedir. Çavuş ve Günbatar (2008) çalışmalarında belirttiği gibi kaygı bireylerin yaratıcı ve yapıcı bir hareketi engelleyen unsurların tetiklediği duygu olarak nitelenmektedir. Bilişsel yüklenmenin ve programlama kaygısı oluşumunun öğretim sürecinde aynı anda olabilmesi bu durumu desteklemektedir. Ayrıca her ne kadar nitel bulgularda öğrenciler ilk kez programlama dersi aldıkları için kaygı duyduklarını belirtse de programlama öğretiminin 10 haftalık belli bir sürece yayılarak ve basitten karmaşığa doğru ilerleyerek yapılması kaygı artışını engellemiş olabilir. Diğer taraftan her iki gruptaki öğrencilerin programlama kaygısı yerine sınav kaygısı yaşadıkları ve bu nedenle dersi geçmeye yetecek kadar çaba sarf ettikleri nitel bulgulardan ortaya çıkmıştır. Bu durumlar programlama kaygısı ile bilişsel yük arasında ilişki çıkmamasına neden olmuş olabilir. Aynı zamanda dersi geçmeye yönelik çabanın bilişsel yükü engellemiş olabileceği düşünüldüğünde bu durum; programlama başarısının bilişsel yükü ilişkili çıkmamasını açıklayabilir.

## **BG ve PG Öğrencilerinin Programlama Öğretim Sürecindeki Kaygılarını, Bilişsel Yüklerini ve Akademik Başarılarını Etkileyen Faktörler**

### **Programlama kaygısını etkileyen faktörler.**

Her iki grupta da ortaokuldaki BTY dersi kapsamına blok tabanlı algoritma yazmaya yönelik ders alan ve almayan öğrenciler olduğu tespit edilmiştir. Dersi alan öğrencilerin bu derse yönelik kaygıları olmadığı ancak dersi daha önce almayan öğrencilerin programlama kaygısına kapıldıkları görülmüştür. Ders hakkında bilgi sahibi olmamanın bu durumun doğal bir sonucu olduğu söylenebilir. Diğer taraftan her iki grupta da bilgisayar kullanmayı seven ve sevmeyen öğrenciler olduğu, bu durumun BG ve PG’de farklı programlama aracı kullanımından bağımsız olarak öğrencilerin derse olan ilgileri ile programlama kaygılarını olumlu veya olumsuz etkilediği söylenebilir. Gürcan-Namlu ve Ceyhan (2003) bilgisayar kaygısını etkileyen faktörleri tespit etmek için yaptıkları çalışmalarında kaygıya sebep olan en büyük etkenin kişinin kendini bilgisayarda yeterli görmesi algısı olduğunu ifade etmişlerdir. Bu çalışmada da bilgisayar kaygısı dolaylı olarak programlama kaygısını etkilemiş olabilir.

Elde edilen temalardan biri de öğrencilerin derste başarısız olacakları yönünde bir inancıya sahip oluşlarıdır. Bu düşünce öğrenme sürecinin başında öğrencilerde programlama kaygısı oluşturmuş olabilir. Nitekim Charleson (2012) kişinin kendi yeteneklerine olan inanç eksikliğinin başarısızlık korkusuna dayanarak kaygı duygusunu oluşturduğunu belirtmiştir. Bunun yanı sıra hem BG hem de PG öğrencileri hata yapmaktan çekindiklerini, öğretmen ve arkadaş çevresinin bunda etkili olduğunu, arkadaşlarıyla kıyaslanmanın kendisinde sıkıntı yarattığını belirtmişlerdir. Connolly *vd.* (2007) programlama dilinde sözdizimi yapısı nedeniyle yapılan hataların varlığına işaret etmiş ve bu yapıların programlama kaygısına neden olduğunu ortaya koymuşlardır. Cevahir ve Özdemir (2017) öğretmenlerle yaptıkları görüşmelerde öğrencilerin programlama öğrenme sürecinde tek başlarına uygulama yapamadıklarını, bunun sonucunda öğrencilerin çekince ve özgüven eksikliği duyduklarını ve öğretimin bu durumdan olumsuz etkilendiğini ifade etmişlerdir. Rogerson ve Scott (2010) da dış faktörlerin kaygıyı artırdığını vurgulamıştır.

Diğer taraftan her iki gruptaki öğrencilerin büyük çoğunluğu programlama öğrenme sürecinde hata yapmaktan korkmadıklarını, herkesin hata yapabileceğini, programlamanın hata yaparak öğrenilebilecek bir beceri olduğunu belirttikleri görülmüştür. Bu durum her iki gruptaki öğrencilerin son test programlama kaygılarının düşük çıkmasının nedeni olarak gösterilebilir. Alanyazında bu durumu destekler nitelikte çalışmalar yer almaktadır (Connolly *vd.*, 2007; Owolabi, Adedayo, & Olayanju, 2018; Zainal *vd.*, 2012). Ayrıca programlama öğrenme sürecinin 10 haftalık uzun bir sürece yayılmasıyla öğrencilerin programlama

kavramlarına aşinalık kazanmış olmalarının da bu durumda payı olduğu düşünülmektedir. Bu nedenle ileride yapılacak çalışmaların basitten karmaşığa doğru etkinliklerin yer aldığı uzun bir sürece yayılmasının programlama kaygısını azaltmak açısından önemli olduğu söylenebilir. Nitekim alanyazında bu durumun önemi vurgulanmaktadır (Katai & Toth, 2010; Lahtinen, Ala-Mutka ve Jarvinen, 2005).

### **Bilişsel yükü etkileyen faktörler.**

Her iki gruptaki öğrenciler kodlama sürecinde hatalı kod yazdıklarında sinirlenme ve sıkılma gibi olumsuz duygulara kapıldıklarını, bu durumda bazıları hırslanma, bazıları ise sinirlenme ve pes etme gibi çeşitli reaksiyonlar verdiklerini ifade etmişlerdir. Alanyazında benzer sonuçlara rastlanmıştır (Li, 2016, Moons & De Backer, 2013). Programlama kodunda yaptıkları hatanın sebebini bulmakta zorluk çektiklerini, hataları düzeltmek için tekrar tekrar deneme yapmak zorunda kaldıklarını söyleyen bazı öğrenciler olduğu görülmüştür. Bu durum öğrencilerde içsel bilişsel yüklenmeye neden olmuş olabilir. Nitekim konu içeriğinin karmaşıklığına göre bilişsel yükün artabileceği alanyazında vurgulanmıştır (Çakmak, 2007; Sweller *vd.*, 2011).

Her iki gruptaki öğrenciler de programlama dillerinin karmaşık olduğunu, özellikle fonksiyonlarla ilgili konuları anlamakta zorluk çektiklerini belirtmişlerdir. Programlama içeriğinin doğal yapısından kaynaklanan bu durumun programlama öğrenimini güçleştirdiği (Ozoron, Çağiltay, & Topallı, 2012) ve öğrencilerde içsel bilişsel yük oluşturduğu söylenebilir. Nitekim Paas ve Van Merriënboer (1994) görev ortamı ve konu özellikleri arasındaki etkileşimin içsel bilişsel yükü etkilediğini ifade etmişlerdir. PG öğrencileri genel olarak akış şemalarını/algoritmaları ve karar yapılarını anlamakta zorlanırken BG öğrencileri bu durumu özelleştirerek döngüleri ve listeleri anlamakta zorluk çektiklerini ifade etmişlerdir. Farklı konularda öğrencilerin zihinsel olarak zorlanma durumları öğretim materyalinin blok yapısının tam anlaşılmadığından ya da aşına olunamadığından kaynaklandığı şeklinde yorumlanabilir (Korbach, Brünken, & Park, 2018). Cevahir ve Özdemir (2017) de programlama eğitiminde en çok zorlanılan konuları döngüler, diziler, if-else (karar) yapıları ve algoritma akış şemaları olarak ifade etmişlerdir. Bu sonuçlar öğrencilerdeki zihinsel zorlanmanın çalışmada kullanılan programlama araçlarına özgü bir durum olmadığını yapısal bir durum olduğunu göstermektedir.

Diğer taraftan her iki gruptaki öğrenciler 10 haftalık süreçte çeşitli etkinlikler yaparak programlama sözdizimlerini tekrar tekrar kullandıklarını böylece programlama diline alıştıklarını belirtmişlerdir. Programlamaya yeni başlayan acemi programcıların kullanılan bu

etkinlikler sayesinde kendilerini hızla geliştirme imkânı buldukları söylenebilir (Idrees, Aslam, Shahzad, & Sarwar, 2018).

Her iki gruptaki öğrenciler programlama aracından kaynaklanan çeşitli zorluklar yaşadıklarını belirtmişlerdir. BG öğrencileri yeni bir araca alışmakta zorlandıklarını ve aracın dilinin İngilizce oluşunun bu durumu daha da olumsuz etkilediğini belirtmişlerdir. Alanyazında programlama dillerindeki kavramların yabancı dille ifade edilmesinin programlama öğrenmede zorluklara sebep olabileceği vurgulanmıştır (Arabacıoğlu, Bülbül, & Filiz, 2007; Esteves, Mendes, 2004; Ozaran, Çağıltay, & Topallı, 2012). Bu durum öğrencilerin dışsal olarak bilişsel yüklenmelerini göstermektedir. Ancak öğrencilerin haftalık bilişsel yük seviyeleri incelendiğinde programlama öğrenme sürecinin sadece ilk haftalarında bilişsel yüklenmenin yüksek olduğu görülmüştür. İlerleyen haftalarda ise bilişsel yükte düşüş yaşandığı belirlenmiştir. Bu durumun programlamaya yeni başlayan öğrencilerin programlama kavramlarına aşina olmamalarından kaynaklanmış olabileceği ve ilk kez programlama dersi aldıkları için bu durumun normal karşılanması gerektiği söylenebilir. Ancak PG öğrencilerinden bazıları Python dilinde kodlama yapmakta zorlandıklarını vurgulamışlardır. Bu durumun kullanılan programlama aracının programlama sözdizimi bilgisi gerektirmesine bağlı olduğu düşünülmektedir. Alanyazında metin tabanlı programlama araçlarının daha çok zihinsel çaba gerektirdiği vurgulanmıştır (Esteves & Mendes, 2004; Lahtinen, Ala-Mutka & Jarvinen, 2005; Ozoran, Cagiltay & Topalli, 2012).

Blockly blok tabanlı görsel programlama aracının yapısından kaynaklanan açık ve anlaşılır öğeler, hazır kod yapıları, biçimlendirilebilir kod blokları gibi unsurlardan ötürü kullanım kolaylığı sağladığı BG öğrencilerinin görüşlerinden anlaşılmıştır. Alanyazında bu bulguyu destekler nitelikte çalışmalar olduğu görülmektedir (Mladenović, Boljat & Žanko, 2018; Weintrop & Wilensky, 2018). Bu durum öğrencilerin dışsal bilişsel yüklerinin düşük çıkmasını sağlamış olabilir (Hromkovič, Serafini & Staub, 2017). PG öğrencileri ise bilgisayarda kod yazmanın kâğıda kod yazmaktan daha kolay olduğunu düşündükleri için Python programlama aracının kolaylık sağladığını ve öğretmenden destek aldıkları için programı kullanmakta çok zorlanmadıklarını ifade etmişlerdir. Bu nedenle öğrencilerin dışsal bilişsel yükleri düşük çıkmış olabilir. Her iki grup öğrencilerden bazıları dersi sıkıcı ve gereksiz olarak gördüklerini ve dersi dinlemediklerini ifade etmişlerdir. Cevahir ve Özdemir (2017) bu durumun programlama öğretimine olumsuz etkilediğini belirtmişlerdir. Uygulama yapılan okul türünün sosyal bilimler lisesi olması öğrencilerdeki bu algının sebebi olabilir. Ayrıca ortaöğretim sınavlarında sözel ağırlık ya da sayısal ağırlık gibi bir puan türü olmadan öğrencilerin bu okullara girdiği fakat bu durumdan dolayı öğrencilerin kendilerini bir alana

yakın hissettiklerini ifade etmeleri bu savı kuvvetlenmektedir. Tüm bu görüşlerin etkili bilişsel yüke olumsuz yönde etki yaptığı, hatta PG'yi çok daha fazla olumsuz etkilediği söylenebilir (Salleh, Shukur, & Judi, 2018). Ayrıca her iki gruptaki öğrencilerin derse motive olma sebepleri arasında sınav kaygısı ve dersi geçme amacı olduğu belirlenmiştir. Bu durum öğrencilerin bilişsel yüklerinin düşük çıkmasına neden olan önemli bir faktör olarak gösterilebilir.

Diğer taraftan her iki gruptaki bazı öğrencilerin Python programlama diliyle kod yazma sürecinde öğretmenin desteğine veya arkadaşlarının yardımına ihtiyaç duydukları belirlenmiştir. Bu bulgular öğrencilerin bilişsel yüklerinin düşük çıkmasını açıklayabilecek kanıt niteliğindedir. Ayrıca özellikle PG öğrencilerinden bazıları bilgisayar kullanmayı sevmediklerini veya bilgisayar kullanmadıklarını ifade etmişlerdir. Bu durum programlama başarısının BG grubu lehine yüksek çıkmasının kanıtı olarak gösterilebilir.

### **Programlama başarısını etkileyen faktörler.**

Her iki gruptaki öğrencilerden bazıları elde edilecek programlama becerisiyle belirli bir amaca hizmet eden kendi istedikleri kodu yazabilmeyi, aynı zamanda gelecek yıllarda karşılıklarına çıkacak olan derslerde bu kazanımlarını kullanabilmeyi istediklerini ifade etmişlerdir. Bu bulguların öğrencilerin başarılarındaki artışı etkileyen faktörlerden olduğu söylenebilir. Bu durum alan yazında da desteklenmektedir. Lye ve Koh (2014) programlama eğitiminin bir dil aracılığıyla bireylerin problem çözme becerisini ve çözüme ulaşma isteğini arttırdığını ifade etmişlerdir. Diğer taraftan PG öğrencilerinden bazıları kendi kendilerine kod yazmakta sorun yaşadıklarını belirtmişlerdir. Bu durum BG'nin programlama başarısının PG'den yüksek oluşunun sebebi olabilir. Alanyazında bu durumu kanıtlar bulgulara rastlanmıştır (Brown, Altadmri, & Kölling, 2016; Weintrop & Wilensky, 2015). Ayrıca BG öğrencilerinden bazıları niçin programlama dili öğrenmeleri gerektiğini anlamadıklarını ifade etmişlerdir. Bu durum öğrencilerin sosyal bilimler alanında öğrenim görmelerinden kaynaklanmış ve programlama başarılarını etkilemiş olabilir. Bu dersin gerekliliği uygulama başında öğrencilere açıklanmış olmasına rağmen öğrencilerin olumsuz görüş belirtmeleri; programlama öğrenimi gerekliliğinin bir kültür haline gelebilmesinin zaman alıcı olduğunun göstergesidir.

### **BG ve PG Öğrencilerinin Programlama Öğretim Sürecine Yönelik Görüşleri**

Her iki gruptaki öğrenciler programlama öğretim sürecine yönelik hem olumlu hem olumsuz görüşler belirtmişlerdir. PG öğrencilerinden bazıları bir problemin çözümü için gereken adımları ve hatalardan ders çıkarmayı öğrendiklerini belirtmişlerdir. BG

öğrencilerinden bazıları ise kodlama mantığını öğrendiklerini ve programlama becerisi edindiklerini ifade etmişlerdir. Bu bulgular öğrencilerin programlama kaygılarını azaltmaya teşvik olarak gösterilebilir. Her iki grup öğrencilerinin de sahip olduklarını düşündükleri becerilere bakıldığında bilgi işlemsel düşünme becerileri içerisinde yer alan kazanımlar olduğu görülmektedir (TTKB, 2017). BG öğrencilerinden bazıları ise bu derste hiçbir kazanım elde etmediklerini söyleyerek derse karşı olumsuz bakış açılarını gözler önüne sermişlerdir. Öğrencilerin sosyal bilimler alanında öğrenim görmelerinin bu durumun sebebi olabileceği söylenebilir. Yine öğrenci görüşlerinden bilgisayar bilimi dersinin öğrenci inisiyatifine göre alınmasına yönelik verilen cevaplarda öğrencilerin ikiye ayrıldıkları saptanmıştır. Cevahir ve Özdemir (2017)'nin de çalışma sonuçlarıyla desteklendiği üzere bu çalışmada bir kısım öğrenci derse ilgisinin olmaması nedeniyle böyle bir durumda dersi tekrar almayacaklarını ifade ederken, bir kısım öğrenci bilgisayar ve eğlence gibi faktörleri içerdiğinden dersi tekrar alabileceklerini belirtmişlerdir.

Öğretim sürecini değerlendiren her iki grup öğrencilerinin büyük çoğunluğu bilişim alanında bilgi sahibi olmak için bu dersi gerekli gördüğünü ve bu derin ileriki yıllarda meslek seçimini etkileyeceği belirtmişlerdir. Bu durum öğrencilerin derse motive olmalarına ve son test başarılarına katkı sağlamış olabilir. Ayrıca öğrencilerin dışsal bilişsel yüklenmelerini azaltan faktörler arasında gösterilebilir. Diğer taraftan her iki gruptaki öğrencilerin bazıları bu dersin saçma ve zor olduğunu, BG öğrencilerinden bazıları ise faydasız bir ders olarak gördüklerini belirtmişlerdir. Bu durum öğrencilerin programlama öğretimiyle ilgili olarak olumsuz bir algıya sahip olduklarını göstermektedir. Çalışmanın yapıldığı okul türünün sosyal bilimler lisesi olmasının, dersin ilk defa bu okul türünde zorunlu bir ders olarak okutuluyor olmasının (MEB, 2017) ve bu seviyeye gelmeden önce öğrencilerin birçoğunun programlama deneyimine sahip olmamasının bu görüşleri oluşturmuş olabileceği düşünülmektedir. Ancak bu bakış açısı; sosyal bilimler alanlarındaki öğrencilerin de 21. yy öğrencileri arasında yer aldıklarından mantıksal ve algoritmik düşünme becerilerine sahip olmaları gerektiği gerçeğiyle çelişmekte ve bu temelsiz algının yıkılması gerektiği düşünülmektedir. Bunun yanı sıra çalışmanın yapıldığı ders olan bilgisayar bilimi dersinin sosyal bilimler okul türünde hazırlık sınıfında zorunlu, sonraki sınıflarda seçmeli olmasının öğrencilerin konunun devamlılığının olmayacağını düşünmelerine neden olduğu söylenebilir. Bu durum dikkate alınarak ileriki yıllarda MEB'in öğretim programlarında, her sınıf seviyesinde programlama becerisini geliştirmek amacıyla derslerin verilmesine yönelik düzenlemeler yapılması önerilmektedir.

Öğrencilerin bazıları ileriki yıllarda bu bilgilerin lazım olacağını, bilişim alanında bilgi sahibi olmak gerektiğini düşündüğünden, hatta bilgisayarla ilgili bir şeyler öğrenildiği, eğlenceli olduğu için bu dersi almayı istedikleri belirlenmiştir. Bu faktörlerin öğrencilerin derse ilgilerini olumlu etkilediği düşünülmektedir. Bunun yanında, ders içeriğini saçma ve zor olarak görme, faydasız bir ders olarak görme, derse ilgisi olmayanlara verilme zorunluluğundan hoşlanmama, seçmeli ders olursa bu dersi almayı istememe gibi faktörlerin öğrencilerin bu derse olumsuz bakış açılarını ortaya koymaktadır. Bu nedenle ileriki çalışmalarda bu faktörler dikkate alınarak araştırma yapılmasının önemli olduğu söylenebilir.

Bu çalışma süresince ve sonucunda tespit edilerek alanyazınla desteklenen tüm bulgular; hem blok tabanlı programlama öğelerini içeren hem de Python gibi çeşitli programlama dillerinin sözdizimlerini gösteren Blockly görsel programlama aracıyla yapılan programlama öğretiminin öğrenci başarısını olumlu etkilediğini göstermektedir. Alanyazında programlama öğretimine yönelik blok tabanlı araçlarla yapılan çeşitli çalışma bulunmasına rağmen bu çalışmanın Blockly aracıyla yapılan az sayıda araştırmadan biri olması nedeniyle özgün ve değerli olduğu söylenebilir. Blockly'nin öğrencilerin programlama kaygısını azaltmasa da metin tabanlı bir programlama aracına göre olumlu katkı yaptığı, üç alt boyutuyla incelenen bilişsel yükte ise öğrencilerin tüm alt boyutlarda Python programlama aracına göre daha düşük seviyede bilişsel yüklendikleri görülmektedir. Bu nedenle programlama öğrenimi sürecinde bilişsel yükün düşürülmesi için bu değişkenin tüm alt boyutlarıyla birlikte ele alınarak içsel ve dışsal bilişsel yükün kontrol altında tutulmasının önemli olduğu söylenebilir (Paas, Tuovinen, Tabbers, & Van Gerven, 2003). Bu çalışmada bilişsel yükün alt boyutlarını da etkileyen faktörlerin belirtilmiş olması ileriki araştırmalara yön verme açısından önemli görülmektedir. Ayrıca çalışmanın, yapıldığı okul türünde (sosyal bilimler lisesi) bir programlama dili içeren dersin zorunlu olarak ilk defa verilmesi, bu süreçte dikkat edilmesi gerekenleri gözler önüne sermesi ve alan yazında bu konuda çalışmanın çok az olması nedeniyle çalışmanın alan yazına katkı sağlayacağı ve gelecek çalışmalara rehber olacağı düşünülmektedir.

Özet olarak görsel olarak zenginleştirilmiş ve öğrencilerin zihinsel olarak yaşayabilecekleri zorlukları azaltmak amacıyla hem görsel kod bloklarını hem de çeşitli programlama dillerinin sözdizimlerini bir arada sunan Blockly aracının programlama becerilerini kazandırmaya, programlama kaygısını ve bilişsel yükü azaltmaya katkı sağlayacağı ancak programlama öğretimi sürecinin devamlılığının olmayışı nedeniyle “bir ileri iki geri” şeklinde bir yaklaşımın öğrencilerin programlamaya bakış açısını olumsuz etkileyebileceği düşünülmektedir.



## Öneriler

Bu çalışma bulguları ile uygulama öncesi, süreci ve sonrasında yaşananlar dikkate alınarak öğretim tasarımcılarına ve uygulama/materyal/içerik geliştiricilere, uygulayıcılara ve araştırmacılara yönelik bazı öneriler aşağıda sunulmuştur.

- Hem görsel kod bloklarının olduğu hem de farklı programlama dillerinin sözdizimlerinin gösterilebildiği Blockly blok tabanlı görsel programlama aracının kullanılması programlama öğretiminde somut örnekler sunmanın yanı sıra geliştirilen algoritmanın arka planında çalışan kod mantığını görerek ikisi arasında ilişki kurmayı kolaylaştırabilme potansiyeline sahip olduğundan bu yöndeki çalışmaların artırılması önerilmektedir.
- Blockly programlama aracının kullanılması; öğretilen programlama dilindeki kod yapısını anlamlandırmayı kolaylaştırarak ve öğrencilerin hata yapmasının önüne geçerek içsel bilişsel yükü ve programlama kaygısını azaltmak açısından önemli görülmektedir.
- Özellikle programlama dillerinde zorlanılan döngüler, listeler, karar yapıları gibi karmaşık ve çok adımlı konularda gerçek yaşamla ilişkili somut örnekleri içeren Blockly etkinlikleri yapılarak konuların daha kolay kavranması sağlanabilir.
- Farklı programlama dillerinin birçok ortak noktası olduğu ve aynı amaca hizmet edebilen uygulamalar geliştirilebileceği düşünüldüğünde Blockly programlama aracı kullanılarak çeşitli programlama dilleriyle çalışmalar yapılabilir, farklı değişkenler açısından bu çalışmalar karşılaştırılabilir.
- Bilişsel yük değişkeni fiziksel, görev, performansa dayalı çeşitli ölçüm teknikleriyle ölçülerek elde edilen bulgular ışığında sonuçlar irdelenebilir.
- Programlama kaygısının programlama eğitiminde önemli bir değişken olduğu göz önüne alınarak sadece sosyal bilimler lisesi gibi sosyal içerik ağırlıklı eğitim veren ortaöğretim kurumlarında değil aynı zamanda diğer kurumlarda da bu değişkenin irdelendiği araştırmalar yapılabilir.
- Programlama öğretim aracı dışında öğretim içerikleri ve konular üzerine de bilişsel yük değişkeninin etkisi incelenebilir.
- Süreç temelli değerlendirme yöntemlerinin öğrencilerin not kısılacından kurtulmalarına yardımcı ve programlama başarılarına katkısı olup olmadığı araştırılabilir.
- Öğretim programında bilgisayar bilimi dersi olan diğer lise türlerinde de Blockly aracıyla programlama öğretimi yapılarak grupların programlama başarıları karşılaştırılabilir.

- Programlama kaygısının temeli olan çekirdek inançları doğru yönetebilmek adına ortaokul düzeyinden önce programlama öğretimine başlanmasının önemli olduğu düşünülmektediru yönde yeni araştırmalar yapılarak sonuçlar karşılaştırılabilir.



## KAYNAKÇA

- Abdul-Rahman, S.-S., & Du Boulay, B. (2014). Learning programming via worked-examples: Relation of learning styles to cognitive load. *Computers in Human Behavior*, 30, 286-298.
- Achim, N., & Al Kassim, A. (2015). Computer usage: The impact of computer anxiety and computer self-efficacy. *Procedia-Social and Behavioral Sciences*, 172, 701-708.
- Akbulut, Y. (2010). *Sosyal bilimlerde SPSS uygulamaları: Sık kullanılan istatistiksel analizler ve açıklamalı SPSS çözümleri*. İstanbul: İdeal Kültür Yayıncılık.
- Akpınar, Y., & Altun, A. (2014). Bilgi toplumu okullarında programlama eğitimi gereksinimi. *İlköğretim Online*, 13(1).
- Al-Tahat, K. (2014). An innovative instructional method for teaching object-oriented modelling. *International Arab Journal of Information Technology (IAJIT)*, 11(6).
- American Association of School Librarians. (2018). AASL Standards Framework for Learners. <http://standards.aasl.org/wp-content/uploads/2017/11/AASL-Standards-Framework-for-Learners-pamphlet.pdf> adresinden edinilmiştir.
- Anderson, A. A. (1996). Predictors of computer anxiety and performance in information systems. *Computers in Human Behavior*, 12(1), 61-77.
- Arabacıoğlu, T., Bülbül, H. İ., & Filiz, A. (2007). Bilgisayar programlama öğretiminde yeni bir yaklaşım. *Akademik bilişim*.
- Armoni, M., Meerbaum-Salant, O., & Ben-Ari, M. (2015). From scratch to “real” programming. *ACM Transactions on Computing Education (TOCE)*, 14(4), 25.
- Aschwanden, C., & Stelovsky, J. (2003). Measuring cognitive load with EventStream software framework. In System Sciences, 2003. Proceedings of the 36th Annual Hawaii International Conference on (pp. 5-pp). IEEE.ISO 690
- Ayersman, D. J., & Michael Reed, W. (1995). Effects of learning styles, programming, and gender on computer anxiety. *Journal of Research on Computing in Education*, 28(2), 148-161.
- Bak, N., Chang, B.-M., & Choi, K. (2018, Temmuz). *Smart block: a visual programming environment for smarthings*. Paper presented at the 2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC).
- Balanskat, A., & Engelhardt, K. (2014). *Computing our future: Computer programming and coding-Priorities, school curricula and initiatives across Europe*: European Schoolnet.
- Baltalı, S. (2016). *Programlama öğretiminde kullanılacak yazılımlara ilişkin öğretmen görüşleri*. (Yüksek Lisans tezi). Yükseköğretim Kurulu Ulusal Tez Merkezi'nden edinilmiştir. (Tez No. 445158)
- Baltas, A. (2002). *Öğrenmede ve sınavlarda üstün başarı*. İstanbul: Remzi Kitabevi Yayınları.
- Barron, A. E. (2004). Auditory instruction. *Handbook of research on educational communications and technology*, 2, 949-978.

- Basawapatna, A. (2016). Alexander meets Michotte: A simulation tool based on pattern programming and phenomenology. *Journal of Educational Technology & Society*, 19(1), 277-291.
- Basogain, X., Olabe, M. Á., Olabe, J. C., & Rico, M. J. (2018). Computational Thinking in pre-university Blended Learning classrooms. *Computers in Human Behavior*, 80, 412-419.
- Başol, G., Çakan, M., Kan, A., Özbek, Ö., Özdmir, D., & Yaşar, M. (2013). *Eğitimde ölçme ve değerlendirme*. Ankara: Pegem Akademi Yayınları.
- Bayman, P., & Mayer, R. E. (1983). A diagnosis of beginning programmers' misconceptions of BASIC programming statements. *Communications of the ACM*, 26(9), 677-679.
- Baz, F. Ç. (2018). Çocuklar için kodlama yazılımları üzerine karşılaştırmalı bir inceleme. *Current Research in Education*, 4(1), 36-47.
- Bell, T., Witten, I. H., Fellows, M., Adams, R., McKenzie, J., Powell, M., & Jarman, S. (2015). Cs unplugged. *Computer Science Without a Computer*. Creative Commons.
- Benz, C. R., Ridenour, C. S., & Newman, I. (2008). *Mixed methods research: Exploring the interactive continuum*. Carbondale: SIU Press.
- Benzer, A. İ., & Erümit, A. K. (2017). Programlama öğretimine yönelik lisansüstü tezlerin incelenmesi. *Journal of Instructional Technologies & Teacher Education Vol*, 6(3), 99-110.
- BK (2017). Dünyada kodlama eğitimi. <http://www.codingbk.com/dunyada-kodlama-egitimi.php> adresinden edinilmiştir.
- Boldbaatar, N. (2017). *Starlogo tng yazılımını kullanarak görsel programlama yardımıyla eğitsel oyun tasarlanması*. (Yüksek Lisans tezi). Yükseköğretim Kurulu Ulusal Tez Merkezi'nden edinilmiştir. (Tez No. 471987)
- Bogdanchikov, A., Zhaparov, M., & Suliyev, R. (2013). *Python to learn programming*. Paper presented at the Journal of Physics: Conference Series.
- Bosch, N., & D'Mello, S. (2017). The affective experience of novice computer programmers. *International journal of artificial intelligence in education*, 27(1), 181-206.
- Bosch, N., D'Mello, S., & Mills, C. (2013, June). *What emotions do novices experience during their first computer programming learning session?* Paper presented at the International Conference on Artificial Intelligence in Education.
- Bottoni, P., & Ceriani, M. (2015). *Linked Data Queries as Jigsaw Puzzles: a Visual Interface for SPARQL Based on Blockly Library*. Paper presented at the Proceedings of the 11th Biannual Conference on Italian SIGCHI Chapter.
- Bozionelos, N. (1996). Psychology of computer use: XXXIX. Prevalence of computer anxiety in British managers and professionals. *Psychological reports*, 78(3), 995-1002.
- Bradford, G. R. (2011). A relationship study of student satisfaction with learning online and cognitive load: Initial results. *The Internet and Higher Education*, 14(4), 217-226.
- Brennan, K., & Resnick, M. (2012, March). Using artifact-based interviews to study the development of computational thinking in interactive media design. *Paper presented at annual American Educational Research Association meeting*, Vancouver, BC, Canada.
- Brown, N. C., Altadmri, A., & Kölling, M. (2016, April). *Frame-based editing: Combining the best of blocks and text programming*. Paper presented at the 2016 International Conference on Learning and Teaching in Computing and Engineering (LaTICE).

- Bulut, E., & Akçacı, T. (2017). Endüstri 4.0 ve inovasyon göstergeleri kapsamında türkiye analizi. *ASSAM Uluslararası Hakemli Dergi*, 4(7), 55-77.
- Brown, N. C., & Wilson, G. (2018). Ten quick tips for teaching programming. *PLoS computational biology*, 14(4), e1006023.
- Burnett, M. M., & McIntyre, D. W. (1995). Visual programming. *Computer*(3), 14-16.
- Burns, J. (2012). School ICT to be replaced by computer science programme, BBC News. 02.12.2017 tarihinde <http://www.bbc.co.uk/news/education-16493929> adresinden ulaşılmıştır.
- Büyüköztürk, S. (2001). *Deneyisel desenler*. Pegema Yayıncılık. Ankara.
- Büyüköztürk, S. (2002). *Sosyal bilimler için veri analizi elkitabı*. Pegema Yayıncılık. Ankara.
- Büyüköztürk, Ş., Kılıç Çakmak, E., Akgün, Ö. E., Karadeniz, Ş., & Demirel, F. (2008). *Bilimsel araştırma yöntemleri*. Ankara: Pegem Yayınları.
- Büyüköztürk, Ş., Çakmak, E. K., Akgün, Ö. E., Karadeniz, Ş., & Demirel, F. (2017). *Bilimsel araştırma yöntemleri*. *Pegem Atıf İndeksi*, 1-360.
- Byrne, P., & Lyons, G. (2001). The effect of student attributes on success in programming. *Proceedings of The 6th Annual Conference On Innovation And Technology In Computer Science Education*, Canterbury, Kent, United Kingdom.
- Caiping, Q. Y., & Xiong. (2009). The Application research of cognitive load theory in the design of multimedia software [J]. *Journal of Distance Education*, 3.
- Çakmak, E. K. (2007). Çoklu ortamlarda dar boğaz: Aşırı bilişsel yüklenme. *Gazi Üniversitesi Gazi Eğitim Fakültesi Dergisi*, 27(2).
- Çakmur, H. (2012). Araştırmalarda ölçme-güvenilirlik-geçerlilik. *TAF Preventive Medicine Bulletin*, 11(3).
- Çatlak, Ş., Tekdal, M., & Baz, F. Ç. (2015). Scratch yazılımı ile programlama öğretiminin durumu: Bir doküman inceleme çalışması. *Journal of Instructional Technologies & Teacher Education*, 4(3).
- Cazan, A.-M., Cocoradă, E., & Maican, C. I. (2016). Computer anxiety and attitudes towards the computer and the internet with Romanian high-school and university students. *Computers in Human Behavior*, 55, 258-267.
- Chang, S. E. (2005). Computer anxiety and perception of task complexity in learning programming-related skills. *Computers in Human Behavior*, 21(5), 713-728.
- Chang, C.-K. (2014). Effects of using Alice and Scratch in an introductory programming course for corrective instruction. *Journal of Educational Computing Research*, 51(2), 185-204.
- Chang, T.-W., Hsu, J.-M., & Yu, P.-T. (2011). A comparison of single-and dual-screen environment in programming language: Cognitive loads and learning effects. *Journal of Educational Technology & Society*, 14(2), 188-200.
- Chao, P.-Y. (2016). Exploring students' computational practice, design and performance of problem-solving through a visual programming environment. *Computers & Education*, 95, 202-215.
- Cheng, G. (2019). Exploring factors influencing the acceptance of visual programming environment among boys and girls in primary schools. *Computers in Human Behavior*, 92, 361-372.

- Cheung, K. (1990). *Meaningful measurement in the classroom using the rasch model: some exemplars*: Retrieved from ERIC database. (ED 326544)
- Chua, S. L., Chen, D.-T., & Wong, A. F. (1999). Computer anxiety and its correlates: a meta-analysis. *Computers in Human Behavior*, 15(5), 609-623.
- Chuo, Y.-H., Tsai, C.-H., Lan, Y.-L., & Tsai, C.-S. (2011). The effect of organizational support, self efficacy, and computer anxiety on the usage intention of e-learning system in hospital. *African Journal of Business Management*, 5(14), 5518-5523.
- Çirakoğlu, O., C. (2004). Bilgisayar kaygısı. pivolka - *Başkent Üniversitesi İİBF, Eleştirel – Yaratıcı Düşünme ve Davranış Araştırmaları Laboratuvarı*,3(13), 15-18.
- Cierniak, G., Scheiter, K., & Gerjets, P. (2009). Explaining the split-attention effect: Is the reduction of extraneous cognitive load accompanied by an increase in germane cognitive load? *Computers in Human Behavior*, 25(2), 315-324.
- Clark, R. C., Nguyen, F., & Sweller, J. (2011). *Efficiency in learning: Evidence-based guidelines to manage cognitive load*: John Wiley & Sons.
- Cock, P. J., Antao, T., Chang, J. T., Chapman, B. A., Cox, C. J., Dalke, A., & Wilczynski, B. (2009). Biopython: freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics*, 25(11), 1422-1423.
- Code.org. (2018). <https://code.org/international/about> adresinden edinilmiştir.
- Çölkesen, R. (2002). *Bilgisayar programlama ve yazılım mühendisliğinde veri yapıları ve algoritmalar*: Papatya Yayıncılık.
- Cocorada, S. (2014). Computer anxiety, computer self-efficacy and demographic variables. *eLearning & Software for Education*(4).
- Connolly, C., Murphy, E., & Moore, S. (2007). *Second chance learners, supporting adults learning computer programming*. Paper presented at the international conference on engineering education–ICEE.
- Connolly, C., Murphy, E., & Moore, S. (2009). Programming anxiety amongst computing students—a key in the retention debate? *IEEE Transactions on Education*, 52(1), 52-56.
- Courte, J., Howard, E., & Bishop-Clark, C. (2006). Using Alice in a computer science survey course. *Information Systems Education Journal*, 4, 1-7.
- Creswell. (2003). *Research design: Qualitative, quantitative, and mixed methods approaches (2nd ed.)*. Thousand Oaks, CA: Sage.
- Creswell, J. W. (2007). *Qualitative Inquiry and Research Design: Choosing among five approaches (2nd ed.)*. Thousand Oaks, CA: Sage.
- Creswell, J. W., & Creswell, J. D. (2017). *Research design: Qualitative, quantitative, and mixed methods approaches*, CA:Sage publications.
- Csikszentmihalyi, M.(1990). *Flow: the psychology of optimal experience*. New York: Harper&Row.
- Culic, I., Radovici, A., & Vasilescu, L. M. (2015, September). *Auto-generating Google Blockly visual programming elements for peripheral hardware*. Paper presented at the RoEduNet International Conference-Networking in Education and Research (RoEduNet NER), 2015 14th.
- Çakiroğlu, Ü., Suiçmez, S. S., Kurtoğlu, Y. B., Sari, A., Yildiz, S., & Öztürk, M. (2018). Exploring perceived cognitive load in learning programming via Scratch. *Research in Learning Technology*, 26.

- Çakmak, E. K. (2007). Çoklu ortamlarda dar boğaz: Aşırı bilişsel yüklenme. *Gazi Üniversitesi Gazi Eğitim Fakültesi Dergisi*, 27(2).
- Çakmur, H. (2012). Araştırmalarda ölçme-güvenilirlik-geçerlilik. *TAF Preventive Medicine Bulletin*, 11(3).
- Çatlak, Ş., Tekdal, M., & Baz, F. Ç. (2015). Scratch yazılımı ile programlama öğretiminin durumu: Bir doküman inceleme çalışması. *Journal of Instructional Technologies & Teacher Education*, 4(3).
- Çetin, E. (2012). *Bilgisayar programlama eğitiminin çocukların problem çözme becerileri üzerine etkisi*. (Yüksek Lisans tezi). Yükseköğretim Kurulu Ulusal Tez Merkezi'nden edinilmiştir. (Tez No. 349116)
- Çölkesen, R. (2002). *Bilgisayar programlama ve yazılım mühendisliğinde veri yapıları ve algoritmalar*: Papatya Yayıncılık.
- Dalcin, L., Paz, R., Storti, M., & D'Elia, J. (2008). MPI for Python: Performance improvements and MPI-2 extensions. *Journal of Parallel and Distributed Computing*, 68(5), 655-662.
- Dasso, A., Funes, A., Riesco, D. E., Montejano, G. A., Peralta, M., & Salgado, C. (2005, April). *Teaching programming*. Paper presented at the I Jornadas de Educación en Informática y TICs en Argentina.
- Demir, F. (2015). *Programlama öğretiminde eğitsel programlama dilinin farklı kullanımlarının programlama başarısı ve kaygısına etkisi*. (Doktora tezi). Yükseköğretim Kurulu Ulusal Tez Merkezi'nden edinilmiştir. (Tez No. 429631)
- Demirer, V., & Sak, N. (2016). Programming education and new approaches around the world and in Turkey/Dünyada ve Türkiye'de programlama eğitimi ve yeni yaklaşımlar. *Eğitimde Kuram ve Uygulama*, 12(3), 521-546.
- Diñçer, A. (2018). *6.sınıf öğrencilerine scratch ve kodu game lab programlama dillerinin öğretiminde öğrencilerin tutum, öz yeterlilik ve akademik başarılarının karşılaştırılması*. (Yüksek Lisans tezi). Yükseköğretim Kurulu Ulusal Tez Merkezi'nden edinilmiştir. (Tez No. 512965)
- Doğan, U., & Kert, S. B. (2016). Bilgisayar oyunu geliştirme sürecinin, ortaokul öğrencilerinin eleştirel düşünme becerilerine ve algoritma başarılarına etkisi. *Boğaziçi Üniversitesi Eğitim Dergisi*, 33(2), 21-42.
- Dohn, N. B. (2019). Students' interest in Scratch coding in lower secondary mathematics. *British Journal of Educational Technology*.
- Doolittle, P. E., McNeill, A. L., Terry, K. P., & Scheer, S. B. (2005). Multimedia, cognitive load and pedagogy *Interactive multimedia in education and training* (pp. 184-212): IGI Global.
- Du Boulay, B. (1986). Some difficulties of learning to program. *Journal of Educational Computing Research*, 2(1), 57-73.
- Dyck, J. L., & Smither, J. A.-A. (1994). Age differences in computer anxiety: The role of computer experience, gender and education. *Journal of Educational Computing Research*, 10(3), 239-248.
- Dzhenzher, V. O. (2014). Computer simulation at school scratch and programming language choosing criteria. *Paper presented at the Global Engineering Education Conference (EDUCON)*, 2014 IEEE.

- Eguchi, A. (2014). Educational robotics for promoting 21st century skills. *Journal of Automation Mobile Robotics and Intelligent Systems*, 8(1), 5--11.
- Ekici, G., Abide, Ö. F., Canpolat, Y., & Öztürk, A. (2017). 21. yüzyıl becerilerine ait veri kaynaklarının analizi. *Eğitim ve Öğretim Araştırmaları Dergisi*, 6(1).
- Ekin, T. (2012). *Sönümlenme yöntemiyle oluşturulmuş web temelli öğretimin öğrencilerin bilişsel yüklenmesine, akademik başarısına ve transfer becerisine etkisi*. (Doktora tezi). Yükseköğretim Kurulu Ulusal Tez Merkezi'nden edinilmiştir. (Tez No. 317128)
- Esteves, M., & Mendes, A. J. (2004). A simulation tool to help learning of object oriented programming basics. *Paper presented at the Frontiers in Education, FIE 2004. 34th Annual*.
- Erdem, E. (2018). *Blok tabanlı ortamlarda programlama öğretimi sürecinde farklı öğretim stratejilerinin çeşitli değişkenler açısından incelenmesi* (Yüksek Lisans tezi). Yükseköğretim Kurulu Ulusal Tez Merkezi'nden edinilmiştir. (Tez No. 509354)
- Erkoç, M.F. (2018). *İşbirlikli oyun tasarımının eleştirel düşünme, problem çözme ve algoritma geliştirme becerisine etkisi*. (Doktora tezi). Yükseköğretim Kurulu Ulusal Tez Merkezi'nden edinilmiştir. (Tez No. 523420)
- Erol, O. (2015). *Scratch ile programlama öğretiminin bilişim teknolojileri öğretmen adaylarının motivasyon ve başarılarına etkisi*. (Doktora tezi). Yükseköğretim Kurulu Ulusal Tez Merkezi'nden edinilmiştir. (Tez No. 395186)
- Erol, O., & Kurt, A. A. (2017). The effects of teaching programming with scratch on pre-service information technology teachers' motivation and achievement. *Computers in Human Behavior*, 77(C), 11-18. doi: 10.1016/j.chb.2017.08.017
- Erol, O., & Şendağ, S. (2012). *İlköğretim ikinci kademedeki alice 3 boyutlu animasyon yazılımının bilişsel araç olarak kullanımına yönelik bir durum çalışması*. 6. Uluslararası Bilgisayar ve Öğretim Teknolojileri Sempozyumu'nda sunulan sözlü bildiri, Gaziantep.
- Ertuğrul, İ., & Deniz, G.(2018). 4.0 Dünyası: Pazarlama 4.0 ve Endüstri 4.0. *Bitlis Eren Üniversitesi Sosyal Bilimler Enstitüsü Dergisi*, 7(1), 143-170.
- Esteves, M., & Mendes, A. J. (2004, October). *A simulation tool to help learning of object oriented programming basics*. Paper presented at the Frontiers in Education, 2004. FIE 2004. 34th Annual.
- Feng, C. Y., & Chen, M. P. (2014). The effects of goal specificity and scaffolding on programming performance and self-regulation in game design. *British Journal of Educational Technology*, 45(2), 285-302.
- Ferrer-Mico, T., Prats-Fernández, M. À., & Redo-Sanchez, A. (2012). Impact of scratch programming on students' understanding of their own learning process. *Procedia-Social and Behavioral Sciences*, 46, 1219-1223.
- Fesakis, G., & Serafeim, K. (2009, June). *Influence of the familiarization with scratch on future teachers' opinions and attitudes about programming and ICT in education*. Paper presented at the Acm Sigcse Bulletin.
- Fessakis, G., Gouli, E., & Mavroudi, E. (2013). Problem solving by 5–6 years old kindergarten children in a computer programming environment: A case study. *Computers & Education*, 63, 87-97.
- Figueroa Jr, R. B., & Amoloza, E. M. (2015). Addressing programming anxiety among non-computer science distance learners: a upou case study. *International Journal*, 9(1), 56-67.



- Fırat, S. Ü., & Fırat, O. Z. (2017). Sanayi 4.0 devrimi üzerine karşılaştırmalı bir inceleme: kavramlar, küresel gelişmeler ve türkiye. *Toprak İşveren Dergisi*, (114), 10-23.
- Field, A. (2009). *Discovering statistics using SPSS*. (3<sup>rd</sup> Ed.). London: Sage.
- Fowler, A. (2012, October). *Enriching student learning programming through using Kodu*. Paper presented at the Proceedings of the 3rd Annual Conference of Computing and Information Technology, Education and Research in New Zealand (incorporating 24th Annual NACCQ).
- Fowler, A., Fristce, T., & MacLauren, M. (2012). Kodu Game Lab: A programming environment. *The Computer Games Journal*, 1(1), 17-28.
- Fraenkel, J., & Wallen, N. (2006). How to design and evaluate research in education. USA: McGrawHill: Inc.
- Fraenkel, J., Wallen, N., & Hyun, H. (2012). Internal validity. *How to Design and Evaluate Research in Education*. New York: McGraw-Hill, 166-183.
- Fraser, N. (2015, October). *Ten things we've learned from Blockly*. Paper presented at the Blocks and Beyond Workshop (Blocks and Beyond), 2015 IEEE.
- Freeman, S. F., Jaeger, B. K., & Brougham, J. C. (2004). Pair programming: more learning and less anxiety in a first programming course. *Proceedings ASEE Annual Conference*.
- Fujiwara, K., Fushida, K., Tamada, H., Igaki, H., & Yoshida, N. (2012, October). *Why Novice Programmers Fall into a Pitfall?: Coding Pattern Analysis in Programming Exercise*. Paper presented at the Empirical Software Engineering in Practice (IWESep), 2012 Fourth International Workshop on.
- Gelen, İ. (2017). P21-program ve öğretimde 21. yüzyıl beceri çerçeveleri (abd uygulamaları). *Journal of Interdisciplinary Educational Research*, 1(2), 15-29.
- Gerjets, P., Scheiter, K., Opfermann, M., Hesse, F. W., & Eysink, T. H. (2009). Learning with hypermedia: The influence of representational formats and different levels of learner control on performance and learning behavior. *Computers in Human Behavior*, 25(2), 360-370.
- Genç Z. & Karakuş S. (2011, Eylül). *Tasarımla öğrenme: Eğitsel bilgisayar oyunları tasarımında Scratch kullanımı*. 5. Uluslararası Bilgisayar ve Öğretim Teknolojileri Sempozyumu'nda sunulan sözlü bildiri, Elazığ.
- Gezgin, D. M., Özcan, S. N., Ergün, K., Özge, K., & Neriman, E. (2017). Bilgisayar programlama eğitiminde scratch programı kullanımına ilişkin lise öğrencilerinin görüşleri. *Uluslararası Bilimsel Araştırmalar Dergisi (IBAD)*, 182-188.
- Ginat, D. (2006, November). *On novices' local views of algorithmic characteristics*. Paper presented at the International Conference on Informatics in Secondary Schools-Evolution and Perspectives, Lithuania.
- Gomes, A., & Mendes, A. J. (2007, June). *Learning to program-difficulties and solutions*. Paper presented at the International Conference on Engineering Education-ICEE.
- Govender, I., & Grayson, D. J. (2008). Pre-service and in-service teachers' experiences of learning to program in an object-oriented language. *Computers & Education*, 51(2), 874-885.
- Gökoğlu, S. (2017). Programlama eğitiminde algoritma algısı: bir metafor analizi. *Cumhuriyet International Journal of Education*, 6(1).

- Grover, S., & Pea, R. (2013). Computational thinking in K–12: A review of the state of the field. *Educational researcher*, 42(1), 38-43.
- Gülbahar, Y., & Kalelioğlu, F. (2014). The effects of teaching programming via Scratch on problem solving skills: A discussion from learners' perspective. *Informatics in Education-An International Journal*, 13(1), 33-50.
- Güneş, A., & Karabak, D. (2013). Ortaokul birinci sınıf öğrencileri için yazılım geliştirme alanında müfredat önerisi. *Eğitim ve Öğretim Araştırmaları Dergisi*, 2(3), 1-7.
- Gürcan-Namlu, A., & Ceyhan, E. (2003). Bilgisayar kaygısı: öğretmen adayları üzerinde çok yönlü bir inceleme. *Educational Sciences: Theory & Practice*, 3(2).
- Hackbarth, G., Grover, V., & Mun, Y. Y. (2003). Computer playfulness and anxiety: positive and negative mediators of the system experience effect on perceived ease of use. *Information & management*, 40(3), 221-232.
- Harms, K. J. (2013, July). *Applying cognitive load theory to generate effective programming tutorials*. Paper presented at the 2013 IEEE Symposium on Visual Languages and Human Centric Computing, Australia.
- Hongwarittorn, N., & Krairit, D. (2010, April). *Effects of program visualization (jeliot3) on students' performance and attitudes towards java programming*. Paper presented at the The spring 8th International conference on Computing, Communication and Control Technologies, Italy.
- Horwitz, E. K., Horwitz, M. B., & Cope, J. (1986). Foreign language classroom anxiety. *The Modern language journal*, 70(2), 125-132.
- Hromkovič, J., Serafini, G., & Staub, J. (2017, November). *XLogoOnline: A single-page, browser-based programming environment for schools aiming at reducing cognitive load on pupils*. In International Conference on Informatics in Schools: Situation, Evolution, and Perspectives (pp. 219-231). Springer, Cham.
- Hsu, T.-C., Chang, S.-C., & Hung, Y.-T. (2018). How to learn and how to teach computational thinking: Suggestions based on a review of the literature. *Computers & Education*, 126, 296-310.
- Hwang, W.-Y., Wang, C.-Y., Hwang, G.-J., Huang, Y.-M., & Huang, S. (2008). A web-based programming learning environment to support cognitive development. *Interacting with Computers*, 20(6), 524-534.
- Idemudia, E. C., Dasuki, S. I., & Ogedebe, P. (2016). Factors that influence students' programming skills: a case study from a Nigerian university. *International Journal of Quantitative Research in Education*, 3(4), 277-291.
- Idrees, M., Aslam, F., Shahzad, K., & Sarwar, S. M. (2018). Towards a universal framework for visual programming languages. *Pak. J. Engg. Appl. Sci. Vol*, 55-65.
- Igbaria, M., & Parasuraman, S. (1989). A path analytic study of individual characteristics, computer anxiety and attitudes toward microcomputers. *Journal of Management*, 15(3), 373-388.
- Industry 4.0 : A View from the OECD. (2018). EnviroTREC. <http://www.envirotrec.ca/2018/04/17/industry-4-0-a-view-from-the-oecd/>
- International Society for Technology in Education. (2016). ISTE Standarts For Students. <https://www.iste.org/standards/for-students> adresinden edinilmiştir.
- Işık, E. (1996). *Nevrozlar*. Ankara: Kent Matbaası.

- Jancheski, M. (2017). *Improving teaching and learning computer programming in schools through educational software*. Paper presented at the Olympiads in Informatics, Korea.
- Jenkins, T. (2002, August). *On the difficulty of learning to program*. Paper presented at the Proceedings of the 3rd Annual Conference of the LTSN Centre for Information and Computer Sciences.
- Jiao, Q. G., & Onwuegbuzie, A. J. (2017). The impact of information technology on library anxiety: The role of computer attitudes. *Information technology and libraries*, 23(4), 138-144.
- Johnson, R. B., & Onwuegbuzie, A. J. (2004). Mixed methods research: A research paradigm whose time has come. *Educational researcher*, 33(7), 14-26.
- Kafai, Y. B., Fields, D. A., Roque, R., Burke, W. Q., & Monroy-Hernandez, A. (2012). Collaborative agency in youth online and offline creative production in Scratch. *Research and Practice in Technology Enhanced*, 7(2), 63-87.
- Kalelioğlu, F. (2015). A new way of teaching programming skills to K-12 students: Code.org. *Computers in Human Behavior*, 52, 200-210.
- Kalelioğlu, F., & Gülbahar, Y. (2014). The Effects of teaching programming via scratch on problem solving skills: a discussion from learners' perspective. *Informatics in Education*, 13(1).
- Kanemune, S., Nakatani, T., Mitarai, R., Fukui, S., & Kuno, Y. (2004). Dolittle-experiences in teaching programming at K12 schools. In *Creating, Connecting and Collaborating through Computing, 2004. Proceedings. Second International Conference on* (pp. 177-184). IEEE.
- Karasar, N. (2008). *Bilimsel araştırma yöntemi: kavramlar-ilkeler-teknikler*. Ankara: Nobel Yayın Dağıtım.
- Kasalak, İ. (2017). *Robotik kodlama etkinliklerinin ortaokul öğrencilerinin kodlamaya ilişkin öz-yeterlik algularına etkisi ve etkinliklere ilişkin öğrenci yaşantıları*. (Yüksek Lisans tezi). Yükseköğretim Kurulu Ulusal Tez Merkezi'nden edinilmiştir. (Tez No. 454911)
- Katai, Z., & Toth, L. (2010). Technologically and artistically enhanced multi-sensory computer-programming education. *Teaching and teacher education*, 26(2), 244-251.
- Kaya, C.B. (2018). *Programlama dili öğretiminde alice programının öğrencilerin akademik başarıları, problem çözme becerisi alguları, motivasyonları ve programlamaya hazır bulunuşluk düzeyleri üzerine etkisi*. (Doktora tezi). Yükseköğretim Kurulu Ulusal Tez Merkezi'nden edinilmiştir. (Tez No. 532721)
- Kayabaşı, E. (2016). *Öğretmen adaylarının Alice deneyimi: 3B ortamda programlama*. (Yüksek Lisans tezi). Yükseköğretim Kurulu Ulusal Tez Merkezi'nden edinilmiştir. (Tez No. 429742)
- Kelleher, C., & Pausch, R. (2005). Lowering the barriers to programming: A taxonomy of programming environments and languages for novice programmers. *ACM Computing Surveys (CSUR)*, 37(2), 83-137.
- Kert, S. B., & Uğraş, T. (2009, Mayıs). *Programlama eğitiminde sadelik ve Eğlence: Scratch Örneği*. Paper presented at the The First International Congress of Educational Research, Çanakkale, Turkey.
- Kesici, T., & Kocabaş, Z. (2007). *Bilgisayar 2 Ders Kitabı (2. Baskı)*. Ankara: Semih Ofset.

- Kılıç, E., & Karadeniz, Ş. (2004). Hiper ortamlarda öğrencilerin bilişsel yüklenme ve kaybolma düzeylerinin belirlenmesi. *Kuram ve Uygulamada Eğitim Yönetimi Dergisi*, 10(4), 562-579.
- Kirschner, P. A. (2002). Cognitive load theory: Implications of cognitive load theory on the design of learning. *Learning and instruction*, 12, 1-10.
- Kirwan, C., Costello, E., & Donlon, E. (2018). Computational thinking and online learning: A systematic literature review. *Proceedings of the 17<sup>th</sup> European Conference on e-Learning*, Athens, Greece. (pp. 650-657). ECEL.
- Ko, P. (2013, October). *A longitudinal study of the effects of a high school robotics and computational thinking class on academic achievement (WIP)*. Paper presented at the Frontiers in Education Conference, 2013 IEEE.
- Koorsse, M., Cilliers, C., & Calitz, A. (2015). Programming assistance tools to support the learning of IT programming in South African secondary schools. *Computers & Education*, 82, 162-178.
- Korbach, A., Brünken, R., & Park, B. (2018). Differentiating different types of cognitive load: A comparison of different measures. *Educational psychology review*, 30(2), 503-529.
- Koren, I., & Klamma, R. (2018). Enabling visual community learning analytics with Internet of Things devices. *Computers in Human Behavior*, 89, 385-394.
- Korkmaz, Ö. (2016). The Effect of scratch-based game activities on students' attitudes, self-efficacy and academic achievement. *Online Submission*, 8(1), 16-23.
- Kölling, M. (1999). The problem of teaching object-oriented programming, Part 1: Languages. *Journal of Object-oriented programming*, 11(8), 8-15.
- Kölling, M., Brown, N. C., & Altadmri, A. (2015, November). *Frame-based editing: Easing the transition from blocks to text-based programming*. Paper presented at the Proceedings of the Workshop in Primary and Secondary Computing Education.
- Kristof-Brown, A. L., Zimmerman, R. D., & Johnson, E. C. (2005). Consequences of individuals' fit at work: a meta-analysis of person–job, person–organization, person–group, and person–supervisor fit. *Personnel psychology*, 58(2), 281-342.
- Kurihara, A., Sasaki, A., Wakita, K., & Hosobe, H. (2015). A programming environment for visual block-based domain-specific languages. *Procedia Computer Science*, 62, 287-296.
- Lahtinen, E., Ala-Mutka, K., & Järvinen, H.-M. (2005). A study of the difficulties of novice programmers. *Acm Sigcse Bulletin*, 37(3), 14-18.
- Lai, A.-F., & Yang, S.-M. (2011, September). *The learning effect of visualized programming learning on 6<sup>th</sup> graders' problem solving and logical reasoning abilities*. Paper presented at the Electrical and Control Engineering (ICECE), 2011 International Conference on.
- Lau, W. W., & Yuen, A. H. (2011). Modelling programming performance: Beyond the influence of learner characteristics. *Computers & Education*, 57(1), 1202-1213.
- Leahy, W., & Sweller, J. (2004). Cognitive load and the imagination effect. *Applied Cognitive Psychology: The Official Journal of the Society for Applied Research in Memory and Cognition*, 18(7), 857-875.
- Lee, C., & Baba, M. S. (2005, October). Work in progress-development of a 3-tier architecture C++ STL ITS. In Frontiers in Education, 2005. FIE'05. *Proceedings 35<sup>th</sup> Annual Conference* (pp. T3E-11). IEEE.

- Leso, T., & Peck, K. L. (1992). Computer anxiety and different types of computer courses. *Journal of Educational Computing Research*, 8(4), 469-478.
- Li, X. (2016). Application of cognitive load theory in programming teaching. *Journal of Higher Education Theory and Practice*, 16(6), 57.
- Lister, R. (2011). Computing education research programming, syntax and cognitive load. *ACM Inroads*, 2(2).
- Liu, M., Michael Reed, W., & Phillips, P. D. (1992). Teacher education students and computers: Gender, major, prior computer experience, occurrence, and anxiety. *Journal of Research on Computing in Education*, 24(4), 457-467.
- Luh, C.-J. (1994). Modelling and simulation in an object-oriented environment. *Information and Software Technology*, 36(6), 343-352.
- Lye, S. Y., & Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12? *Computers in Human Behavior*, 41, 51-61.
- Malan, D. J., & Leitner, H. H. (2007, April). *Scratch for budding computer scientists*. Paper presented at the Acm Sigcse Bulletin.
- Mannila, L., Peltomäki, M., & Salakoski, T. (2006). What about a simple language? Analyzing the difficulties in learning to program. *Computer Science Education*, 16(3), 211-227.
- Martens, R. L., Valcke, M., & Portier, S. J. (1997). Interactive learning environments to support independent learning: The impact of discernability of embedded support devices. *Computers & Education*, 28(3), 185-197.
- Mason, R., Cooper, G., & Wilks, B. (2015). Using cognitive load theory to select an environment for teaching mobile apps development. *Proceedings of the 17th Australasian Computing Education Conference*, Sydney, Australia, Volume: 160. ( pp. 47-56). ISBN: 9781921770425
- Mathew, D. (2012). From fatigue to anxiety? Implications for educational design in a Web 2.0 world. *Interactive Technology and Smart Education*, 9(2), 112-120.
- Mayer, D. P. (2008). *Overcoming school anxiety: How to help your child deal with separation, tests, homework, bullies, math phobia, and other worries*: AMACOM, American Management Association.
- Mazaitis, D. (1993). The object-oriented paradigm in the undergraduate curriculum: a survey of implementations and issues. *Acm Sigcse Bulletin*, 25(3), 58-64.
- McMillan, J. H., & Schumacher, S. (2010). *Research in education: evidence-based inquiry, myeducationlab series*. New Jersey : Pearson.
- Meerbaum-Salant, O., Armoni, M., & Ben-Ari, M. (2011). *Habits of programming in scratch*. Paper presented at the Proceedings of the 16th annual joint conference on Innovation and technology in computer science education.
- Mladenović, M., Boljat, I., & Žanko, Ž. (2018). Comparing loops misconceptions in block-based and text-based programming languages at the K-12 level. *Education and Information Technologies*, 23(4), 1483-1500.
- Moons, J., & De Backer, C. (2013). The design and pilot evaluation of an interactive learning environment for introductory programming influenced by cognitive load theory and constructivism. *Computers & Education*, 60(1), 368-384.

- Moreno, J., & Robles, G. (2014). *Automatic detection of bad programming habits in scratch: A preliminary study*. Paper presented at the Frontiers in Education Conference (FIE), 2014 IEEE.
- Moreno, R., & Mayer, R. E. (1999). Cognitive principles of multimedia learning: The role of modality and contiguity. *Journal of educational psychology, 91*(2), 358.
- Moser, R. (1997). *A fantasy adventure game as a learning environment: why learning to program is so difficult and what can be done about it*. Paper presented at the Acm Sigcse Bulletin.
- Mullins, P., Whitfield, D., & Conlon, M. (2009). Using Alice 2.0 as a first language. *Journal of Computing Sciences in Colleges, 24*(3), 136-143.
- Necessary, J. R., & Parish, T. S. (1996). The relationships between computer usage and computer-related attitudes and behaviors. *Education, 116*(3), 384-387.
- Noone, M., & Mooney, A. (2018). Visual and textual programming languages: a systematic review of the literature. *Journal of Computers in Education, 5*(2), 149-174.
- OECD. (2017). *Connected minds: Technology and today's learners, educational research and innovation*. OECD Publishing. 12.03.2019 tarihinde [http://www.oecd-ilibrary.org/education/connected-minds\\_9789264111011-en](http://www.oecd-ilibrary.org/education/connected-minds_9789264111011-en) adresinden erişildi.
- Olelewe, C. J., & Agomuo, E. E. (2016). Effects of B-learning and F2F learning environments on students' achievement in QBASIC programming. *Computer & Education, 103*(C), 76-86. doi: 10.1016/j.compedu.2016.09.012
- Orru, G., & Longo, L. (2018, September). *The evolution of Cognitive Load Theory and the measurement of its intrinsic, extraneous and germane loads: a review*. Paper presented at the International Symposium on Human Mental Workload: Models and Applications.
- Owolabi, J., Olanipekun, P., & Iwerima, J. (2018). Mathematics ability and anxiety, computer and programming anxieties, age and gender as determinants of achievement in Basic programming. *GSTF Journal on Computing (JoC), 3*(4).
- Owolabi, J. O., Adedayo, O., & Olayanju, T. (2018). Effects of solo and pair programming instructional strategies on students' academic achievement in visual-basic. net computer programming language. *GSTF Journal on Computing (JoC), 3*(3).
- Ozoran, D., Cagiltay, N., & Topalli, D. (2012, September). *Using scratch in introduction to programming course for engineering students*. Paper presented at the 2nd International Engineering Education Conference, Ankara. (IEEC2012).
- Özçınar, H., Yecan, E., & Tanyeri, T. (2016, Eylül). Öğretmen Gözüyle Görsel Programlama Öğretimi. Uluslararası Eğitimde Yeni Eğilimler Konferansı içinde (s.71-79). İzmir: Ege Üniversitesi.
- Öymen, E. E. (2014, 15 Eylül). Bilişim dili BBC'de program. 01.12.2017 tarihinde <http://www.bthaber.com/yazarlar/bilisim-dili-bbc%E2%80%99de-program/1/13512> adresinden ulaşılmıştır.
- Özçakmak, Ş. (2014). Bilgisayar kullanımı çocukta bağımlılık yapar mı? 01.12.2017 tarihinde <http://www.haberturk.com/polemik/haber/973204-bilgisayar-kullanimi-cocukta-bagimlilik-yapar-mi?> adresinden ulaşılmıştır.
- Özmen, B., & Altun, A. (2014). Undergraduate students' experiences in programming: Difficulties and obstacles. *Turkish Online Journal of Qualitative Inquiry, 5*(3), 1-27.

- Paas, F., Renkl, A., & Sweller, J. (2003). Cognitive load theory and instructional design: Recent developments. *Educational psychologist*, 38(1), 1-4.
- Paas, F., Tuovinen, J. E., Tabbers, H., & Van Gerven, P. W. (2003). Cognitive load measurement as a means to advance cognitive load theory. *Educational psychologist*, 38(1), 63-71.
- Paas, F. G., & Van Merriënboer, J. J. (1994). Variability of worked examples and transfer of geometrical problem-solving skills: A cognitive-load approach. *Journal of educational psychology*, 86(1), 122.
- Papavlasopoulou, S., Giannakos, M. N., & Jaccheri, L. (in press). Exploring children's learning experience in constructionism-based coding activities through design-based research. *Computers in Human Behavior*.
- Park, C. J., Hyun, J. S., Jin, H., & Jeong, H. (2015). Relation analysis of high school students' abstract thinking, language familiarity, and academic achievement on c and scratch program understanding. *The Journal of Korean Association of Computer Education*, 18(6), 1-12.
- Pasternak, E., Fenichel, R., & Marshall, A. N. (2017, October). *Tips for creating a block language with blockly*. Paper presented at the Blocks and Beyond Workshop (B&B), 2017 IEEE.
- Pea, R. D., & Kurland, D. M. (1984). On the cognitive effects of learning computer programming. *New ideas in psychology*, 2(2), 137-168.
- Pérez-Marín, D., Hijón-Neira, R., Bacelo, A., & Pizarro, C. (in press). Can computational thinking be improved by using a methodology based on metaphors and Scratch to teach computer programming to children? *Computers in Human Behavior*.
- Perković, L., Settle, A., Hwang, S., & Jones, J. (2010). *A framework for computational thinking across the curriculum*. Paper presented at the Proceedings of the fifteenth annual conference on Innovation and technology in computer science education.
- Prisacari, A. A., & Danielson, J. (2017). Computer-based versus paper-based testing: Investigating testing mode with cognitive load and scratch paper use. *Computers in Human Behavior*, 77, 1-10.
- Qian, M., & Clark, K. R. (2016). Game-based Learning and 21st century skills: A review of recent research. *Computers in Human Behavior*, 63, 50-58.
- Renumol, V., Jayaprakash, S., & Janakiram, D. (2009). Classification of cognitive difficulties of students to learn computer programming. *Indian Institute of Technology, India*, 12.
- Robertson, J., & Howells, C. (2008). Computer game design: Opportunities for successful learning. *Computer & Education*, 50(2), 559-578.
- Rodger, S. H., Bashford, M., Dyck, L., Hayes, J., Liang, L., Nelson, D., & Qin, H. (2010). *Enhancing K-12 education with alice programming adventures*. Paper presented at the Proceedings of the fifteenth annual conference on Innovation and technology in computer science education.
- Rogerson, C., & Scott, E. (2010). The fear factor: How it affects students learning to program in a tertiary environment. *Journal of Information Technology Education: Research*, 9, 147-171.
- Ropp, M. M. (1999). Exploring individual characteristics associated with learning to use computers in preservice teacher preparation. *Journal of Research on Computing in Education*, 31(4), 402-424.

- Rosen, L. D., & Weil, M. M. (1995). Computer anxiety: A cross-cultural comparison of university students in ten countries. *Computers in Human Behavior*, 11(1), 45-64.
- Russell, G., & Bradley, G. (1997). Teachers' computer anxiety: Implications for professional development. *Education and Information Technologies*, 2(1), 17-30.
- Saeli, M., Perrenet, J., Jochems, W. M., & Zwaneveld, B. (2011). Teaching programming in Secondary school: A pedagogical content knowledge perspective. *Informatics in Education*, 10(1), 73-88.
- Sáez-López, J.-M., Román-González, M., & Vázquez-Cano, E. (2016). Visual programming languages integrated across the curriculum in elementary school: A two year case study using "Scratch" in five schools. *Computers & Education*, 97, 129-141.
- Salleh, S. M., Shukur, Z., & Judi, H. M. (2018). Scaffolding model for efficient programming learning based on cognitive load theory. *International Journal of Pure and Applied Mathematics*, 118(7 Special Issue), 77-82.
- Sam, H. K., Othman, A. E. A., & Nordin, Z. S. (2005). Computer self-efficacy, computer anxiety, and attitudes toward the Internet: A study among undergraduates in UNIMAS. *Journal of Educational Technology & Society*, 8(4).
- Sanders, I. D., & Langford, S. (2008, December). *Students' perceptions of python as a first programming language at wits*. Paper presented at the Acm Sigcse Bulletin.
- Sanner, M. F. (1999). Python: A programming language for software integration and development. *J Mol Graph Model*, 17(1), 57-61.
- Saygıner, Ş. (2017). *Blok tabanlı görsel ve metin tabanlı programlama öğretimlerinin erişimi, mantıksal düşünme ve motivasyona etkileri*. (Yüksek Lisans tezi). Yükseköğretim Kurulu Ulusal Tez Merkezi'nden edinilmiştir. (Tez No. 454912)
- Saygıner, Ş., & Tüzün, H. (2017, Şubat). İlköğretim Düzeyinde Programlama Eğitimi: Yurt Dışı Ve Yurt İçi Perspektifinden Bir Bakış. *Akademik Bilişim Konferansı*.
- Sayın, Z., & Seferoğlu, S. S. (2016, Şubat). Yeni bir 21. yüzyıl becerisi olarak kodlama eğitimi ve kodlamanın eğitim politikalarına etkisi. *Akademik Bilişim Konferansı*, 3-5.
- Sener, S., & Elevation, B. (2017). Endüstri 4.0'da yeni iş kolları ve yüksek öğrenim. *MÜHENDİS BEYİNLER DERGİSİ*.
- Shaffer, C. A., Cooper, M. L., Alon, A. J. D., Akbar, M., Stewart, M., Ponce, S., & Edwards, S. H. (2010). Algorithm visualization: The state of the field. *ACM Transactions on Computing Education (TOCE)*, 10(3), 9.
- Shin, S., Park, P., & Bae, Y. (2013). The effects of an information-technology gifted program on friendship using scratch programming language and clutter. *International Journal of Computer and Communication Engineering*, 2(3), 246.
- Shin, S., Park, P. (2014). A Study on the effect affecting problem solving ability of primary students through the scratch programming. *Advanced Science and Technology Letters*, 59, 117-120.
- Simsek, A. (2011). The relationship between computer anxiety and computer self-efficacy. *Online Submission*, 2(3), 177-187.
- Sivasakthi, M., & Rajendran, R. (2011). Learning difficulties of object-oriented programming paradigm using Java: students' perspective. *Indian Journal of Science and Technology*, 4(8), 983-985.
- Solmaz, E. (2014). *Programlama dili öğretiminde Alice yazılımının ders başarısı, eleştirel düşünme ve problem çözme becerileri ile üstbilişsel farkındalık düzeyine etkisi*.



- (Doktora tezi). Yükseköğretim Kurulu Ulusal Tez Merkezi'nden edinilmiştir. (Tez No. 381475)
- Sözbilir, M. (2010). Madde analizi ve test geliştirme. *Content Analysis and Test Development*, 16, 2013.
- Stachel, J., Marghitu, D., Brahim, T. B., Sims, R., Reynolds, L., & Czelusniak, V. (2013). Managing cognitive load in introductory programming courses: A cognitive aware scaffolding tool. *Journal of Integrated Design and Process Science*, 17(1), 37-54.
- Stenerson, M. E. (2012). Academic game development: practices and design strategies for creating STEM games. *Graduate Theses and Dissertations*.
- Su, A. Y., Huang, C. S., Yang, S. J., Ding, T., & Hsieh, Y. (2015). Effects of annotations and homework on learning achievement: An empirical study of Scratch programming pedagogy. *Journal of Educational Technology & Society*, 18(4), 331.
- Sullivan, A., & Bers, M. U. (2013). Gender differences in kindergarteners' robotics and programming achievement. *International journal of technology and design education*, 23(3), 691-702.
- Sweller, J. (1994). Cognitive load theory, learning difficulty, and instructional design. *Learning and instruction*, 4(4), 295-312.
- Sweller, J. (2010). Element interactivity and intrinsic, extraneous, and germane cognitive load. *Educational psychology review*, 22(2), 123-138.
- Sweller, J. (2011). Cognitive load theory *psychology of learning and motivation* (Vol. 55, pp. 37-76): Elsevier.
- Sweller, J., Ayres, P., & Kalyuga, S. (2011). Measuring cognitive load. *Cognitive load theory* (pp. 71-85): Springer.
- Sweller, J., Van Merriënboer, J. J., & Paas, F. G. (1998). Cognitive architecture and instructional design. *Educational psychology review*, 10(3), 251-296.
- Sykes, E. R. (2007). Determining the effectiveness of the 3D Alice programming environment at the computer science I level. *Journal of Educational Computing Research*, 36(2), 223-244.
- Şencan, H. (2005). *Güvenilirlik ve geçerlilik*: Ankara:Seçkin Yayıncılık.
- Şuman, N. (2017). Akıllı üretim çağı: Endüstri 4.0.<http://www.fortuneturkey.com/akilli-uretim-cagi-endustri-40-42841> [14 Aralık 2017].
- Tanriogen, Z. M. (2018). African university students' intercultural experiences with impeding factors: case from northern cyprus. *Eurasian Journal of Educational Research*, 77, 163-184.
- Tepgeç, M., & Çevik, Y. D. (2018). Comparison of three instructional strategies in teaching programming: restudying material, testing and worked example. *Journal of Learning and Teaching in Digital Age (JOLTIDA)*, 3(2), 42-50.
- Topalli, D., & Cagiltay, N. E. (2018). Improving programming skills in engineering education through problem-based game projects with Scratch. *Computers & Education*, 120, 64-74.
- Topu, F. B., Baydaş, Ö., Turan, Z., & Göktaş, Y. (2013). Common reliability and validity strategies in instructional technology research. *Çukurova Üniversitesi Eğitim Fakültesi Dergisi*, 42(1), 110-126.

- Trilling, B., & Fadel, C. (2009). *21st century skills: Learning for life in our times*. San Francisco: John Wiley & Sons.
- Trilling, B., & Fadel, C. (2010). 21st century skills: Learning for life in our times. *Teacher Librarian*, 37(4), 74.
- Tsai, C.-Y. (2019). Improving students' understanding of basic programming concepts through visual programming language: The role of self-efficacy. *Computers in Human Behavior*.
- TTKB (2012). Bilişim teknolojileri ve yazılım dersi öğretim programı. <http://ttkb.meb.gov.tr/program2.aspx/program2.aspx?islem=1&kno=196> adresinden 30.11.2017 tarihinde alınmıştır.
- TTKB (2017). Bilgisayar bilimi dersi öğretim programı. <http://ttkb.meb.gov.tr/program2.aspx/program2.aspx?islem=1&kno=196> adresinden 30.11.2017 tarihinde alınmıştır.
- TTKB (2018). Bilgisayar bilimi dersi (kur 1-2) öğretim programı. <http://mufredat.meb.gov.tr/ProgramDetay.aspx?PID=335> adresinden edinilmiştir.
- Tuckman, B. W. (2003). The effect of learning and motivation strategies training on college students' achievement. *Journal of College Student Development*, 44(3), 430-437.
- Tuomi, P., Multisilta, J., Saarikoski, P., & Suominen, J. (2018). Coding skills as a success factor for a society. *Education and Information Technologies*, 23(1), 419-434.
- Uluyol, Ç., & Eryılmaz, S. (2015). 21. Yüzyıl becerileri ışığında fatih projesi değerlendirmesi. *Gazi Üniversitesi Gazi Eğitim Fakültesi Dergisi*, 35(2), 209-229.
- Uzgun, B. Ç. ve Aykaç, N. (2016). Bilişim teknolojileri ve yazılım dersi öğretim programının öğretmen görüşlerine göre değerlendirilmesi (Ege bölgesi örneği). *Mustafa Kemal Üniversitesi Sosyal Bilimler Enstitüsü Dergisi*, 13(34), 273-297.
- Ünal & Topu. (2018, Nisan). Blok tabanlı programlama öğretimi: 2012-2017 araştırma eğilimleri. 27. Uluslararası Eğitim Bilimleri Kongresi'nde sunulan sözlü bildiri, Antalya.
- Van-Roy, P., & Haridi, S. (2004). *Concepts, techniques, and models of computer programming*: MIT press.
- Vaidyanathan, S. (2013). Opinion: We need coding in schools, but where are the teachers? 01.12.2017 tarihinde <https://www.edsurge.com/n/2013-12-09-opinion-we-need-coding-in-schools-but-where-are-the-teachers> adresinden ulaşılmıştır.
- Van Rossum, G. (1999). Computer programming for everybody. *Proposal to the Corporation for National Research Initiatives*.
- Vatansever, Ö. (2018). *Scratch İle Programlama Öğretiminin Ortaokul 5. ve 6. Sınıf Öğrencilerinin Problem Çözme Becerileri Üzerindeki Etkisinin İncelenmesi*. (Yüksek Lisans tezi). Yükseköğretim Kurulu Ulusal Tez Merkezi'nden edinilmiştir. (Tez No. 501053)
- Wang, T. C., Mei, W. H., Lin, S. L., Chiu, S. K., & Lin, J. C. (2009, October). Teaching programming concepts to high school students with alice. *Proceedings of the 39th ASEE/IEEE Frontiers in Education Conference* (pp. 1-6). San Antonio, TX: IEEE.
- Wang, H. Y., Huang, I., & Hwang, G. J. (2014, July). *Effects of an integrated Scratch and project-based learning approach on the learning achievements of gifted students in computer courses*. Paper presented at the 2014 IIAI 3rd International Conference on Advanced Applied Informatics.

- Van Rossum, G., & Drake, F. L. (1995). Extending and embedding the Python interpreter. Centrum voor Wiskunde en Informatica.
- Webber, C. G., & Possamai, R. (2009, July). *An immune-based approach to evaluate programming learning*. Paper presented at the 9th IFIP World Conference on Computers in Education, Bento Gonçalves, RS, Brasil.
- Weinberg, A. E. (2013). Computational thinking: An investigation of the existing scholarship and research. *Colorado State University*.
- Weintrop, D., & Wilensky, U. (2015, June). *To block or not to block, that is the question: students' perceptions of blocks-based programming*. Paper presented at the Proceedings of the 14th International Conference on Interaction Design and Children.
- Weintrop, D., & Wilensky, U. (2015b, March). *Using commutative assessments to compare conceptual understanding in blocks-based and text-based programs*. Paper presented at the ICER.
- Weintrop, D., & Wilensky, U. (2018). How block-based, text-based, and hybrid block/text modalities shape novice programming practices. *International Journal of Child-Computer Interaction*, 17, 83-92.
- Woszczyński, A. B., Guthrie, T. C., Chen, T. L., & Shade, S. (2004). Personality as a predictor of student success in programming principles. I. *Proceedings of the Southern Association for Information Systems*, 1-6.
- Wu, W.-Y., Chang, C.-K., & He, Y.-Y. (2010, March). *Using Scratch as game-based learning tool to reduce learning anxiety in programming course*. Paper presented at the Global Learn.
- Yecan, E., Özçınar, H., & Tanyeri, T. (2017). Bilişim teknolojileri öğretmenlerinin görsel programlama öğretimi deneyimleri. *İlköğretim Online*, 16(1).
- Yıldırım, E. (2017). *Scratch programlama dili eğitimine yönelik bir mobil uygulamanın geliştirilmesi*. (Yüksek Lisans tezi). Yükseköğretim Kurulu Ulusal Tez Merkezi'nden edinilmiştir. (Tez No. 471031)
- Yıldırım, A., & Şimşek, H. (2006). *Sosyal bilimlerde nitel araştırma yöntemleri*. Ankara: Seçkin Yayıncılık.
- Yıldırım, K. (2010). Nitel araştırmalarda niteliği artırma. *İlköğretim Online*, 9(1).
- Yıldız, A. (2018). Endüstri 4.0 ve akıllı fabrikalar. *Sakarya University Journal of Science*, 22(2), 548-558.
- Yılmaz, M. (2012). *C# programlama dersinde çoklu ortam tasarım ilkelerine göre hazırlanmış materyallerin moodle öğrenme yönetim sistemi üzerinde kullanılmasının yüksek öğrenim öğrencilerinin bilişsel yüklerine ve ders başarılarına etkisi*. (Yüksek Lisans tezi). Yükseköğretim Kurulu Ulusal Tez Merkezi'nden edinilmiştir. (Tez No. 344965)
- Yiğit, M.F. (2016). *Görsel programlama ortamı ile öğretimin öğrencilerin bilgisayar programlamayı öğrenmesine ve programlamaya karşı tutumlarına etkisinin incelenmesi*. (Yüksek Lisans tezi). Yükseköğretim Kurulu Ulusal Tez Merkezi'nden edinilmiştir. (Tez No. 442990)
- Yoo, S., Kim, K. A., Kim, Y., Yeum, Y., Kanemune, S., & Lee, W. (2006). Empirical study of educational programming language for k12: *Between dolittle and visual basic*. *IJCSNS*, 6(6), 119.

- Yousoof, M., Sapiyan, M., & Kamaluddin, K. (2006). Reducing cognitive load in learning computer programming. *Proceedings of the World Academy of Science, Engineering and Technology*, 12, 259-262.
- Yukselturk, E., & Altıok, S. (2015). Bilişim teknolojileri öğretmen adaylarının bilgisayar programlama öğretimine yönelik görüşleri. *Amasya Üniversitesi Eğitim Fakültesi Dergisi*, 4(1), 50-65.
- Yukselturk, E., & Altıok, S. (2017). An investigation of the effects of programming with Scratch on the preservice IT teachers' self-efficacy perceptions and attitudes towards computer programming. *British Journal of Educational Technology*, 48(3), 789-801.
- Yurdugül, H., & Aşkar, P. (2013). Learning programming, problem solving and gender: A longitudinal study. *Procedia-Social and Behavioral Sciences*, 83, 605-610.
- Yüksel, S., & Gündoğdu, K. (2018). Scratch öğretiminde ayrılıp birleşme tekniği kullanımının derse yönelik tutuma akademik başarıya ve kalıcılığa etkisi. *Ege Eğitim Dergisi*, 19(1), 245-261.
- Yükseltürk, E., & Altıok, S. (2015). Bilişim teknolojileri öğretmen adaylarının bilgisayar programlama öğretimine yönelik görüşleri. *Amasya Üniversitesi Eğitim Fakültesi Dergisi*, 4(1), 50-65.
- Zainal, N. F. A., Shahrani, S., Yatim, N. F. M., Rahman, R. A., Rahmat, M., & Latih, R. (2012). Students' perception and motivation towards programming. *Procedia-Social and Behavioral Sciences*, 59, 277-286.
- Zamin, N., Ab Rahim, H., Savita, K., Bhattacharyya, E., Zaffar, M., & Jamil, S. N. K. M. (2018, August). *Learning Block Programming using Scratch among School Children in Malaysia and Australia: An Exploratory Study*. Paper presented at the 2018 4th International Conference on Computer and Information Sciences, Dubai. (ICCOINS).

## EKLER

### Ek-1. Bilgisayar Programlama Kaygı Ölçeği

#### PROGRAMLAMA KAYGI ÖLÇEĞİ

Sevgili öğrenciler,

Bu ölçek programlama öğretim sürecinde yaşadığınız deneyimi belirlemek amacıyla hazırlanmıştır. Her soruyu dikkatli bir şekilde okuyup cevaplamanızı ve hiçbir soruyu cevapsız bırakmamanızı rica ederiz. Vermiş olduğunuz cevaplar sadece araştırma amaçlı kullanılacaktır. Katılımınız için teşekkür ederiz.

Dr. Öğr. Üyesi Fatma Burcu TOPU  
Bilgisayar Bilimi Öğretmeni Alper ÜNAL

- Okul Numaranız:.....
- Sınıf Şubeniz: Hazırlık - A  Hazırlık - B  Hazırlık - C  Hazırlık - D
- Cinsiyetiniz:  Kız  Erkek

Lütfen, programlama ile ilgili kendinizi nasıl hissettiğinizi en iyi açıklayan numarayı yuvarlak içine alarak tüm ifadeleri işaretleyiniz.

Hiçbir zaman doğru değil	0	Nadiren doğru	1	Bazen doğru	2	Sıkla doğru	3	Her zaman doğru	4
--------------------------	---	---------------	---	-------------	---	-------------	---	-----------------	---

1.	Bilgisayarlar kendimi tedirgin ve rahatsız hissetmeme sebep oluyor.	0	1	2	3	4
2.	Programlarda hata ayıklama benim için büyük bir endişe kaynağıdır.	0	1	2	3	4
3.	Program yazmayı başaramayarak kendimi arkadaşlarımdan önünde komik duruma düşürmekten tedirgin oluyorum.	0	1	2	3	4
4.	Birçok bilgisayar terimi yüzünden aklım karışıyor.	0	1	2	3	4
5.	Programlamayı anlayabileceğimi düşünmüyorum.	0	1	2	3	4
6.	Yazdığım programlarda birçok hata çıkıyor.	0	1	2	3	4
7.	Programımda hata yaptığım anda öğretmenim tarafından seçilmekten endişe duyuyorum.	0	1	2	3	4
8.	Eğer programım çalışmazsa, sonraki aşamada ne yapacağımı bilemiyorum.	0	1	2	3	4
9.	Programım çalışmadığı zaman endişeleniyorum.	0	1	2	3	4
10.	Öğretmenimin benim iyi program yazmadığımı biliyor olması beni endişelendiriyor.	0	1	2	3	4
11.	Programımın satırlarında hata olduğunda düzgün düşünemiyorum.	0	1	2	3	4
12.	Programımdaki hataları tekrar tekrar düzeltmek zorunda kaldığımda kendimi gergin hissediyorum.	0	1	2	3	4
13.	Program satırları daha karmaşık olmaya başladığında yoğunlaşamadığımı hissediyorum.	0	1	2	3	4
14.	Daha yetenekli bilgisayar programlama öğrencilerinin varlığı beni geriyor.	0	1	2	3	4
15.	Arkadaşlarım programlama yapabildiği fakat benim yapamadığım zamanlarda kendimi üzgün hissediyorum.	0	1	2	3	4
16.	Bir bilgisayar sorununu çözemediğimde zihnim boşalmış gibi oluyor.	0	1	2	3	4
17.	Bilgisayar programımda hataların olduğu her zaman kendimi gergin hissediyorum.	0	1	2	3	4
18.	Arkadaşlarım benim anlamadığım programlama konularında konuştuğunda sıkıntı yaşıyorum.	0	1	2	3	4
19.	Bilgisayarla çalışma konusunda her zaman kendimi güvensiz hissediyorum.	0	1	2	3	4

## Ek-2. Bilişsel Yük Ölçeği (Haftalık)

### BİLİŞSEL YÜK ÖLÇEĞİ

Sevgili öğrenciler,

Bu ölçek programlama öğretim sürecinde yaşadığınız zihinsel süreçleri ve yüklenmeleri belirlemek amacıyla hazırlanmıştır. Her soruyu dikkatli bir şekilde okuyup cevaplamanızı ve hiçbir soruyu cevapsız bırakmamanızı rica ederiz. Vermiş olduğunuz cevaplar sadece araştırma amaçlı kullanılacaktır. Katılımınız için teşekkür ederiz.

Dr. Öğr. Üyesi Fatma Burcu TOPU  
Bilgisayar Bilimi Öğretmeni Alper ÜNAL

4. Okul Numaranız:.....

5. Sınıf Şubeniz: Hazırlık - A  Hazırlık - B  Hazırlık - C  Hazırlık - D

6. Cinsiyetiniz:  Kız  Erkek

**Hafta-1:** Belirlenen programlama dilinde değişkenleri ve sabitleri uygun şekilde kullanarak programlar geliştirir.

Hafta-1 Konu Başlıkları	
• Sabit Değerler	
• Karışık Türü İfadeler	
• Aritmetik Örnekler	
• Aritmetik İfadeler	

Aşağıdaki her bir konu sizin için şu anda ne kadar zor gelmektedir?

	Çok çok az	Çok az	Az	Kısmen az	Ne az ne de fazla	Kısmen fazla	Fazla	Çok fazla	Çok çok fazla
Konu	1	2	3	4	5	6	7	8	9
Sabit Değerler									
Karışık Türü İfadeler									
Aritmetik Örnekler									
Aritmetik İfadeler									

Bu dersin öğrenme ortamında önemli ve önemsiz bilgileri ayırt etmek ne kadar zordur?

	Çok çok az	Çok az	Az	Kısmen az	Ne az ne de fazla	Kısmen fazla	Fazla	Çok fazla	Çok çok fazla
Konu	1	2	3	4	5	6	7	8	9
Sabit Değerler									
Karışık Türü İfadeler									
Aritmetik Örnekler									
Aritmetik İfadeler									

Dersin öğrenme ortamı sizin için ne kadar zordur?

	Çok çok az	Çok az	Az	Kısmen az	Ne az ne de fazla	Kısmen fazla	Fazla	Çok fazla	Çok çok fazla
Konu	1	2	3	4	5	6	7	8	9
Sabit Değerler									
Karışık Türü İfadeler									
Aritmetik Örnekler									
Aritmetik İfadeler									

Derste yapılan en son örneği takip etmek için sarf ettiğiniz çabayı aşağıda gösterilen ölçek üzerinde belirtiniz.

	Çok çok az	Çok az	Az	Kısmen az	Ne az ne de fazla	Kısmen fazla	Fazla	Çok fazla	Çok çok fazla
Konu	1	2	3	4	5	6	7	8	9
Sabit Değerler									
Karışık Türü İfadeler									
Aritmetik Örnekler									
Aritmetik İfadeler									

Öğrenme ortamında ihtiyacınız olan tüm bilgileri toplamak sizin için ne kadar zordur?

	Çok çok az	Çok az	Az	Kısmen az	Ne az ne de fazla	Kısmen fazla	Fazla	Çok fazla	Çok çok fazla
Konu	1	2	3	4	5	6	7	8	9
Sabit Değerler									
Karışık Türü İfadeler									
Aritmetik Örnekler									
Aritmetik İfadeler									

**Hafta-2:** Belirlenen programlama dilinde deęişkenleri ve sabitleri uygun şekilde kullanarak programlar geliştirir.

Hafta-2 Konu Başlıkları	
• Sabit Deęerler	
• Karışık Türü İfadeler	
• Aritmetik Örnekler	
• Aritmetik İfadeler	

Aşğıdaki her bir konu sizin için şu anda ne kadar zor gelmektedir?

	Çok çok az	Çok az	Az	Kısmen az	Ne az ne de fazla	Kısmen fazla	Fazla	Çok fazla	Çok çok fazla
Konu	1	2	3	4	5	6	7	8	9
Sabit Deęerler									
Karışık Türü İfadeler									
Aritmetik Örnekler									
Aritmetik İfadeler									

Bu dersin öğrenme ortamında önemli ve önemsiz bilgileri ayırt etmek ne kadar zordur?

	Çok çok az	Çok az	Az	Kısmen az	Ne az ne de fazla	Kısmen fazla	Fazla	Çok fazla	Çok çok fazla
Konu	1	2	3	4	5	6	7	8	9
Sabit Deęerler									
Karışık Türü İfadeler									
Aritmetik Örnekler									
Aritmetik İfadeler									

Dersin öğrenme ortamı sizin için ne kadar zordur?

	Çok çok az	Çok az	Az	Kısmen az	Ne az ne de fazla	Kısmen fazla	Fazla	Çok fazla	Çok çok fazla
Konu	1	2	3	4	5	6	7	8	9
Sabit Deęerler									
Karışık Türü İfadeler									
Aritmetik Örnekler									
Aritmetik İfadeler									



Derste yapılan en son örneği takip etmek için sarf ettiğiniz çabayı aşağıda gösterilen ölçek üzerinde belirtiniz.

	Çok çok az	Çok az	Az	Kısmen az	Ne az ne de fazla	Kısmen fazla	Fazla	Çok fazla	Çok çok fazla
Konu	1	2	3	4	5	6	7	8	9
Sabit Değerler									
Karışık Türü İfadeler									
Aritmetik Örnekler									
Aritmetik İfadeler									

Öğrenme ortamında ihtiyacınız olan tüm bilgileri toplamak sizin için ne kadar zordur?

	Çok çok az	Çok az	Az	Kısmen az	Ne az ne de fazla	Kısmen fazla	Fazla	Çok fazla	Çok çok fazla
Konu	1	2	3	4	5	6	7	8	9
Sabit Değerler									
Karışık Türü İfadeler									
Aritmetik Örnekler									
Aritmetik İfadeler									

**Hafta-3:** Belirlenen programlama dilinde kontrol yapılarını kullanarak programlar geliştirir.

Hafta-3 Konu Başlıkları
• İlişkisel Operatörler
• İf ifadesi
• İf / Else ifadesi
• İç içe koşul ifadeleri

Aşağıdaki her bir konu sizin için şu anda ne kadar zor gelmektedir?

	Çok çok az	Çok az	Az	Kısmen az	Ne az ne de fazla	Kısmen fazla	Fazla	Çok fazla	Çok çok fazla
Konu	1	2	3	4	5	6	7	8	9
İlişkisel Operatörler									
İf ifadesi									
İf / Else ifadesi									
İç içe koşul ifadeleri									

Bu dersin öğrenme ortamında önemli ve önemsiz bilgileri ayırt etmek ne kadar zordur?

	Çok çok az	Çok az	Az	Kısmen az	Ne az ne de fazla	Kısmen fazla	Fazla	Çok fazla	Çok çok fazla
Konu	1	2	3	4	5	6	7	8	9
İlişkisel Operatörler									
İf ifadesi									
İf / Else ifadesi									
İç içe koşul ifadeleri									

Dersin öğrenme ortamı sizin için ne kadar zordur?

	Çok çok az	Çok az	Az	Kısmen az	Ne az ne de fazla	Kısmen fazla	Fazla	Çok fazla	Çok çok fazla
Konu	1	2	3	4	5	6	7	8	9
İlişkisel Operatörler									
İf ifadesi									
İf / Else ifadesi									
İç içe koşul ifadeleri									

Derste yapılan en son örneği takip etmek için sarf ettiğiniz çabayı aşağıda gösterilen ölçek üzerinde belirtiniz.

	Çok çok az	Çok az	Az	Kısmen az	Ne az ne de fazla	Kısmen fazla	Fazla	Çok fazla	Çok çok fazla
Konu	1	2	3	4	5	6	7	8	9
İlişkisel Operatörler									
İf ifadesi									
İf / Else ifadesi									
İç içe koşul ifadeleri									

Öğrenme ortamında ihtiyacınız olan tüm bilgileri toplamak sizin için ne kadar zordur?

	Çok çok az	Çok az	Az	Kısmen az	Ne az ne de fazla	Kısmen fazla	Fazla	Çok fazla	Çok çok fazla
Konu	1	2	3	4	5	6	7	8	9
İlişkisel Operatörler									
İf ifadesi									
İf / Else ifadesi									
İç içe koşul ifadeleri									

**Hafta-4:** Belirlenen programlama dilinde kontrol yapılarını kullanarak programlar geliştirir.

Hafta-4 Konu Başlıkları	
• Çok yönlü karar ifadeleri	
• Zincirleme durum ifadeleri	
• Koşullu ifadeler	

Aşağıdaki her bir konu sizin için şu anda ne kadar zor gelmektedir?

	Çok çok az	Çok az	Az	Kısmen az	Ne az ne de fazla	Kısmen fazla	Fazla	Çok fazla	Çok çok fazla
Konu	1	2	3	4	5	6	7	8	9
Çok yönlü karar ifadeleri									
Zincirleme durum ifadeleri									
Koşullu ifadeler									

Bu dersin öğrenme ortamında önemli ve önemsiz bilgileri ayırt etmek ne kadar zordur?

	Çok çok az	Çok az	Az	Kısmen az	Ne az ne de fazla	Kısmen fazla	Fazla	Çok fazla	Çok çok fazla
Konu	1	2	3	4	5	6	7	8	9
Çok yönlü karar ifadeleri									
Zincirleme durum ifadeleri									
Koşullu ifadeler									

Dersin öğrenme ortamı sizin için ne kadar zordur?

	Çok çok az	Çok az	Az	Kısmen az	Ne az ne de fazla	Kısmen fazla	Fazla	Çok fazla	Çok çok fazla
Konu	1	2	3	4	5	6	7	8	9
Çok yönlü karar ifadeleri									
Zincirleme durum ifadeleri									
Koşullu ifadeler									

Derste yapılan en son örneği takip etmek için sarf ettiğiniz çabayı aşağıda gösterilen ölçek üzerinde belirtiniz.

	Çok çok az	Çok az	Az	Kısmen az	Ne az ne de fazla	Kısmen fazla	Fazla	Çok fazla	Çok çok fazla
Konu	1	2	3	4	5	6	7	8	9
Çok yönlü karar ifadeleri									
Zincirleme durum ifadeleri									
Koşullu ifadeler									

Öğrenme ortamında ihtiyacınız olan tüm bilgileri toplamak sizin için ne kadar zordur?

	Çok çok az	Çok az	Az	Kısmen az	Ne az ne de fazla	Kısmen fazla	Fazla	Çok fazla	Çok çok fazla
Konu	1	2	3	4	5	6	7	8	9
Çok yönlü karar ifadeleri									
Zincirleme durum ifadeleri									
Koşullu ifadeler									

**Hafta-5:** Belirlenen programlama dilinde tekrarlı yapıları kullanarak programlar geliştirir.

Hafta-5 Konu Başlıkları
• Döngü yapıları
• For döngüsü
• İç içe döngüler

Aşağıdaki her bir konu sizin için şu anda ne kadar zor gelmektedir?

	Çok çok az	Çok az	Az	Kısmen az	Ne az ne de fazla	Kısmen fazla	Fazla	Çok fazla	Çok çok fazla
Konu	1	2	3	4	5	6	7	8	9
Döngü yapıları									
For döngüsü									
İç içe döngüler									

Bu dersin öğrenme ortamında önemli ve önemsiz bilgileri ayırt etmek ne kadar zordur?

	Çok çok az	Çok az	Az	Kısmen az	Ne az ne de fazla	Kısmen fazla	Fazla	Çok fazla	Çok çok fazla
Konu	1	2	3	4	5	6	7	8	9
Döngü yapıları									
For döngüsü									
İç içe döngüler									

Dersin öğrenme ortamı sizin için ne kadar zordur?

	Çok çok az	Çok az	Az	Kısmen az	Ne az ne de fazla	Kısmen fazla	Fazla	Çok fazla	Çok çok fazla
Konu	1	2	3	4	5	6	7	8	9
Döngü yapıları									
For döngüsü									
İç içe döngüler									

Derste yapılan en son örneği takip etmek için sarf ettiğiniz çabayı aşağıda gösterilen ölçek üzerinde belirtiniz.

	Çok çok az	Çok az	Az	Kısmen az	Ne az ne de fazla	Kısmen fazla	Fazla	Çok fazla	Çok çok fazla
Konu	1	2	3	4	5	6	7	8	9
Döngü yapıları									
For döngüsü									
İç içe döngüler									

Öğrenme ortamında ihtiyacınız olan tüm bilgileri toplamak sizin için ne kadar zordur?

	Çok çok az	Çok az	Az	Kısmen az	Ne az ne de fazla	Kısmen fazla	Fazla	Çok fazla	Çok çok fazla
Konu	1	2	3	4	5	6	7	8	9
Döngü yapıları									
For döngüsü									
İç içe döngüler									

**Hafta-6:** Belirlenen programlama dilinde tekrarlı yapıları kullanarak programlar geliştirir.

Hafta-6 Konu Başlıkları	
•	While döngüsü
•	Belirli ve belirsiz döngüler
•	Döngüden çıkma komutları

Aşağıdaki her bir konu sizin için şu anda ne kadar zor gelmektedir?

	Çok çok az	Çok az	Az	Kısmen az	Ne az ne de fazla	Kısmen fazla	Fazla	Çok fazla	Çok çok fazla
Konu	1	2	3	4	5	6	7	8	9
While döngüsü									
Belirli ve belirsiz döngüler									
Döngüden çıkma komutları									

Bu dersin öğrenme ortamında önemli ve önemsiz bilgileri ayırt etmek ne kadar zordur?

	Çok çok az	Çok az	Az	Kısmen az	Ne az ne de fazla	Kısmen fazla	Fazla	Çok fazla	Çok çok fazla
Konu	1	2	3	4	5	6	7	8	9
While döngüsü									
Belirli ve belirsiz döngüler									
Döngüden çıkma komutları									

Dersin öğrenme ortamı sizin için ne kadar zordur?

	Çok çok az	Çok az	Az	Kısmen az	Ne az ne de fazla	Kısmen fazla	Fazla	Çok fazla	Çok çok fazla
Konu	1	2	3	4	5	6	7	8	9
While döngüsü									
Belirli ve belirsiz döngüler									
Döngüden çıkma komutları									

Derste yapılan en son örneği takip etmek için sarf ettiğiniz çabayı aşağıda gösterilen ölçek

üzerinde belirtiniz.

	Çok çok az	Çok az	Az	Kısmen az	Ne az ne de fazla	Kısmen fazla	Fazla	Çok fazla	Çok çok fazla
Konu	1	2	3	4	5	6	7	8	9
While döngüsü									
Belirli ve belirsiz döngüler									
Döngüden çıkma komutları									

Öğrenme ortamında ihtiyacınız olan tüm bilgileri toplamak sizin için ne kadar zordur?

	Çok çok az	Çok az	Az	Kısmen az	Ne az ne de fazla	Kısmen fazla	Fazla	Çok fazla	Çok çok fazla
Konu	1	2	3	4	5	6	7	8	9
While döngüsü									
Belirli ve belirsiz döngüler									
Döngüden çıkma komutları									

**Hafta-7: Belirlenen programlama dilinde parametre almayan fonksiyon içeren programlar geliştirir.**

Hafta-7 Konu Başlıkları
• Fonksiyonlara giriş
• Fonksiyona parametre tanımlama

Aşağıdaki her bir konu sizin için şu anda ne kadar zor gelmektedir?

	Çok çok az	Çok az	Az	Kısmen az	Ne az ne de fazla	Kısmen fazla	Fazla	Çok fazla	Çok çok fazla
Konu	1	2	3	4	5	6	7	8	9
Fonksiyonlara giriş									
Fonksiyona parametre tanımlama									

Bu dersin öğrenme ortamında önemli ve önemsiz bilgileri ayırt etmek ne kadar zordur?

	Çok çok az	Çok az	Az	Kısmen az	Ne az ne de fazla	Kısmen fazla	Fazla	Çok fazla	Çok çok fazla
Konu	1	2	3	4	5	6	7	8	9
Fonksiyonlara giriş									
Fonksiyona parametre tanımlama									

Dersin öğrenme ortamı sizin için ne kadar zordur?

	Çok çok az	Çok az	Az	Kısmen az	Ne az ne de fazla	Kısmen fazla	Fazla	Çok fazla	Çok çok fazla
Konu	1	2	3	4	5	6	7	8	9
Fonksiyonlara giriş									
Fonksiyona parametre tanımlama									

Derste yapılan en son örneği takip etmek için sarf ettiğiniz çabayı aşağıda gösterilen ölçek üzerinde belirtiniz.

	Çok çok az	Çok az	Az	Kısmen az	Ne az ne de fazla	Kısmen fazla	Fazla	Çok fazla	Çok çok fazla
Konu	1	2	3	4	5	6	7	8	9
Fonksiyonlara giriş									
Fonksiyona parametre tanımlama									

Öğrenme ortamında ihtiyacınız olan tüm bilgileri toplamak sizin için ne kadar zordur?

	Çok çok az	Çok az	Az	Kısmen az	Ne az ne de fazla	Kısmen fazla	Fazla	Çok fazla	Çok çok fazla
Konu	1	2	3	4	5	6	7	8	9
Fonksiyonlara giriş									
Fonksiyona parametre tanımlama									

**Hafta-8:** Belirlenen programlama dilinde parametre almayan fonksiyon içeren programlar geliştirir.

Hafta-8 Konu Başlıkları
<ul style="list-style-type: none"> <li>Standart matematik fonksiyonları</li> <li>Fonksiyon tanımlama</li> </ul>

Aşağıdaki her bir konu sizin için şu anda ne kadar zor gelmektedir?



	Çok çok az	Çok az	Az	Kısmen az	Ne az ne de fazla	Kısmen fazla	Fazla	Çok fazla	Çok çok fazla
Konu	1	2	3	4	5	6	7	8	9
Standart matematik fonksiyonları									
Fonksiyon tanımlama									

Bu dersin öğrenme ortamında önemli ve önemsiz bilgileri ayırt etmek ne kadar zordur?

	Çok çok az	Çok az	Az	Kısmen az	Ne az ne de fazla	Kısmen fazla	Fazla	Çok fazla	Çok çok fazla
Konu	1	2	3	4	5	6	7	8	9
Standart matematik fonksiyonları									
Fonksiyon tanımlama									

Dersin öğrenme ortamı sizin için ne kadar zordur?

	Çok çok az	Çok az	Az	Kısmen az	Ne az ne de fazla	Kısmen fazla	Fazla	Çok fazla	Çok çok fazla
Konu	1	2	3	4	5	6	7	8	9
Standart matematik fonksiyonları									
Fonksiyon tanımlama									

Derste yapılan en son örneği takip etmek için sarf ettiğiniz çabayı aşağıda gösterilen ölçek üzerinde belirtiniz.

	Çok çok az	Çok az	Az	Kısmen az	Ne az ne de fazla	Kısmen fazla	Fazla	Çok fazla	Çok çok fazla
Konu	1	2	3	4	5	6	7	8	9
Standart matematik fonksiyonları									
Fonksiyon tanımlama									

Öğrenme ortamında ihtiyacınız olan tüm bilgileri toplamak sizin için ne kadar zordur?

	Çok çok az	Çok az	Az	Kısmen az	Ne az ne de fazla	Kısmen fazla	Fazla	Çok fazla	Çok çok fazla
Konu	1	2	3	4	5	6	7	8	9
Standart matematik fonksiyonları									
Fonksiyon tanımlama									

**Hafta-9:** Belirlenen programlama dilinde dizgi tipinde veri tanımlar.

Hafta-11 Konu Başlıkları	
•	Liste kavramı
•	Değerler arası gezinme
•	Liste üyeliği
•	Listeye değer atama ve eşitleme

Aşağıdaki her bir konu sizin için şu anda ne kadar zor gelmektedir?

	Çok çok az	Çok az	Az	Kısmen az	Ne az ne de fazla	Kısmen fazla	Fazla	Çok fazla	Çok çok fazla
Konu	1	2	3	4	5	6	7	8	9
Liste kavramı									
Değerler arası gezinme									
Liste üyeliği									
Listeye değer atama ve eşitleme									

Bu dersin öğrenme ortamında önemli ve önemsiz bilgileri ayırt etmek ne kadar zordur?

	Çok çok az	Çok az	Az	Kısmen az	Ne az ne de fazla	Kısmen fazla	Fazla	Çok fazla	Çok çok fazla
Konu	1	2	3	4	5	6	7	8	9
Liste kavramı									
Değerler arası gezinme									
Liste üyeliği									
Listeye değer atama ve eşitleme									

Dersin öğrenme ortamı sizin için ne kadar zordur?

	Çok çok az	Çok az	Az	Kısmen az	Ne az ne de fazla	Kısmen fazla	Fazla	Çok fazla	Çok çok fazla
Konu	1	2	3	4	5	6	7	8	9
Liste kavramı									
Değerler arası gezinme									
Liste üyeliği									
Listeye değer atama ve eşitleme									

Derste yapılan en son örneği takip etmek için sarf ettiğiniz çabayı aşağıda gösterilen ölçek üzerinde belirtiniz.

	Çok çok az	Çok az	Az	Kısmen az	Ne az ne de fazla	Kısmen fazla	Fazla	Çok fazla	Çok çok fazla
Konu	1	2	3	4	5	6	7	8	9
Liste kavramı									
Değerler arası gezinme									
Liste üyeliği									
Listeye değer atama ve eşitleme									

Öğrenme ortamında ihtiyacınız olan tüm bilgileri toplamak sizin için ne kadar zordur?

	Çok çok az	Çok az	Az	Kısmen az	Ne az ne de fazla	Kısmen fazla	Fazla	Çok fazla	Çok çok fazla
Konu	1	2	3	4	5	6	7	8	9
Liste kavramı									
Değerler arası gezinme									
Liste üyeliği									
Listeye değer atama ve eşitleme									

**Hafta-10:** Belirlenen programlama dilinde dizgi tipinde veri tanımlar.

Hafta-12 Konu Başlıkları
• Dilimleme
• Listedeki eleman çıkarma
• Listede kullanılan yöntemler

Aşağıdaki her bir konu sizin için şu anda ne kadar zor gelmektedir?

	Çok çok az	Çok az	Az	Kısmen az	Ne az ne de fazla	Kısmen fazla	Fazla	Çok fazla	Çok çok fazla
Konu	1	2	3	4	5	6	7	8	9
Dilimleme									
Listeden eleman çıkarma									
Listede kullanılan yöntemler									

Bu dersin öğrenme ortamında önemli ve önemsiz bilgileri ayırt etmek ne kadar zordur?

	Çok çok az	Çok az	Az	Kısmen az	Ne az ne de fazla	Kısmen fazla	Fazla	Çok fazla	Çok çok fazla
Konu	1	2	3	4	5	6	7	8	9
Dilimleme									
Listeden eleman çıkarma									
Listede kullanılan yöntemler									

Dersin öğrenme ortamı sizin için ne kadar zordur?

	Çok çok az	Çok az	Az	Kısmen az	Ne az ne de fazla	Kısmen fazla	Fazla	Çok fazla	Çok çok fazla
Konu	1	2	3	4	5	6	7	8	9
Dilimleme									
Listeden eleman çıkarma									
Listede kullanılan yöntemler									

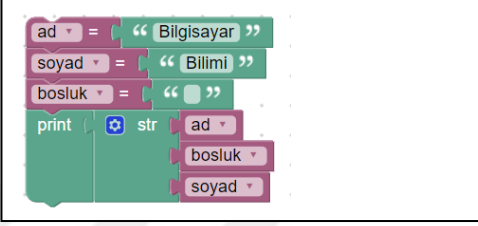
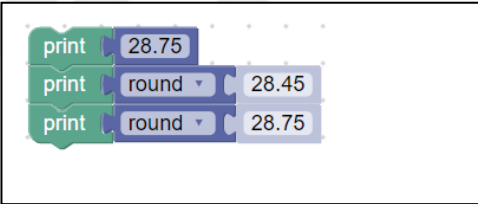
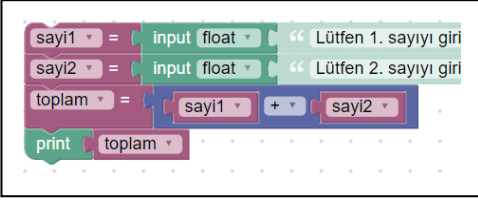
Derste yapılan en son örneği takip etmek için sarf ettiğiniz çabayı aşağıda gösterilen ölçek üzerinde belirtiniz.

	Çok çok az	Çok az	Az	Kısmen az	Ne az ne de fazla	Kısmen fazla	Fazla	Çok fazla	Çok çok fazla
Konu	1	2	3	4	5	6	7	8	9
Dilimleme									
Listeden eleman çıkarma									
Listede kullanılan yöntemler									

Öğrenme ortamında ihtiyacınız olan tüm bilgileri toplamak sizin için ne kadar zordur?

	Çok çok az	Çok az	Az	Kısmen az	Ne az ne de fazla	Kısmen fazla	Fazla	Çok fazla	Çok çok fazla
Konu	1	2	3	4	5	6	7	8	9
Dilimleme									
Listeden eleman çıkarma									
Listede kullanılan yöntemler									

### Ek-3. Başarı Testi (Blockly)

PROGRAMLAMA BAŞARI TESTİ	
Türkler İMKB Sosyal Bilimler Lisesi	
Hazırlık Sınıfları Bilgisayar Bilimi Dersi 2.Yazılı Soruları	
<b>Yönerge</b> :Aşağıda 25 adet soru vardır. Sorular iki farklı gösterimde sunulmuş olup sağ tarafta Blockly programı sol tarafta ise metin tabanlı Python programlama dili editörü kullanılmıştır. Her doğru cevap 4 puandır. Soruların cevapları için cevap kağıdını kullanınız.Sınav süresi 40 dakikadır.	
1-)Aşağıdakilerden hangisi değişken ismi olabilir? a-)Okul No b-)İf, else c-)tcKimlikNo d-)1Soru? e-)pass	1-)Aşağıdakilerden hangisi değişken ismi olabilir? a-)Okul No b-)İf, else c-)tcKimlikNo d-)1Soru? e-)pass
2-) Aşağıdaki programın çıktısı nedir ? <pre>ad = "Bilgisayar" soyad = "Bilimi" bosluk = " " print ad + bosluk + soyad</pre> a-)BilgisayarBilimi b-)BilimiBilgisayar c-)Bilgisayar Bilimi d-)BilgisayarBilgisayar e-)BilimiBilimi	2-) Aşağıdaki programın çıktısı nedir ?  a-)BilgisayarBilimi b-)BilimiBilgisayar c-)Bilgisayar Bilimi d-)BilgisayarBilgisayar e-)BilimiBilimi
3-)Aşağıdaki programın çıktısı nedir ? <pre>print(28.75) print(round(28.45)) print(round(28.75))</pre> a-)28.75 b-) 2875 c-)28,75 d-)28.75 e-)2875 28 28 28.45 29 29 29 29 28.75 28 28	3-)Aşağıdaki programın çıktısı nedir ?  a-)28.75 b-) 2875 c-)28,75 d-)28.75 e-)2875 28 28 28.45 29 29 29 29 28.75 28 28
4-)Aşağıdaki programın işlevi nedir ? <pre>sayi1 = float(input('Lütfen 1. sayıyı giriniz :')) sayi2 = float(input ('Lütfen 2. sayıyı giriniz :')) toplam = sayi1 + sayi2</pre> a-)Kullanıcıdan iki metin alır ve birleşimini yazar. b-)Kullanıcıdan iki tamsayı alır ve toplamını yazar. c-)Kullanıcıdan iki rasyonel sayı alır ve toplamını yazar. d-)Kullanıcıdan iki negatif sayı alır ve toplamını yazar. e-)Kullanıcıdan iki pozitif sayı alır ve toplamını yazar.	4-)Aşağıdaki programın işlevi nedir ?  a-)Kullanıcıdan iki metin alır ve birleşimini yazar. b-)Kullanıcıdan iki tamsayı alır ve toplamını yazar. c-)Kullanıcıdan iki rasyonel sayı alır ve toplamını yazar. d-)Kullanıcıdan iki negatif sayı alır ve toplamını yazar. e-)Kullanıcıdan iki pozitif sayı alır ve toplamını yazar.

5-)Aşağıdaki programın çıktısı nedir ?

```
x = 5
y = 5
x +=15
y = y + 15
print(str('x= ') + str(x))
print(str('y= ') + str(y))
```

- a-)x=15 y=20    b-)x= 5 y=20    c-)x=20 y=20    d-)x=25 y=25    e-)x=30 y=30

5-)Aşağıdaki programın çıktısı nedir ?

- a-)x=15 y=20    b-)x= 5 y=20    c-)x=20 y=20    d-)x=25 y=25    e-)x=30 y=30

6-)Aşağıdaki programın çıktısı nedir ?

```
x = 10
y = 20
z = 30
x = (y + z) + z * x
print(x)
```

- a-)250    b-)300    c-)350    d-)400    e-)375

6-)Aşağıdaki programın çıktısı nedir ?

- a-)250    b-)300    c-)350    d-)400    e-)375

Tablo-1

Fonksiyon	Tanım	Örnek	Sonuç
Sum(list)	Birkaç sayının toplam değerini döndürür.	Sum(3,5,8)	16

Tablo-1

Fonksiyon	Tanım	Örnek	Sonuç
Sum(list)	Birkaç sayının toplam değerini döndürür.	Sum(3,5,8)	16

Tablo-2

Değişken	Fonksiyon	Çıktı
s=25 a=15 y=s+10	Sum(s,a,y)	?

Tablo-2

Değişken	Fonksiyon	Çıktı
s=25 a=15 y=s+10	Sum(s,a,y)	?

7-)Yukarıda Tablo-1’de istatistiksel bir fonksiyona örnek verilmiştir. Tablo-2’deki değişkenlerle fonksiyon çalıştırıldığında komut çıktı değeri ne olur?

- a-)25    b-)90    c-)50    d-)75    e-)80

7-)Yukarıda Tablo-1’de istatistiksel bir fonksiyona örnek verilmiştir. Tablo-2’deki değişkenlerle fonksiyon çalıştırıldığında komut çıktı değeri ne olur?

- a-)25    b-)90    c-)50    d-)75    e-)80

```
x = 2
y = 3
a = 6
b = 8
c = 10
print(c % y)
print(y ** b)
```

8-) Yukarıdaki program için aşağıdakilerden hangisi doğrudur ?

- a-)c'nin y'ye göre yüzdesini alır ve yazdırır.  
b-)y'nin b ile çarpımını yazdırır.  
c-)c'nin y'ye göre modunu, y'nin b'ninci kuvvetini alır ve yazdırır.  
d-)y'nin c'ye göre modunu alır ve b ile çarpımını yazdırır.  
e-)c'nin b ile çarpımını sağlar.

8-) Yukarıdaki program için aşağıdakilerden hangisi doğrudur ?

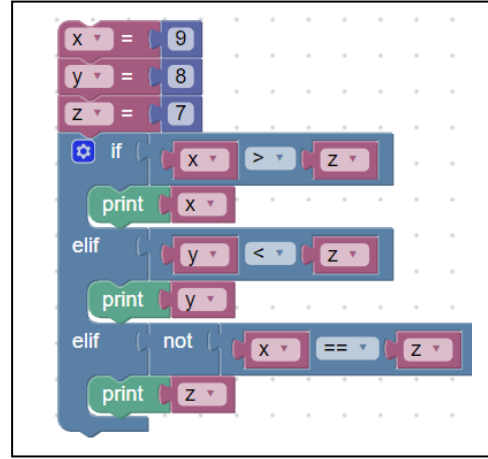
- a-)c'nin y'ye göre yüzdesini alır ve yazdırır.  
b-)y'nin b ile çarpımını yazdırır.  
c-)c'nin y'ye göre modunu, y'nin b'ninci kuvvetini alır ve yazdırır.  
d-)y'nin c'ye göre modunu alır ve b ile çarpımını yazdırır.  
e-)c'nin b ile çarpımını sağlar.

9-) Aşağıdaki programın çıktısı nedir ?

```
x=9
y=8
z=7
if (x > z) :
print (x)
elif(y < z) :
print (y)
elif (not (x==z)):
print (z)
```

a-) 7 b-)8 c-)9 d-)99 e-)Null

9-) Aşağıdaki programın çıktısı nedir ?



The Scratch code for question 9 is as follows: x = 9, y = 8, z = 7. An if-elif-elif block is used. The first condition is 'x > z', which is true, so 'print x' is executed. The other conditions are not checked.

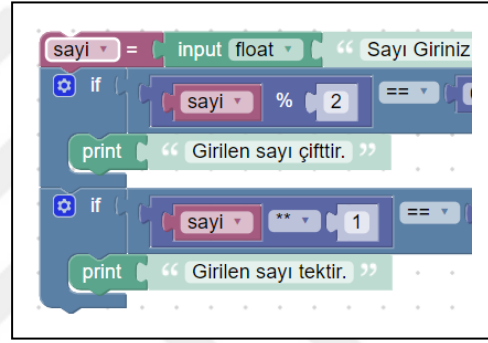
a-) 7 b-)8 c-)9 d-)99 e-)Null

10-) Aşağıdaki programda yapılan hata hangi şıkta doğru olarak verilmiştir ?

```
sayi = float(input('Sayı Giriniz : '))
if sayi % 2 == 0:
print('Girilen sayı çifttir.')
if sayi ** 1 == 1:
print('Girilen sayı tektir.')
```

a-) sayi = float(input('Sayı Giriniz : '))---- **Veri tipi hatası**  
b-) if sayi % 2 == 0: ----- **Koşul yapısındaki hata**  
c-) print('Girilen sayı çifttir.')----- **Print komutundaki hata**  
d-) if sayi \*\* 1 == 1: ----- **Operatör hatası**  
e-) print('Girilen sayı tektir.')----- **Print komutundaki hata**

10-) Aşağıdaki programda yapılan hata hangi şıkta doğru olarak verilmiştir ?



The Scratch code for question 10 is as follows: sayi = input float 'Sayı Giriniz'. An if block checks 'sayi % 2 == 0'. If true, it prints 'Girilen sayı çifttir.'. Another if block checks 'sayi \*\* 1 == 1'. If true, it prints 'Girilen sayı tektir.'.

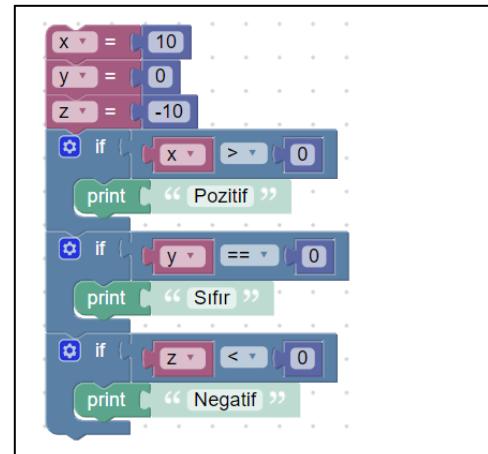
a-) sayi = float(input('Sayı Giriniz : '))---- **Veri tipi hatası**  
b-) if sayi % 2 == 0: ----- **Koşul yapısındaki hata**  
c-) print('Girilen sayı çifttir.')----- **Print komutundaki hata**  
d-) if sayi \*\* 1 == 1: ----- **Operatör hatası**  
e-) print('Girilen sayı tektir.')----- **Print komutundaki hata**

11-) Aşağıdaki programın çıktısı nedir ?

```
x = 10
y = 0
z = -10
if x > 0:
print('Pozitif')
if y == 0:
print('Sıfır')
if z < 0:
print('Negatif')
```

a-) Pozitif b-)Negatif c-)Sıfır d-)Pozitif e-)Negatif  
Negatif Pozitif Negatif Sıfır Sıfır  
Sıfır Sıfır Pozitif Negatif Pozitif

11-) Aşağıdaki programın çıktısı nedir ?



The Scratch code for question 11 is as follows: x = 10, y = 0, z = -10. Three if blocks are used. The first checks 'x > 0', which is true, so 'Pozitif' is printed. The second checks 'y == 0', which is true, so 'Sıfır' is printed. The third checks 'z < 0', which is true, so 'Negatif' is printed.

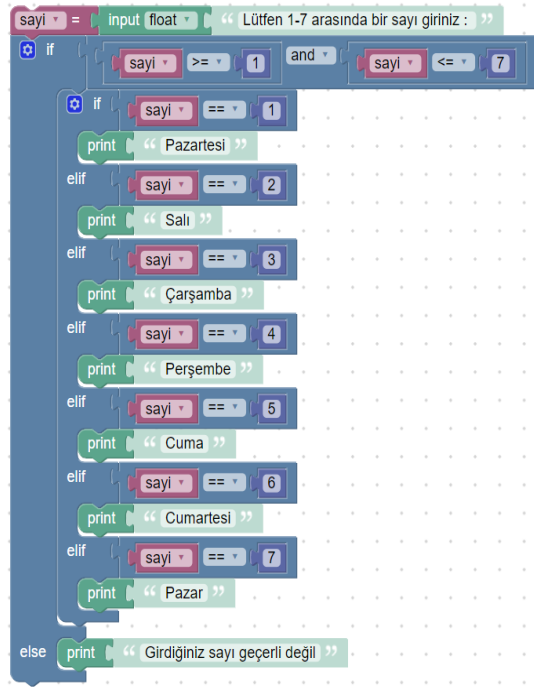
a-) Pozitif b-)Negatif c-)Sıfır d-)Pozitif e-)Negatif  
Negatif Pozitif Negatif Sıfır Sıfır  
Sıfır Sıfır Pozitif Negatif Pozitif

12-) Aşağıdaki programın işlevi nedir ?

```
if sayi >= 1 and sayi <= 7:  
    if sayi == 1:  
        print ("Pazartesi")  
    elif sayi == 2:  
        print ("Salı")  
    elif sayi == 3:  
        print ("Çarşamba")  
    elif sayi == 4:  
        print ("Perşembe")  
    elif sayi == 5:  
        print ("Cuma")  
    elif sayi == 6:  
        print ("Cumartesi")  
    elif sayi == 7:  
        print ("Pazar")  
else:  
    print("Geçersiz bir değer girdiniz!")
```

- a-) Haftanın günlerini ekrana yazdıran bir programdır.
- b-)1-7 arasındaki sayıları karıştıran ve günlerle eşleştiren bir programdır.
- c-)Girilen sayıya göre haftanın hangi günü olduğunu ekrana getiren programdır.
- d-)1-7 arasındaki sayıları ekrana yazdıran bir programdır.
- e-)1-7 arasındaki sayıları ve haftanın günlerini ekrana yazdıran programdır.

12-) Aşağıdaki programın işlevi nedir ?



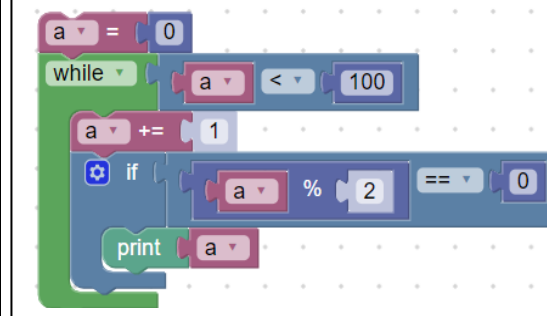
- a-) Haftanın günlerini ekrana yazdıran bir programdır.
- b-)1-7 arasındaki sayıları karıştıran ve günlerle eşleştiren bir programdır.
- c-)Girilen sayıya göre haftanın hangi günü olduğunu ekrana getiren programdır.
- d-)1-7 arasındaki sayıları ekrana yazdıran bir programdır.
- e-)1-7 arasındaki sayıları ve haftanın günlerini ekrana yazdıran programdır.

13-)Aşağıdaki programın işlevi nedir ?

```
a = 0  
while a < 100:  
    a += 1  
    if a % 2 == 0:  
        print(a)
```

- a-)Çıktı olarak "a=0" verir.
- b-)Çıktı olarak 2-100 arasındaki çift sayıları verir.
- c-) Çıktı olarak 2-100 arasındaki tek sayıları verir.
- d-)Sonsuz döngüye girer.
- e-)Çıktı olarak 1-100 arasındaki sayıları yazar.

13-)Aşağıdaki programın işlevi nedir ?



- a-)Çıktı olarak "a=0" verir.
- b-)Çıktı olarak 2-100 arasındaki çift sayıları verir.
- c-) Çıktı olarak 2-100 arasındaki tek sayıları verir.
- d-)Sonsuz döngüye girer.
- e-)Çıktı olarak 1-100 arasındaki sayıları yazar.



14-) Aşağıdaki programın çıktısı nedir ?

```
sayilar = '123456789'
for k in sayilar:
    print(int(k) * 2)
```

- |       |       |       |        |       |
|-------|-------|-------|--------|-------|
| a-) 2 | b-) 1 | c-) 9 | d-) 18 | e-) 3 |
| 4     | 2     | 8     | 16     | 6     |
| 6     | 3     | 7     | 14     | 9     |
| 8     | 4     | 6     | 12     | 12    |
| 10    | 5     | 5     | 10     | 15    |
| 12    | 6     | 4     | 8      | 18    |
| 14    | 7     | 3     | 6      | 21    |
| 16    | 8     | 2     | 4      | 24    |
| 18    | 9     | 1     | 2      | 27    |

14-) Aşağıdaki programın çıktısı nedir ?

```
sayilar = "123456789"
for k in sayilar:
    print(k * 2)
```

- |       |       |       |        |       |
|-------|-------|-------|--------|-------|
| a-) 2 | b-) 1 | c-) 9 | d-) 18 | e-) 3 |
| 4     | 2     | 8     | 16     | 6     |
| 6     | 3     | 7     | 14     | 9     |
| 8     | 4     | 6     | 12     | 12    |
| 10    | 5     | 5     | 10     | 15    |
| 12    | 6     | 4     | 8      | 18    |
| 14    | 7     | 3     | 6      | 21    |
| 16    | 8     | 2     | 4      | 24    |
| 18    | 9     | 1     | 2      | 27    |

15-) Aşağıdaki programın çıktısı nedir ?

```
for i in range (1,5,1):
    for y in range (1,3,1):
        print i,'*',y,'=',i*y
```

- |               |               |               |               |               |
|---------------|---------------|---------------|---------------|---------------|
| a-) 1 * 1 = 1 | b-) 1 * 1 = 1 | c-) 4 * 2 = 8 | d-) 1 * 1 = 1 | e-) 1 * 1 = 1 |
| 1 * 2 = 2     | 2 * 2 = 4     | 3 * 2 = 6     | 1 * 2 = 2     | 2 * 2 = 4     |
| 2 * 1 = 2     | 3 * 1 = 3     | 3 * 1 = 3     | 2 * 1 = 2     | 2 * 2 = 4     |
| 2 * 2 = 4     | 2 * 2 = 4     | 2 * 2 = 4     | 2 * 2 = 4     | 2 * 2 = 4     |
| 3 * 1 = 3     | 3 * 1 = 3     | 3 * 2 = 6     | 3 * 3 = 9     | 3 * 3 = 9     |
| 3 * 2 = 6     | 3 * 2 = 6     | 4 * 1 = 4     | 3 * 2 = 6     | 3 * 3 = 9     |
| 4 * 1 = 4     | 4 * 1 = 4     | 4 * 2 = 8     | 4 * 1 = 4     | 4 * 4 = 16    |
| 4 * 2 = 8     | 4 * 2 = 8     |               | 4 * 3 = 12    | 4 * 4 = 16    |

15-) Aşağıdaki programın çıktısı nedir ?

```
for i in range (1, 5, 1):
    for j in range (1, 3, 1):
        print str(i) * j
```

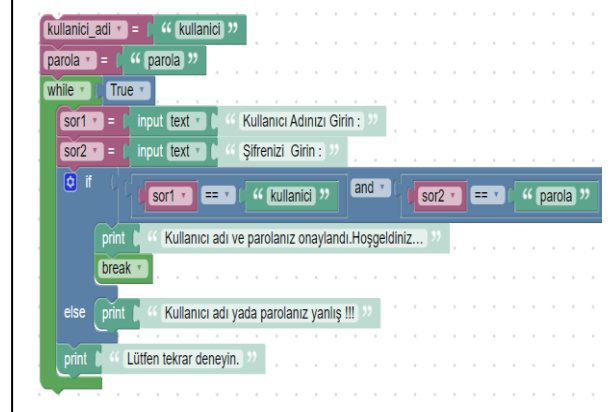
- |               |               |               |               |               |
|---------------|---------------|---------------|---------------|---------------|
| a-) 1 * 1 = 1 | b-) 1 * 1 = 1 | c-) 4 * 2 = 8 | d-) 1 * 1 = 1 | e-) 1 * 1 = 1 |
| 1 * 2 = 2     | 2 * 2 = 4     | 4 * 1 = 4     | 1 * 2 = 2     | 2 * 2 = 4     |
| 2 * 1 = 2     | 3 * 1 = 3     | 3 * 2 = 6     | 2 * 1 = 2     | 2 * 2 = 4     |
| 2 * 2 = 4     | 2 * 2 = 4     | 3 * 1 = 3     | 2 * 2 = 4     | 2 * 2 = 4     |
| 3 * 1 = 3     | 3 * 1 = 3     | 2 * 2 = 4     | 3 * 3 = 9     | 3 * 3 = 9     |
| 3 * 2 = 6     | 3 * 2 = 6     | 3 * 2 = 6     | 3 * 2 = 6     | 3 * 3 = 9     |
| 4 * 1 = 4     | 4 * 1 = 4     | 4 * 1 = 4     | 4 * 1 = 4     | 4 * 4 = 16    |
| 4 * 2 = 8     | 4 * 2 = 8     | 4 * 2 = 8     | 4 * 3 = 12    | 4 * 4 = 16    |

16-)Aşağıdaki programda “break” komutu ne iş yapar ?

```
kullanici_adi = 'kullanici'
parola = 'parola'
while True:
    sor1 = input('Kullanıcı Adınızı Girin :')
    sor2 = input('Şifrenizi Girin :')
    if sor1 == 'kullanici' and sor2 == 'parola':
        print('Kullanıcı adı ve parolanız onaylandı.Hoşgeldiniz !!!')
        break
    else:
        print('Kullanıcı adı yada parolanız yanlış !!!')
        print('Lütfen tekrar deneyin.')
```

- a-)Programı sonlandırmaya yarar.
- b-)Döngüyü başa döndürür.
- c-)”True” komutu ile beraber kullanılır ve programı başa döndürür.
- d-)”if “ koşulunu sona erdirir.
- e-)”else” komutunu başlatır.

16-)Aşağıdaki programda “break” komutu ne iş yapar ?



```
kullanici_adi = "kullanici"
parola = "parola"
while True:
    sor1 = input text "Kullanıcı Adınızı Girin : "
    sor2 = input text "Şifrenizi Girin : "
    if sor1 == "kullanici" and sor2 == "parola":
        print "Kullanıcı adı ve parolanız onaylandı.Hoşgeldiniz..."
        break
    else:
        print "Kullanıcı adı yada parolanız yanlış !!!"
        print "Lütfen tekrar deneyin. "
```

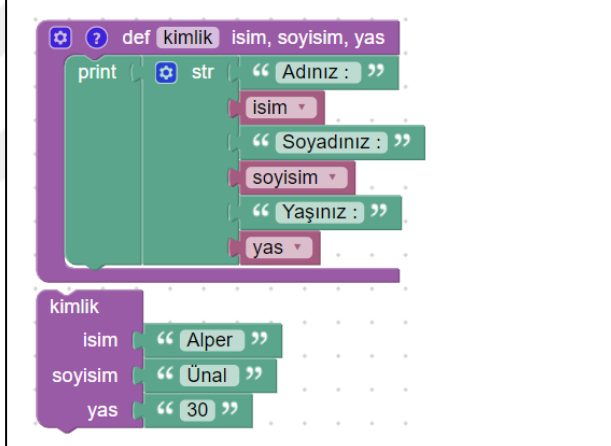
- a-)Programı sonlandırmaya yarar.
- b-)Döngüyü başa döndürür.
- c-)”True” komutu ile beraber kullanılır ve programı başa döndürür.
- d-)”if “ koşulunu sona erdirir.
- e-)”else” komutunu başlatır.

17-) Aşağıdaki programın çıktısı nedir ?

```
def kimlik(isim,soyisim,yas):
    print "Adınız :",isim,"Soyadınız"
    ,soyisim," Yaşınız :",yas
    kimlik('Alper','Ünal',30)
```

- a-) Alper Ünal 30
- b-)Adınız : Soyadınız : Yas :
- c-)Adınız : Alper Soyadınız : Ünal Yas :30
- d-)’Alper’,’Ünal’,’30’
- e-)kimlik(isim,soyisim,yas)

17-) Aşağıdaki programın çıktısı nedir ?



```
def kimlik isim, soyisim, yas
    print str "Adınız : "
    isim
    " Soyadınız : "
    soyisim
    " Yaşınız : "
    yas
kimlik
    isim " Alper "
    soyisim " Ünal "
    yas " 30 "
```

- a-) Alper Ünal 30
- b-)Adınız : Soyadınız : Yas :
- c-)Adınız : Alper Soyadınız : Ünal Yas :30
- d-)’Alper’,’Ünal’,’30’
- e-)kimlik(isim,soyisim,yas)

18-)Aşağıdaki programın çıktısı nedir ?

```
def isim(x):  
    a = x + str('Merhaba ')  
    b = x + str('Naber ')  
    c = x + str('Nasılsın')  
    return a + (b + c)  
print(isim('Ali '))
```

- a-) Merhaba Merhaba Merhaba
- b-)Nasılsın Nasılsın Nasılsın
- c-)Naber Naber Naber
- d-)Ali Merhaba Ali Naber Ali Nasılsın
- e-)Merhaba Naber Nasılsın

18-)Aşağıdaki programın çıktısı nedir ?

- a-) Merhaba Merhaba Merhaba
- b-)Nasılsın Nasılsın Nasılsın
- c-)Naber Naber Naber
- d-)Ali Merhaba Ali Naber Ali Nasılsın
- e-)Merhaba Naber Nasılsın

19-)Örnek bir sayı girişinde aşağıdaki programın çıktısı ne olur ?

```
def sonuc(notu):  
    if notu >= 50:  
        print('Tebrikler ! Dersten geçtiniz...')  
    else:  
        print('Üzgünüm ! Dersten kaldınız...')  
    sonuc(float(input('Lütfen notunuzu girin : ')))
```

- a-)Lütfen notunuzu girin : 30      b-) Lütfen notunuzu girin : 30  
 sonuç()      (30)
- c-) Lütfen notunuzu girin : 30      d-) Lütfen notunuzu girin : 30  
 Dersten geçtiniz...      Tebrikler ! Dersten  
 geçtiniz...
- e-) Lütfen notunuzu girin : 30  
 Üzgünüm ! Dersten kaldınız...

19-) Örnek bir sayı girişinde aşağıdaki programın çıktısı ne olur ?

- a-)Lütfen notunuzu girin : 30      b-) Lütfen notunuzu girin : 30  
 sonuç()      (30)
- c-) Lütfen notunuzu girin : 30      d-) Lütfen notunuzu girin : 30  
 Dersten geçtiniz...      Tebrikler ! Dersten geçtiniz...
- e-) Lütfen notunuzu girin : 30  
 Üzgünüm ! Dersten kaldınız...

20-) Aşağıdaki programın çıktısı nedir ?

```
def com(a, b):  
    if a > b:  
        return 'Girdiğiniz 1.sayı büyüktür.'  
    return 'Girdiğiniz 2.sayı büyüktür.'  
  
print(com(float(input('Lütfen 1.sayıyı giriniz :')),  
float(input('Lütfen 2.sayıyı giriniz :'))))
```

- a-)Girdiğiniz 1.sayı büyüktür.  
Girdiğiniz 2.sayı büyüktür  
b-)Lütfen 1.sayıyı giriniz :3  
Lütfen 2.sayıyı giriniz :4  
c-) Lütfen 1.sayıyı giriniz : 3  
Lütfen 2.sayıyı giriniz : 4  
Girdiğiniz 1.sayı büyüktür.  
d-) Lütfen 1.sayıyı giriniz : 3  
Lütfen 2.sayıyı giriniz : 4  
Girdiğiniz 2.sayı büyüktür.  
e-) Lütfen 1.sayıyı giriniz :3  
Lütfen 2.sayıyı giriniz :4  
Girdiğiniz 1.sayı büyüktür.  
Girdiğiniz 2.sayı büyüktür.

20-) Aşağıdaki programın çıktısı nedir ?

- a-)Girdiğiniz 1.sayı büyüktür.  
Girdiğiniz 2.sayı büyüktür  
b-)Lütfen 1.sayıyı giriniz :3  
Lütfen 2.sayıyı giriniz :4  
c-) Lütfen 1.sayıyı giriniz : 3  
Lütfen 2.sayıyı giriniz : 4  
Girdiğiniz 1.sayı büyüktür.  
d-) Lütfen 1.sayıyı giriniz : 3  
Lütfen 2.sayıyı giriniz : 4  
Girdiğiniz 2.sayı büyüktür.  
e-) Lütfen 1.sayıyı giriniz :3  
Lütfen 2.sayıyı giriniz :4  
Girdiğiniz 1.sayı büyüktür.  
Girdiğiniz 2.sayı büyüktür.

21-) Aşağıdaki programın çıktısı nedir ?

```
def toplam(liste):  
    toplam = 0  
    for i in liste:  
        toplam += i  
    print(toplam)  
listee= [6, 8, 10]  
toplam(listee)
```

- a-) 12 b-)14 c-)18 d-)20 e-)24

21-) Aşağıdaki programın çıktısı nedir ?

- a-) 12 b-)14 c-)18 d-)20 e-)24

22-)Aşağıdaki programın çıktısı nedir ?

```
L = [1, 9, 14, 4, 6]  
S = []  
for i in L:  
    i +=2  
    S.append(i)  
print(S)
```

- a-)[1,9,14,4,6]  
b-)[1,9,4]  
c-)[6,4,14,9,1]  
d-)[2,9,28,8,6]  
e-)[3,11,16,6,8]

22-)Aşağıdaki programın çıktısı nedir ?

- a-)[1,9,14,4,6]  
b-)[1,9,4]  
c-)[6,4,14,9,1]  
d-)[2,9,28,8,6]  
e-)[3,11,16,6,8]

23-) Aşağıdaki programın çıktısı nedir ?

```
liste_1 = ['İstanbul', 'Ankara']  
liste_2 = ['Samsun', 'Trabzon']  
liste_3 = ['Erzurum', 'Erzincan']  
liste_4 = (liste_1 + liste_2) + liste_3  
print(liste_4)
```

- a-)[İstanbul, Ankara, Samsun, Trabzon]
- b-)[İstanbul, Ankara, Samsun, Trabzon, Erzurum, Erzincan]
- c-)[Samsun, Erzurum, Erzincan]
- d-)[İstanbul, Ankara, Erzurum, Erzincan]
- e-)[Erzurum, Erzincan]

23-) Aşağıdaki programın çıktısı nedir ?



The Scratch code defines four lists: liste\_1 with 'İstanbul' and 'Ankara', liste\_2 with 'Samsun' and 'Trabzon', liste\_3 with 'Erzurum' and 'Erzincan', and liste\_4 which is the concatenation of liste\_1, liste\_2, and liste\_3. A print block outputs liste\_4.

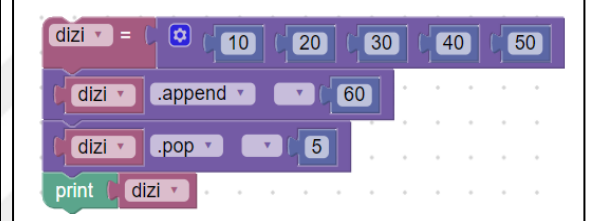
- a-)[İstanbul, Ankara, Samsun, Trabzon]
- b-)[İstanbul, Ankara, Samsun, Trabzon, Erzurum, Erzincan]
- c-)[Samsun, Erzurum, Erzincan]
- d-)[İstanbul, Ankara, Erzurum, Erzincan]
- e-)[Erzurum, Erzincan]

24-)Aşağıdaki programın çıktısı nedir ?

```
dizi=[10,20,30,40,50]  
dizi.append(60)  
dizi.pop(5)
```

- a-)[10,20,30]
- b-)[10,20,30,40,50]
- c-)[10,20,30,40,50,60]
- d-)[10]
- e-)[10,20,30,40,60]

24-)Aşağıdaki programın çıktısı nedir ?



The Scratch code defines a list dizi with elements 10, 20, 30, 40, 50. It then appends 60 to the list and removes the element at index 5. A print block outputs the final list dizi.

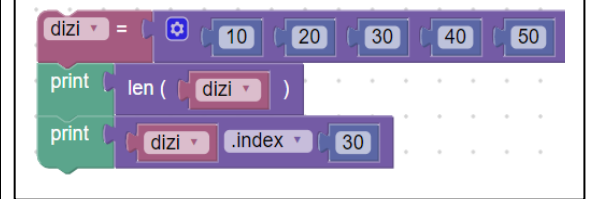
- a-)[10,20,30]
- b-)[10,20,30,40,50]
- c-)[10,20,30,40,50,60]
- d-)[10]
- e-)[10,20,30,40,60]

25-)Aşağıdaki programın çıktısı nedir ?

```
dizi = [10, 20, 30, 40, 50]  
print len(dizi)  
print dizi.index(30)
```

- |      |      |      |      |      |
|------|------|------|------|------|
| a-)3 | b-)4 | c-)5 | d-)4 | e-)5 |
| 1    | 2    | 2    | 3    | 4    |

25-)Aşağıdaki programın çıktısı nedir ?



The Scratch code defines a list dizi with elements 10, 20, 30, 40, 50. It then prints the length of the list and the index of the element 30. A print block outputs the index of 30.

- |      |      |      |      |      |
|------|------|------|------|------|
| a-)3 | b-)4 | c-)5 | d-)4 | e-)5 |
| 1    | 2    | 2    | 3    | 4    |

## Ek-4. Başarı Testi (Python Editör)

<b>PROGRAMLAMA BAŞARI TESTİ</b>															
Türkler İMKB Sosyal Bilimler Lisesi															
Hazırlık Sınıfları Bilgisayar Bilimi Dersi 2.Yazılı Soruları															
<p><b>Yönerge</b> :Aşağıda 25 adet soru vardır. Her doğru cevap 4 puandır. Soruların cevapları için cevap kağıdını kullanınız.Sınav süresi 40 dakikadır.</p>															
<p>1-)Aşağıdakilerden hangisi değişken ismi olabilir? a-)Okul No b-)İf, else c-)tcKimlikNo d-)1Soru? e-)pass</p>	<p>2-) Aşağıdaki programın çıktısı nedir ?</p> <pre style="border: 1px solid black; padding: 5px;">ad = "Bilgisayar" soyad = "Bilimi" bosluk = " " print ad + bosluk + soyad</pre> <p>a-)BilgisayarBilimi b-)BilimiBilgisayar c-)Bilgisayar Bilimi d-)BilgisayarBilgisayar e-)BilimiBilimi</p>														
<p>3-)Aşağıdaki programın çıktısı nedir ?</p> <pre style="border: 1px solid black; padding: 5px;">print(28.75) print(round(28.45)) print(round(28.75))</pre> <p>a-)28.75 b-) 2875 c-)28,75 d-)28.75 e-)2875 28 28 28.45 29 29 29 29 28.75 28 28</p>	<p>4-)Aşağıdaki programın işlevi nedir ?</p> <pre style="border: 1px solid black; padding: 5px;">sayi1 = float(input('Lütfen 1. sayıyı giriniz :')) sayi2 = float(input ('Lütfen 2. sayıyı giriniz :')) toplam = savi1 + savi2</pre> <p>a-)Kullanıcıdan iki metin alır ve birleşimini yazar. b-)Kullanıcıdan iki tamsayı alır ve toplamını yazar. c-)Kullanıcıdan iki rasyonel sayı alır ve toplamını yazar. d-)Kullanıcıdan iki negatif sayı alır ve toplamını yazar. e-)Kullanıcıdan iki pozitif sayı alır ve toplamını yazar.</p>														
<p>5-)Aşağıdaki programın çıktısı nedir ?</p> <pre style="border: 1px solid black; padding: 5px;">x = 5 y = 5 x +=15 y = y + 15 print(str('x= ') + str(x)) print(str('y= ') + str(y))</pre> <p>a-)x=15 b-)x= 5 c-)x=20 d-)x=25 e-)x=30 y=20 y=20 y=20 y=25 y=30</p>	<p>6-)Aşağıdaki programın çıktısı nedir ?</p> <pre style="border: 1px solid black; padding: 5px;">x = 10 y = 20 z = 30 x = (y + z) + z * x print(x)</pre> <p>a-)250 b-)300 c-)350 d-)400 e-)375</p>														
<p>Tablo-1</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Fonksiyon</th> <th>Tanım</th> <th>Örnek</th> <th>Sonuç</th> </tr> </thead> <tbody> <tr> <td>Sum(list)</td> <td>Birkaç sayının toplam değerini döndürür.</td> <td>Sum(3,5,8)</td> <td>16</td> </tr> </tbody> </table> <p>Tablo-2</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Değişken</th> <th>Fonksiyon</th> <th>Çıktı</th> </tr> </thead> <tbody> <tr> <td>s=25 a=15 y=s+10</td> <td>Sum(s,a,y)</td> <td>?</td> </tr> </tbody> </table> <p>7-)Yukarıda Tablo-1'de istatistiksel bir fonksiyona örnek verilmiştir. Tablo-2'deki değişkenlerle fonksiyon çalıştırıldığında komut çıktı değeri ne olur? a-)25 b-)90 c-)50 d-)75 e-)80</p>	Fonksiyon	Tanım	Örnek	Sonuç	Sum(list)	Birkaç sayının toplam değerini döndürür.	Sum(3,5,8)	16	Değişken	Fonksiyon	Çıktı	s=25 a=15 y=s+10	Sum(s,a,y)	?	<pre style="border: 1px solid black; padding: 5px;">x = 2 y = 3 a = 6 b = 8 c = 10 print(c % y) print(y ** b)</pre> <p>8-) Yukarıdaki program için aşağıdakilerden hangisi doğrudur ? a-)c'nin y'ye göre yüzdesini alır ve yazdırır. b-)y'nin b ile çarpımını yazdırır. c-)c'nin y'ye göre modunu, y'nin b'ninci kuvvetini alır ve yazdırır. d-)y'nin c'ye göre modunu alır ve b ile çarpımını yazdırır. e-)c'nin b ile çarpımını sağlar.</p>
Fonksiyon	Tanım	Örnek	Sonuç												
Sum(list)	Birkaç sayının toplam değerini döndürür.	Sum(3,5,8)	16												
Değişken	Fonksiyon	Çıktı													
s=25 a=15 y=s+10	Sum(s,a,y)	?													

9-) Aşağıdaki programın çıktısı nedir ?

```
x=9
y=8
z=7
if (x > z) :
print (x)
elif(y < z) :
print (y)
elif (not (x==z)):
print (z)
```

a-) 7 b-)8 c-)9 d-)99 e-)Null

10-) Aşağıdaki programda yapılan hata hangi şıkta doğru olarak verilmiştir ?

```
sayi = float(input('Sayı Giriniz : '))
if sayi % 2 == 0:
print('Girilen sayı çifttir.')
if sayi ** 1 == 1:
print('Girilen sayı tektir.')
```

a-) sayi = float(input('Sayı Giriniz : '))---- **Veri tipi hatası**  
b-) if sayi % 2 == 0: ----- **Koşul yapısındaki hata**  
c-) print('Girilen sayı çifttir.')----- **Print komutundaki hata**  
d-) if sayi \*\* 1 == 1: ----- **Operatör hatası**  
e-) print('Girilen sayı tektir.')----- **Print komutundaki hata**

11-) Aşağıdaki programın çıktısı nedir ?

```
x = 10
y = 0
z = -10
if x > 0:
print('Pozitif')
if y == 0:
print('Sıfır')
if z < 0:
print('Negatif')
```

a-) Pozitif Negatif Sıfır Pozitif Negatif  
Negatif Pozitif Negatif Sıfır Sıfır  
Sıfır Sıfır Pozitif Negatif Pozitif

12-) Aşağıdaki programın işlevi nedir ?

```
if sayi >= 1 and sayi <= 7:
if sayi == 1:
print ("Pazartesi")
elif sayi == 2:
print ("Salı")
elif sayi == 3:
print ("Çarşamba")
elif sayi == 4:
print ("Perşembe")
elif sayi == 5:
print ("Cuma")
elif sayi == 6:
print ("Cumartesi")
elif sayi == 7:
print ("Pazar")
else:
print("Geçersiz bir değer girdiniz!")
```

a-) Haftanın günlerini ekrana yazdıran bir programdır.  
b-)1-7 arasındaki sayıları karıştıran ve günlerle eşleştiren bir programdır.  
c-)Girilen sayıya göre haftanın hangi günü olduğunu ekrana getiren programdır.  
d-)1-7 arasındaki sayıları ekrana yazdıran bir programdır.  
e-)1-7 arasındaki sayıları ve haftanın günlerini ekrana yazdıran programdır.

13-)Aşağıdaki programın işlevi nedir ?

```
a = 0
while a < 100:
    a += 1
    if a % 2 == 0:
        print(a)
```

- a-)Çıktı olarak “a=0” verir.  
b-)Çıktı olarak 2-100 arasındaki çift sayıları verir.  
c-) Çıktı olarak 2-100 arasındaki tek sayıları verir.  
d-)Sonsuz döngüye girer.  
e-)Çıktı olarak 1-100 arasındaki sayıları yazar.

14-) Aşağıdaki programın çıktısı nedir ?

```
sayilar = '123456789'
for k in sayilar:
    print(int(k) * 2)
```

a-) 2	b-) 1	c-) 9	d-) 18	e-) 3
4	2	8	16	6
6	3	7	14	9
8	4	6	12	12
10	5	5	10	15
12	6	4	8	18
14	7	3	6	21
16	8	2	4	24
18	9	1	2	27

15-)Aşağıdaki programın çıktısı nedir ?

```
for i in range (1,5,1):
    for y in range (1,3,1):
        print i,'*',y,'=',i*y
```

a-)1 * 1 =	b-)1 * 1 =	c-)4 *2= 8	d-)1 * 1 =	e-)1 * 1 =
1	1	4* 1 =4	1	1
1 * 2 = 2	2 * 2 = 4	3* 2 =6	1 * 2 = 2	2 * 2 = 4
2 * 1 = 2	3 * 1 = 3	3* 1 = 3	2 * 1 = 2	2 * 2 = 4
2 * 2 = 4	2 * 2 = 4	2 * 2 =	2 * 2 = 4	2 * 2 = 4
3 * 1 = 3	3 * 1 = 3	4	3 * 3 = 9	3 * 3 = 9
3 * 2 = 6	3 * 2 = 6	3 * 2 = 6	3 * 2 = 6	3 * 3 = 9
4 * 1 = 4	4 * 1 = 4	4 * 1 = 4	4 * 1 = 4	4 * 4 = 16
4 * 2 = 8	4 * 2 = 8	4 * 2 = 8	4 * 3 = 12	4 * 4 = 16

17-) Aşağıdaki programın çıktısı nedir ?

```
def kimlik(isim,soyisim,yas):
    print "Adınız :",isim,"Soyadınız"
    ,soyisim," Yaşınız :",yas

kimlik('Alper','Ünal','30')
```

- a-) Alper Ünal 30  
b-)Adınız : Soyadınız : Yas :  
c-)Adınız : Alper Soyadınız : Ünal Yas :30  
d-)’Alper’,’Ünal’,’30’  
e-)kimlik(isim,soyisim,yas)

16-)Aşağıdaki programda “break” komutu ne iş yapar ?

```
kullanici_adi = 'kullanici'
parola = 'parola'
while True:
    sor1 = input('Kullanıcı Adınızı Girin :')
    sor2 = input('Şifrenizi Girin :')
    if sor1 == 'kullanici' and sor2 == 'parola':
        print('Kullanıcı adı ve parolanız
onaylandı.Hoşgeldiniz !!!')
        break
    else:
        print('Kullanıcı adı yada parolanız yanlış
!!!')
        print('Lütfen tekrar deneyin.')
```

- a-)Programı sonlandırmaya yarar.  
b-)Döngüyü başa döndürür.  
c-)”True” komutu ile beraber kullanılır ve programı başa döndürür.  
d-)”if” koşulunu sona erdirir.  
e-)”else” komutunu başlatır.

18-)Aşağıdaki programın çıktısı nedir ?

```
def isim(x):
    a = x + str('Merhaba ')
    b = x + str('Naber ')
    c = x + str('Nasılsın')
    return a + (b + c)
print(isim('Ali '))
```

- a-) Merhaba Merhaba Merhaba  
b-)Nasılsın Nasılsın Nasılsın  
c-)Naber Naber Naber  
d-)Ali Merhaba Ali Naber Ali Nasılsın  
e-)Merhaba Naber Nasılsın





25-)Aşağıdaki programın çıktısı nedir ?

```
dizi = [10, 20, 30, 40, 50]  
print len(dizi)  
print dizi.index(30)
```

a-)3      b-)4      c-)5      d-)4      e-)5  
1          2          2          3          4



## Ek-5. Görüşme Formu

### PROGRAMLAMA ÖĞRETİMİ GÖRÜŞME FORMU

**Görüşülen Kişi :**

**Tarih :**

**Görüşme Başlangıç Saati :**

**Görüşme Bitiş Saati :**

**Öğrenci :**

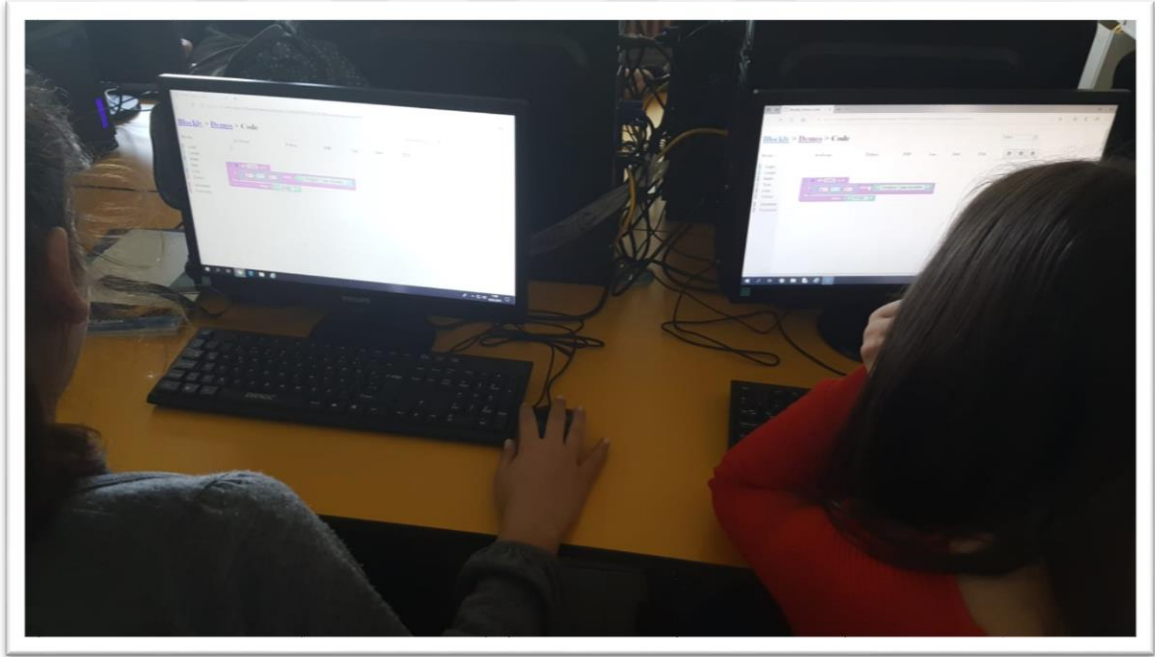
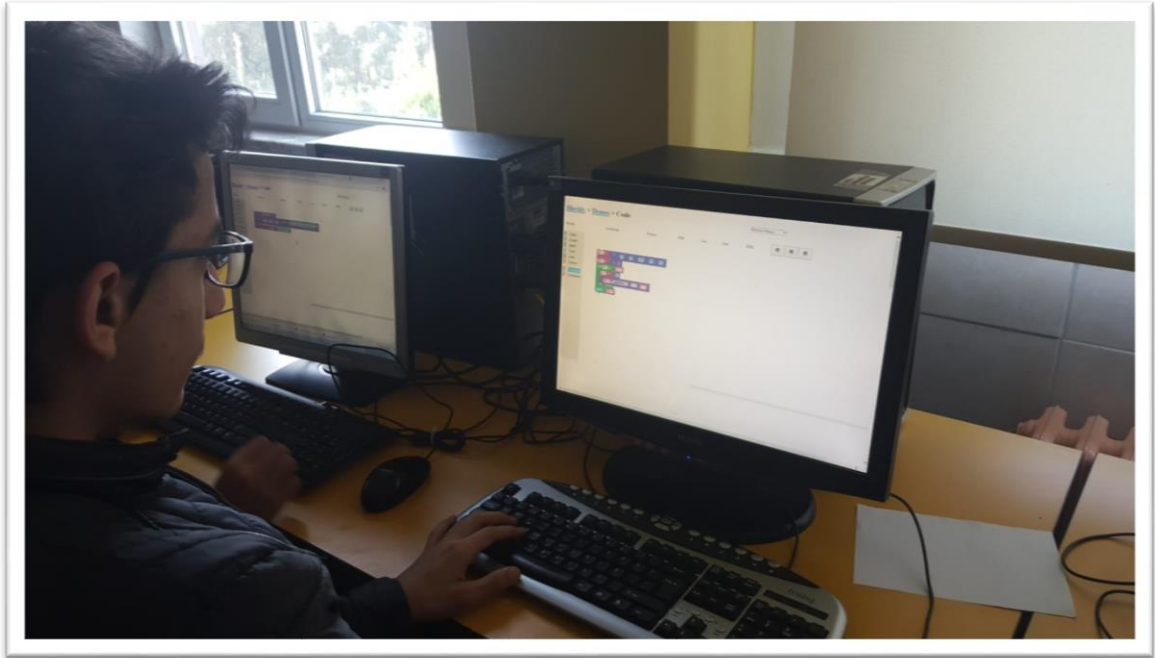
Merhaba,

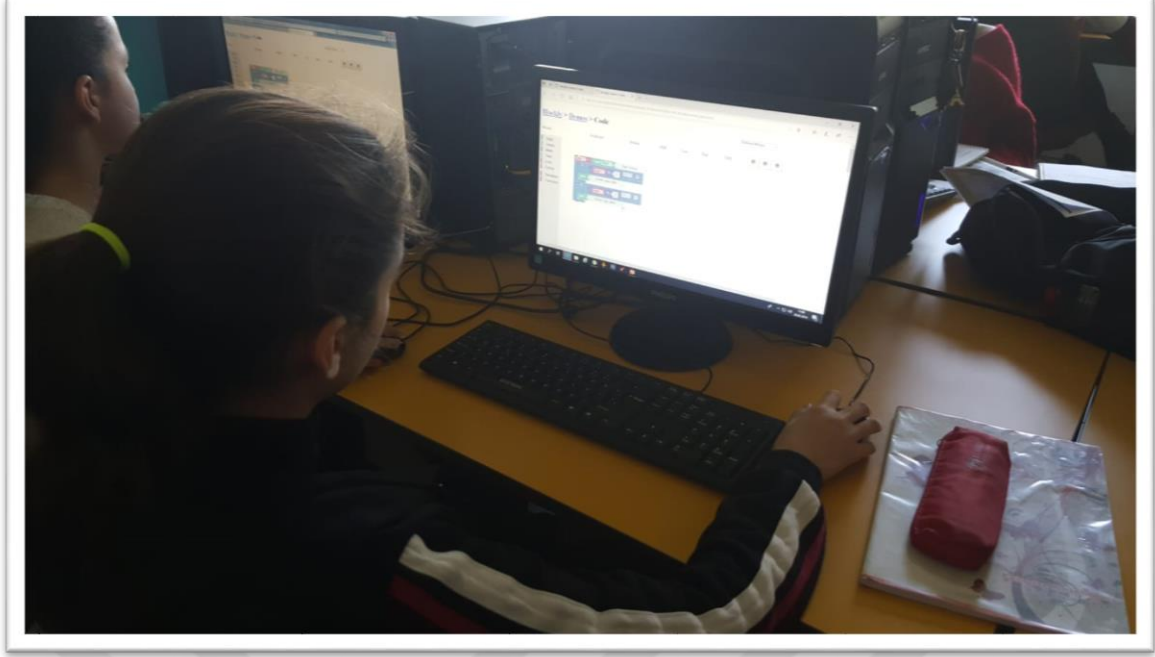
Atatürk Üniversitesi Bilgisayar ve Öğretim Teknolojileri bölümünde yüksek lisans yapmaktayım. "Sosyal Bilimler Lisesi öğrencilerine blok tabanlı programlama öğretiminin kaygı, bilişsel yük ve başarıya etkisi" konulu araştırmamla ilgili görüşme yapmayı kabul ettiğiniz için teşekkür ederim. Yapacağımız görüşme yalnızca araştırma amaçlı olup sonucunda elde edilecek dokümanlarda isminiz hiçbir şekilde geçmeyecektir. Araştırma sonucu isterseniz sizlerle paylaşılacaktır.

Eğer sizin için de bir sakıncası yoksa görüşmeyi kayıt altına almak istiyorum.

- 1- Programlama dersinin verilmesi hakkındaki düşüncelerin nelerdir?
- 2- Programlama öğretimi yapılacağını öğrendiğinde bu seni endişelendirdi mi? neden?
- 3- Öğretmenin bir program kodu yazmanı istediğinde seni endişelendirir mi? neden?
- 4- Sınıfta öğretmenin bir program kodu yazmanı istediğinde hata yaptığında öğretmenin ve arkadaşlarının vereceği tepkilerden çekiniyor musun?
  - a. Evet ise neden?
  - b. Hayırsa neden?
- 5- Programlama öğretimi süresince zorlandığını düşünüyor musun? Neden?
- 6- Program kodundaki hataları düzeltmek zorunda kaldığında ne hissediyorsun?
- 7- Öğrenme aracı Python/Blockly programlama aracını kullanırken zorlandın mı? neden?
- 8- Derste yapılan programlama örnekleri için ne kadar çaba sarf ettiniz? Neden?
- 9- En çok hangi konuda zorlandın? Neden?
- 10- Bu uygulama sonunda hangi kazanımları elde ettiğini düşünüyorsun? Neler öğrendin?
- 11- Bu ders seçmeli bir ders olsaydı yine de almak ister miydin?

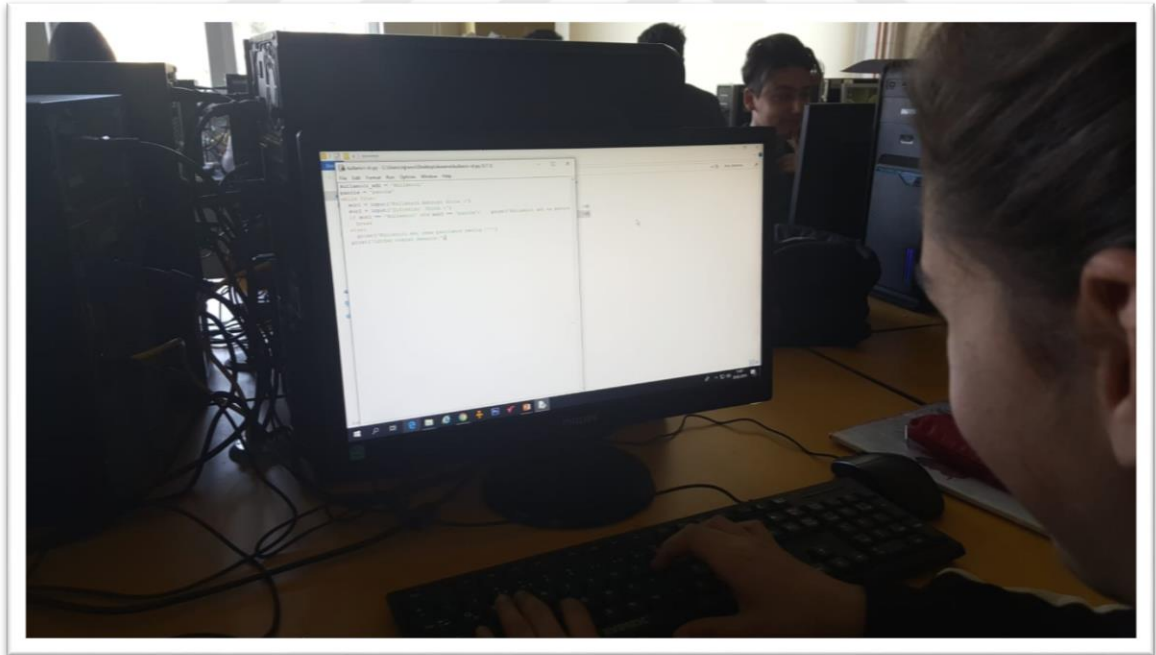
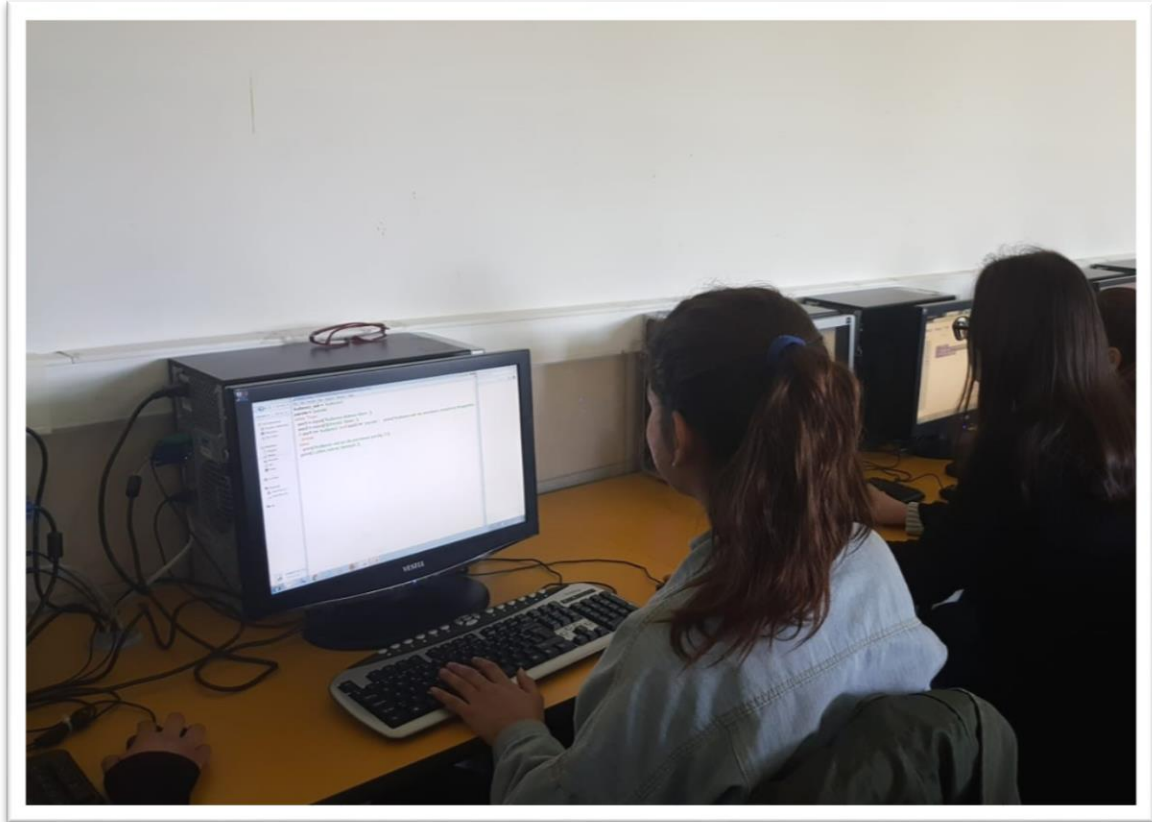
## Ek-6. Blockly Tabanlı Programlama Grubu Kamera Görüntüleri





**Ek-7. Python Tabanlı Programlama Grubu Kamera Görüntüleri**





## Ek-8. Blockly Grubu ve Python Grubu Programlama Etkinlik İçerikleri

### Ünite -1: İfadeler ve Aritmetik İşlemler

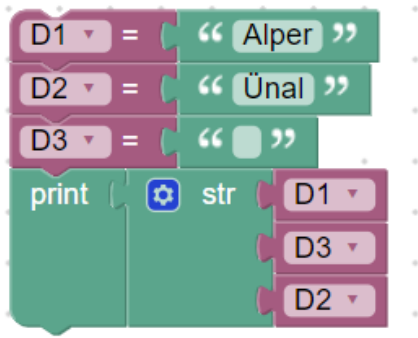
#### Uygulama haftası: 1. ve 2. Hafta

#### Ünite Kazanımları

- Belirlenen programlama dilinde değişkenleri ve sabitleri uygun şekilde kullanarak programlar geliştirir.
- Belirlenen programlama dilinde girdilerin belirlendiği ve çıktılarının gözlemlendiği programlar geliştirir.

**SORU:** 3 değişken kullanarak metin türünde adınızı ve soyadınızı çıktı olarak veren bir program yazınız.

#### Blockly Ekranı



#### Python Kod Editör Ekranı

```
main.py
1 D1 = "Alper"
2 D2 = "Ünal"
3 D3 = ""
4 print D1+D3+D2
5
```

#### Ekran Çıktısı:

Bu sayfanın mesajı  
Alper Ünal

Tamam

```
Python 3.6.1 (default, Dec 2015, 13:05:11)
[GCC 4.8.2] on linux
Alper Ünal
```

**SORU:** Aşağıdaki programın çıktısını yazınız.

#### Blockly Ekranı



#### Python Kod Editör Ekranı

```
main.py saved
1 print(1.49)
2 print(round(1.49))
3 print(round(1.99))
4
5
```

#### Ekran Çıktısı:

1.49

1

2

```
Python 3.6.1 (default, Dec 2015, 13:05:11)
[GCC 4.8.2] on linux
```

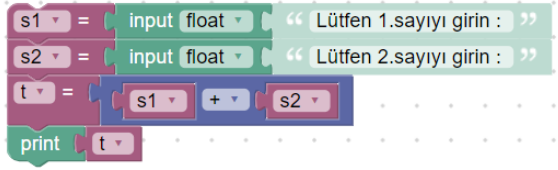
```
1.49
```

```
1
```

```
2
```

**SORU:** Kullanıcıdan sayı tipinde iki adet değişken alan ve bu değişkenlerin toplamını ekrana yazdıran program kodunu oluşturunuz.

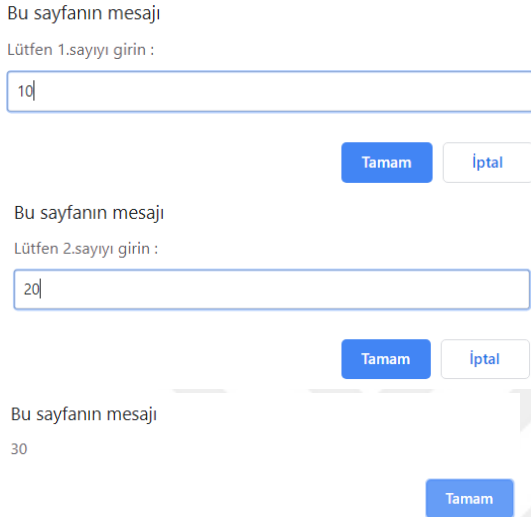
### Blockly Ekranı



### Python Kod Editör Ekranı

```
main.py saved
1 s1=float(input("Lütfen 1.sayıyı giriniz : "))
2 s2=float(input("Lütfen 2.sayıyı giriniz : "))
3 t=s1+s2
4 print (t)
```

### Ekran Çıktısı:



```
Python 3.6.1 (default, Dec 2015, 13:05:11)
[GCC 4.8.2] on linux
Lütfen 1.sayıyı giriniz : 10
Lütfen 2.sayıyı giriniz : 20
30.0
```

## Ünite -2:Koşullu İfadeler

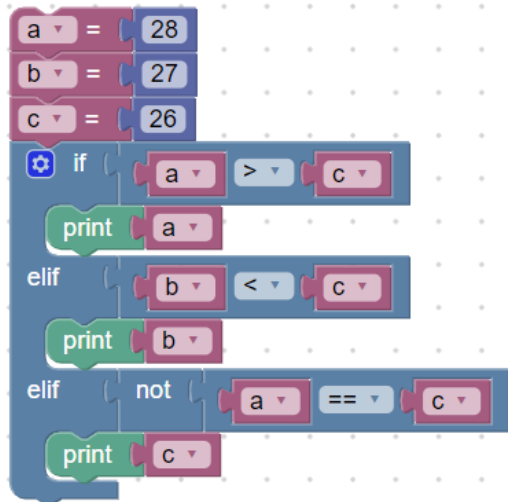
### Uygulama haftası: 3. ve 4. Hafta

#### Ünite Kazanımları

- Belirlenen programlama dilinde kontrol yapılarını kullanarak programlar geliştirir.

**SORU:** 26,27,28 sayılarını değer olarak kıyaslayan bir program yazınız.

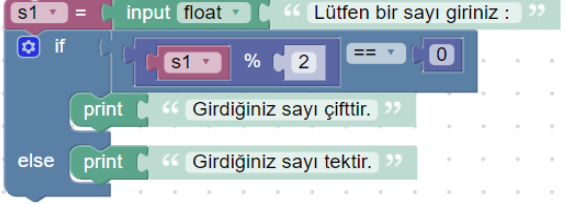
### Blockly Ekranı



### Python Kod Editör Ekranı

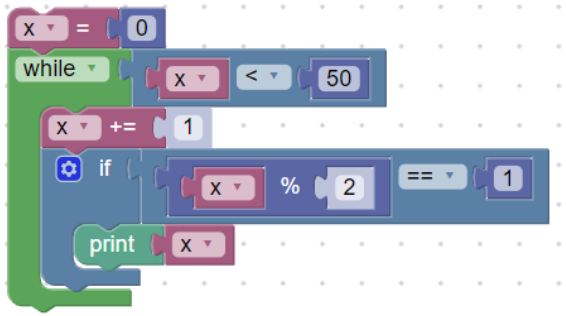
```
main.py saved
1 x=28
2 y=27
3 z=26
4 if x>z:
5 | print(x)
6 elif y<z:
7 | print(y)
8 elif not x==z:
9 | print(z)
10
```



<b>Ekran Çıktısı:</b>	
<p>Bu sayfanın mesajı</p> <p>28</p> <p>Tamam</p>	<pre>Python 3.6.1 (default, Dec 2015, 13:05:11) [GCC 4.8.2] on linux 28 &gt;</pre>
<b>SORU: Klavyeden girilen bir sayının tek mi çift mi olduğunu söyleyen bir program kodu oluşturunuz.</b>	
<b>Blockly Ekranı</b>	<b>Python Kod Editör Ekranı</b>
	<pre>main.py saved 1 s1=float(input("Lütfen bir sayı giriniz : ")) 2 if (s1 % 2)==0 : 3     print ("Girdiğinizi sayı çifttir.") 4 else: 5     print ("Girdiğinizi sayı tektir.")</pre>
<b>Ekran Çıktısı:</b>	
<p>Lütfen bir sayı giriniz :</p> <input type="text" value="111"/> <p>Tamam İptal</p> <p>Girdiğiniz sayı tektir.</p> <p>Tamam</p>	<pre>Python 3.6.1 (default, Dec 2015, 13:05:11) [GCC 4.8.2] on linux Lütfen bir sayı giriniz : 111 Girdiğinizi sayı tektir. &gt;</pre>

### Ünite -3:Döngüler

Uygulama haftası: 5. ve 6. Hafta

<b>Ünite Kazanımları</b>	
<ul style="list-style-type: none"> <li>Belirlenen programlama dilinde tekrarlı yapıları kullanarak programlar geliştirir.</li> </ul>	
<b>SORU: 0-50 arasındaki tek sayıları yazdıran bir program kodu yazınız.</b>	
<b>Blockly Ekranı</b>	<b>Python Kod Editör Ekranı</b>
	<pre>main.py saved 1 x=0 2 while x &lt; 50 : 3     x+=1 4     if (x%2)!=0: 5         print (x) 6</pre>

## Ekran Çıktısı:

1		Python 3.6.1 (default, Dec 2015, 13:05:11) [GCC 4.8.2] on linux
3	Tamam	1
5	Tamam	3
	Tamam	5
	Tamam	7
	Tamam	9
	Tamam	11
	Tamam	13
	Tamam	15
	Tamam	17
	Tamam	19
	Tamam	21
	Tamam	23
	Tamam	25
	Tamam	27
	Tamam	29
	Tamam	31
	Tamam	33
	Tamam	35
	Tamam	37
	Tamam	39
	Tamam	41
	Tamam	43
	Tamam	45
	Tamam	47
	Tamam	49

**SORU :** Verilen program kodunun ekran çıktısını yazınız.

## Blockly Ekranı



## Python Kod Editör Ekranı

```
main.py saved
1 sayilar = '2468'
2 for k in sayilar:
3     print(int(k) * 5)
4
```

## Ekran Çıktısı:

Bu sayfanın mesajı		Python 3.6.1 (default, Dec 2015, 13:05:11) [GCC 4.8.2] on linux
10	Tamam	10
Bu sayfanın mesajı		20
20	Tamam	30
Bu sayfanın mesajı		40
30	Tamam	
Bu sayfanın mesajı		
40	Tamam	

## Ünite -4:Fonksiyonlar

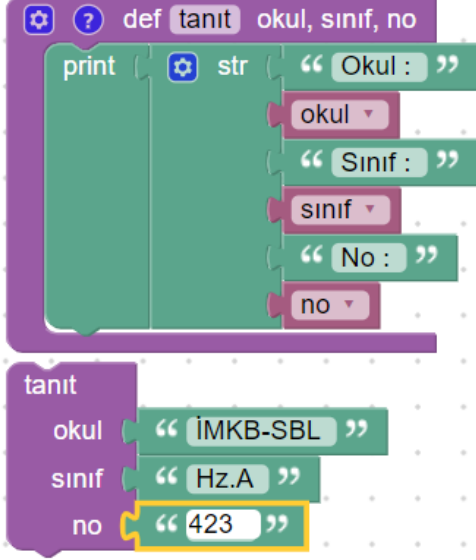
Uygulama haftası: 7. Ve 8. Hafta

### Ünite Kazanımları

- Belirlenen programlama dilinde parametre almayan fonksiyon içeren programlar geliştirir.
- Belirlenen programlama dilinde değer döndüren fonksiyon içeren programlar geliştirir.

**SORU:** Okul adı, okul numarası ve sınıf bilgisini ekrana yazdıran bir fonksiyon kodu yazınız.

### Blockly Ekranı



### Python Kod Editör Ekranı

```
main.py saved
1 def tanıt(okul,sınıf,no):
2     print ('Okul :',okul,'Sınıf',sınıf,'No :',no)
3     tanıt('İMKB-SBL','Hz.A','423')
```

### Ekran Çıktısı:

Bu sayfanın mesajı

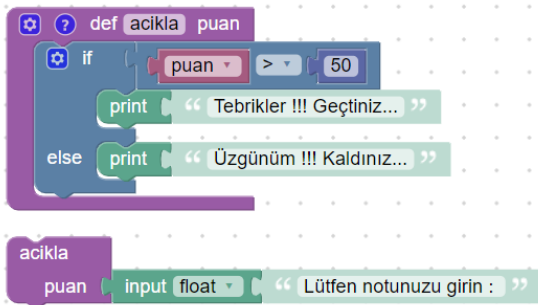
Okul : İMKB-SBL Sınıf : Hz.A No : 423

Tamam

```
Python 3.6.1 (default, Dec 2015, 13:05:11)
[GCC 4.8.2] on linux
Okul : İMKB-SBL Sınıf Hz.A No : 423
```

**SORU:** Kullanıcıdan alınan notun 50 ve üzerinde olması durumuna göre öğrencinin geçip kaldığını hesaplayan bir fonksiyon yazınız.

### Blockly Ekranı



### Python Kod Editör Ekranı

```
main.py saved
1 def acikla(puan):
2     if puan >= 50:
3         print('Tebrikler ! Dersten geçtiniz...')
4     else :
5         print('Üzgünüm ! Dersten kaldınız...')
6     acikla(puan=float(input('Lütfen notunuzu girin : ')))
```

### Ekran Çıktısı:

Bu sayfanın mesajı

Lütfen notunuzu girin :

49

Tamam

İptal

```
Python 3.6.1 (default, Dec 2015, 13:05:11)
[GCC 4.8.2] on linux
Lütfen notunuzu girin : 49
Üzgünüm ! Dersten kaldınız...
```

## Ünite -5:Diziler (Listeler)

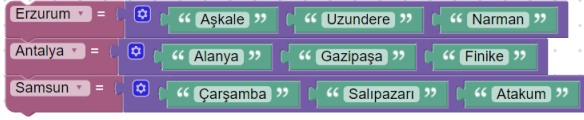
Uygulama haftası: 9. ve 10. Hafta

### Ünite Kazanımları

- Belirlenen programlama dilinde dizi tipinde veri tanımlar.
- Tanımladığı dizi tipindeki veriye ait temel fonksiyonların yer aldığı programları geliştirir.

**SORU:** Son indeksi 2 olan 3 adet dizi tanımlayın.

### Blockly Ekranı

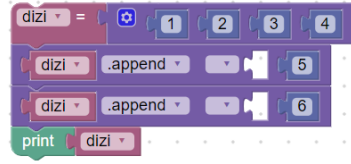


### Python Kod Editör Ekranı

```
main.py saved
1 Erzurum = [Aşkale,Uzundere,Narman]
2 Antalya = [Alanya,Gazipaşa,Finike]
3 Samsun=[Çarşamba,Salıpazarı,Atakum]
```

**SORU:** dizi = [1,2,3,4] dizisine 5 ve 6 sayılarını eleman olarak ekleyen ve ekrana yazdıran bir program kodu yazınız.

### Blockly Ekranı



### Python Kod Editör Ekranı

```
main.py saved
1 dizi=[1,2,3,4]
2 dizi.append(5)
3 dizi.append(6)
4 print (dizi)
```

### Ekran Çıktısı:

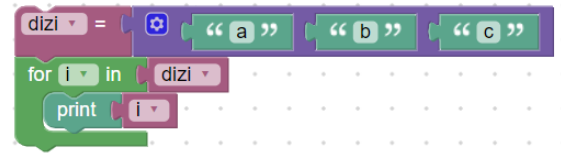
Bu sayfanın mesajı  
1,2,3,4,5,6

Tamam

```
Python 3.6.1 (default, Dec 2015, 13:05:11)
[GCC 4.8.2] on linux
[1, 2, 3, 4, 5, 6]
```

**SORU:** Bir dizi oluşturun ve oluşturduğunuz bu dizinin elemanlarını bir döngüye atayın ve elemanların tek tek çıktısını sağlayın

### Blockly Ekranı



### Python Kod Editör Ekranı

```
main.py saved
1 dizi=['a','b','c']
2 for i in dizi:
3     print(i)
4
```

### Ekran Çıktısı:

Bu sayfanın mesajı

a

Tamam

Bu sayfanın mesajı

b

Tamam

Bu sayfanın mesajı

c

Tamam

```
Python 3.6.1 (default, Dec 2015, 13:05:11)
[GCC 4.8.2] on linux
```

```
a
b
c
```

## Ek-9. Blok Tabanlı Programlama Araçlarıyla İlgili Araştırmalar

Yazar	Amaç	Programlama Aracı	Örneklem Düzeyi ve Sayısı	Değişkenler	Yöntem	Sonuç
Çetin (2012)	Bilgisayar programlama eğitiminin çocukların problem çözme becerileri üzerine etkisini incelemek	Logo	17 ortaokul öğrencisi	Problem çözme becerisi, öğrenci görüşleri	Karma	Öğrencilerin problem çözme becerileri üzerinde olumlu etki yaptığı, programlama eğitiminden memnun kaldıkları, mesleki açısından bilgi ve becerilerine olumlu yansıdığı görülmüştür.
Genç ve Karakuş (2012)	Scratch kullanımı hakkında öğrencilerin başarısını, deneyimlerini ve görüşlerini belirlemek	Scratch	109 üniversite öğrencisi	Öğrenci başarısı, deneyimleri ve görüşleri	Karma	Scratch blok tabanlı programlama ortamını kullanırken başarılı oldukları ve Scratch aracına karşı olumlu görüş bildirdikleri görülmüştür.
Dzhenzher (2014)	Scratch kullanımının bilgisayar simülasyon becerisine etkisini incelemek	Scratch	300 ortaokul öğrencisi	Simülasyon yapma becerisi	Nicel	Scratch'in programlamaya başlamak için iyi bir araç olduğu, sadece eğlenceli bir ortam değil aynı zamanda programlama üzerine ciddi projelerde de kullanılabileceği, modelleme ve yaratıcı çalışma için uygun bir ortam olduğu belirlenmiştir.
Feng ve Chen (2014)	Scratch kullanarak öğrencilerin programlama performansını ve öz-düzenleme becerilerini incelemek	Scratch	232 ortaokul öğrencisi	Programlama performansı ve öz-düzenleme becerisi	Nicel	Öğrencilerin öz-düzenleme becerilerini geliştirebildikleri ve öğrenme başarılarını arttırdıkları sonucu ortaya çıkmıştır.
Gülbahar ve Kalelioğlu (2014)	Scratch kullanımının ilkökul öğrencilerinin problem çözme becerilerine etkisini ve programlama hakkındaki görüşlerini incelemek	Scratch	49 ilkökul öğrencisi	Problem çözme becerisi ve uygulamaya yönelik	Karma	Öğrencilerinin problem çözme becerilerinde anlamlı bir artış görülmediği fakat tüm öğrencilerin programlamayı sevdiği ve ilerde kendilerini geliştirmek istedikleri ortaya çıkmıştır.
Solmaz (2014)	Alice programlama aracının öğrencilerin eleştirel düşünme eğilimleri, problem çözme becerileri, üst bilişsel farkındalık düzeyleri ve ders başarıları üzerindeki etkisini ve ortama ilişkin öğrenci görüşlerini incelemek	Alice	39 üniversite öğrencisi	Başarı, eleştirel düşünme, problem çözme becerisi, üst bilişsel farkındalık, öğrenci görüşleri	Karma	Alice programının bu değişkenler üzerinde anlamlı bir etki yaratmadığı sadece eleştirel düşünme becerileri üzerinde etkili olduğu ve öğrencilerin uygulamayı eğlenceli bulduğu belirlenmiştir.
Demir (2015)	Programlama öğretiminde eğitsel programlama dilinin farklı kullanımlarının programlama başarısı ve kayısına etkisini incelemek	Scratch	87 üniversite öğrencisi	Programlama başarısı ve kayısı	Nicel	Blok tabanlı programlama ortamının dersin teori ve uygulama kısmına entegre edilmesinin öğrencilerin programlama başarısını olumlu yönde etkilediği ve programlama kaygılarını azalttığı belirlenmiştir.
Doğan ve Kert (2015)	Ortaokul öğrencilerinde bilgisayar oyunu geliştirme sürecinin öğrencilerin eleştirel düşünme becerilerine ve algoritma başarılarına etkisini incelemek	Kodu Game Lab	54 ortaokul öğrencisi	Eleştirel düşünme becerisi, algoritma başarısı ve öğrenci görüşleri	Karma	Kodu Game Lab ile oyun tasarımı şeklinde işlenen algoritma eğitimi, geleneksel şekilde işlenen duruma göre öğrencilerin eleştirel düşünme becerilerini ve algoritma başarılarını olumlu etkilemiştir. Çalışma sonunda öğrenciler oyun tasarımı uygulaması ile bilgisayara olan ilgilerinin arttığını ifade etmişlerdir.
Erol & Kurt (2017)	Scratch ile programlama öğretiminin bilişim teknolojileri öğretmen adaylarının motivasyon ve başarılarına etkisini incelemek	Scratch	52 üniversite öğrencisi	Motivasyon ve programlama başarısı	Karma	Scratch kullanan grubun motivasyon puanının diğer gruba göre yüksek olduğu, programlama başarı puanında ise her iki grubun da ilk duruma göre anlamlı fark oluşturduğu, öğrencilerin ortamı eğlenceli ve motivasyon artırıcı bulduğu fakat programlama öğretiminde bazı konularda eksik kaldığı ortaya çıkmıştır.
Koorsse, Cilliers ve Calitz (2015)	Ortaokul öğrencileri için blok tabanlı programlama araçlarının kullanımının öğrencilerin programlama kavramlarını anlamalarına ve motivasyonlarına etkisini incelemek	B#, Scratch, Robomind	27 lise öğrencisi	Programlama kavramlarını anlama üzerindeki etkisi	Nicel	Lise öğrencileri için incelenen 3 blok tabanlı programlama aracı içinde farklı konularda farklı araçları kavramları anlama konusunda etkili gördükleri ortaya çıkmıştır.
Kayabaşı (2016)	Alice blok tabanlı programlama aracıyla programlama eğitimi alan öğretmen adaylarının öz-yeterlik algıları ve programlama hakkındaki düşüncelerinin incelenmesi	Alice	24 üniversite öğrencisi	Öz-yeterlik algısı ve uygulamaya yönelik öğrenci görüşleri	Karma	Çalışma sonucunda öğretmen adaylarının öz-yeterlik algılarında anlamlı düzeyde bir farklılık olduğu, Alice ortamının eğlenerek ve uygulayarak öğrenme imkânı sunmasının olumlu olduğu, 21. Yüzyıl becerilerini aktarmak açısından gerekli bir uygulama olduğu, üç boyutlu tasarımların gelişmesinde katkı yapacağı ve öğrenen kitlenin programlama derslerine olan ilgi ve katılımı artıracağı yönünde görüşler ifade etmişlerdir.

Yazar	Amaç	Programlama Aracı	Örneklem Düzeyi ve Sayısı	Değişkenler	Yöntem	Sonuç
Sáez-López, Román-González ve Vázquez-Cano (2016)	Scratch kullanımının ortaokul öğrencilerinin programlamaya karşı tutumlarını, motivasyonlarını ve algılanan yararlığa karşı etkilerini incelemek	Scratch	107 ortaokul öğrencisi	Programlamaya karşı tutum, motivasyon, algılanan yararlık	Karma	Blok tabanlı programlama ortamının öğrencilerin tutum ve motivasyonunu olumlu etkilediği, algılanan yararlık ortalama puan değerlerinin yüksek çıktığı, eğlenceli bir içerik sunduğu ortaya çıkmıştır.
Yiğit (2016)	Görsel programlama ortamının öğrencilerin bilgisayar programlamayı öğrenmelerine ve programlamaya karşı tutumlarına etkisini incelemek	Blockly	42 üniversite öğrencisi	Programlamaya karşı tutum ve erişimi	Nicel	Çalışma sonunda deney grubu öğrencilerinin kontrol grubu öğrencilerine göre programlama başarılarının daha yüksek olduğu, programlamaya karşı tutum puanları arasında ise anlamlı bir farkın ortaya çıkmadığı tespit edilmiştir.
Zhong, Wang, ve Chen (2016).	İkili gruplar halinde oluşturulan grupların demografik özelliklerinin programlama başarısına etkisini incelemek	Alice	154 ortaokul öğrencisi	Programlama başarısı	Nicel	Erkek grup, kız grup ve kız-erkek grupların arasında programlama başarısı açısından anlamlı bir farkın olmadığı ortaya çıkmıştır. Kız grupların çalışma sonunda daha üretken oldukları ve tüm gruplar arasında arkadaşlık ilişkilerinin geliştiği gözlenmiştir.
Boldbaatar (2017)	Öğrencilerin daha önce edinmiş oldukları programlama deneyimlerini yeni bir öğrenim ortamına nasıl aktardıklarını görmek	Starlogo Tng	21 üniversite öğrencisi	Programlama becerisi	Nicel	Metin tabanlı programlama deneyimi olan öğrencilerin daha yüksek final notları aldıkları, tamamlamalı örneğin tecrübeli öğrenciler açısından daha etkili olduğu, tam uygulamalı örneklerin diğer örneklere göre öğrencileri daha çok zorladığı belirlenmiştir.
Erol (2017)	Scratch ile yapılan programlama ve algoritma öğretiminin öğrencilerin programlama başarılarına ve motivasyonlarına etkisini incelemek	Scratch	52 üniversite öğrencisi	Motivasyon ve programlama başarısı	Karma	Deney ve kontrol grubu öğrencilerinin programlama başarısı puanlarının arttığı, motivasyon puanlarında ise deney grubu lehine anlamlı bir artışın olduğu fakat geleneksel yöntemin kullanıldığı kontrol grubunda motivasyon puanlarında bir azalma olduğu belirlenmiştir.
Kasalak (2017)	Robotik kodlama etkinliklerinin ortaokul öğrencilerinin kodlamaya ilişkin öz-yeterlik algılarına etkisini ve etkinliklere ilişkin öğrenci yaşantılarını incelemek	Scratch	329 ortaokul öğrencisi	Öz-yeterlik algısı ve öğrenci yaşantıları	Karma	Hem basit hem karmaşık blok tabanlı programlama eğitime yönelik öz-yeterlik algılarında her grup için son test lehine anlamlı bir fark olduğu, öğrencilerin etkinlikleri ilgi çekici ve eğlenceli buldukları, kişisel gelişimlerine olumlu katkı sağladığını düşündükleri ve etkinliklere katılmaya istekli oldukları yönünde fikir ortaya koymuşlardır.
Saygıner (2017)	Blok tabanlı görsel ve metin tabanlı programlama öğretimlerinin erişimi, mantıksal düşünme ve motivasyona etkilerini araştırmak	Scratch	60 üniversite öğrencisi	Erişimi, mantıksal düşünme becerisi, motivasyon	Nicel	Deney grubu öğrencilerinin erişimi, motivasyon ve mantıksal düşünme puanlarında artış gözlemlendiği, kontrol grubunda öğrencilerin mantıksal düşünme ve erişimi seviyelerinde artış gözlenirken motivasyon seviyelerinde anlamlı bir artış olmadığı ortaya çıkmıştır.
Yıldırım (2017)	Scratch programlama aracı eğitiminin mobil uygulama ortamı kullanılarak daha verimli bir sonucun sağlanması	Scratch	94 ortaokul öğrencisi	Kodlama becerisi	Nicel	Mobil uygulama üzerinden yapılan etkinliklerin öğrencilerin kodlama becerisi üzerinde olumlu bir etkiye sahip olduğu ortaya çıkmıştır.
Yükseltürk ve Altok (2017)	Scratch ile programlamanın öğretmen adaylarının öz-yeterlik algıları ve bilgisayara karşı tutumları üzerindeki etkilerini incelemek	Scratch	151 üniversite öğrencisi	Öz-yeterlik algısı ve bilgisayara karşı tutum	Nicel	Verilen tüm karmaşık görevlere rağmen öğrencilerin öz-yeterlik algılarında anlamlı düzeyde artış olduğu, bilgisayara karşı tutumlarında ise önemli ölçüde olumlu etkiler görüldüğü ortaya çıkmıştır.
Dinçer (2018)	Scratch ve Kodu Game Lab ortamlarının öğrencilerin öz-yeterlik algısı, programlama başarısı ve tutumuna etkisini incelemek	Scratch, Kodu Game Lab	27 ortaokul öğrencisi	Öz-yeterlik algısı, programlama başarısı ve tutum	Karma	Scratch ile programlama alan öğrencilerin Kodu Game Lab ile programlama alan öğrencilere göre başarı anlamında önde oldukları, gruplar arasında tutum ve öz-yeterlik algıları açısından anlamlı bir farkın olmadığı, tutum ve öz-yeterlik algılarının arttığı görülmüştür.
Erdem (2018)	Yüz yüze eğitim ve ters yüz sınıf modeli teknoloji destekli öğrenme ortamlarında Scratch kullanımının öğrencilerin bilgi işlemsel düşünme becerisine etkisini tespit etmek	Scratch	79 ortaokul öğrencisi	Bilgi işlemsel düşünme becerisi	Karma	Öğrencilerin bilgi işlemsel düşünme becerileri puanları arasında gruplar arası anlamlı olmayan bir fark oluşmuştur. Ayrıca grup içi öz-yeterlik puan ortalamalarında anlamlı bir farklılık olduğu belirlenmiştir.

Yazar	Amaç	Programlama Aracı	Örneklem Düzeyi ve Sayısı	Değişkenler	Yöntem	Sonuç
Erkoç (2018)	İşbirlikli oyun tasarım modellerinin öğrencilerin eleştirel düşünme, problem çözme ve akademik başarılarına etkisini tespit etmek	Kodu Game Lab	48 adet ortaokul öğrencisi	Eleştirel düşünme, problem çözme ve akademik başarı	Nicel	Süreç sonucunda öğrencilerin eleştirel düşünme, problem çözme ve akademik başarı puanları incelenmiş, eleştirel düşünme becerisi açısından herhangi bir grup lehine anlamlı bir fark olmadığı, problem çözme becerisi açısından işbirlikli oyun geliştirme kullanılan grup lehine anlamlı bir fark tespit edilmiştir. Akademik başarı puanlarında ise deney grubu lehine anlamlı bir fark tespit edilmiştir.
Kaya (2018)	Alice görsel programlama ortamının öğrencilerin akademik başarı, problem çözme beceri algıları, motivasyonları ve programlama hazır bulunuşluklarına etkisini incelemek	Alice	63 önlisans öğrencisi	Akademik başarı, problem çözme becerisi algıları, motivasyon ve programlama hazır bulunuşlukları	Karma	Araştırma sonucunda öğrencilerin akademik başarıları, problem çözme becerisi algıları ve motivasyon düzeylerinde deney grubu lehine anlamlı bir farklılık bulunamamış olup Alice aracını kullanan grup lehine programlama hazır bulunuşluk düzeyleri arasında anlamlı bir farklılık tespit edilmiştir.
Marcelino, Pessoa, Vieira, Salvador ve Mendes (2018)	Bilgisayarca düşünme becerilerinin olumlu yönde gelişimine Scratch ve Alice ortamlarının katkısı olup olmadığını belirlemek	Scratch	9 ilköğretim okulu öğretmeni	Bilgisayarca düşünme becerisi	Nicel	Scratch görsel programlama ortamının kursiyerlerin bilgisayarca düşünme becerilerini geliştirdiği belirlenmiştir.
Perez-Marin, Hijon-Neira, Bacelo ve Pizarro (2018)	Çeşitli metaforlar ve Scratch eşliğinde çocuklara bilgisayarca düşünme becerilerinin gelişimini takip etmek	Scratch	132 ilköğretim öğrencisi	Programlama başarısı	Nicel	Araştırmacıların, programlama öğretimi için çeşitli metaforlar geliştirmişler ve bu metaforlar eşliğinde Scratch aracı kullanılarak yapılan programlama eğitiminin öğrencilerin başarısına olumlu etki ettiği ve öğrencilerin bilgisayar kavramlarını kolay bir şekilde öğrendikleri sonucuna ulaşılmıştır.
Topallı ve Çağıltay (2018)	Scratch kullanılarak yapılan ve günlük problemlerin oluşturduğu programlama öğretim içeriğinin öğrencilerin genel mühendislik programı, bitirme projesi ve programlamaya giriş seviyesindeki başarılarına etkisini incelemek	Scratch	395 mühendislik öğrencisi	Programlama performansı	Nicel	Çalışmalarında hayatta karşılaşılan problemlere dayalı olarak geliştirilen oyunlarla, programlama derslerine girişin kolaylaştırılması ve programlama becerileri ile öğrenci motivasyonlarının potansiyel olarak artırılmasını amaçlamışlardır. Çalışma sonucunda öğrencilerin programlama performanslarının çalışma süresi sonunda ciddi oranda arttığı gözlenmiştir.
Vatansever (2018)	Scratch programlama ortamının öğrencilerin problem çözme becerilerine etkisini incelemek	Scratch	226 ortaokul öğrencisi	Problem çözme becerisi	Karma	Çalışmada Scratch'in öğrencilerin problem çözme becerileri üzerinde anlamlı bir fark oluşturduğu, problem çözme becerilerinde cinsiyet ve sınıf seviyesinin etkisinin olmadığı belirlenmiştir. Ayrıca araştırmacı tüm hedeflerin gerçekleşmesi için tasarım odaklı yaklaşım uygulanması önerisini getirmiştir.
Zamin, Ab Rahim, Savita, Bhattacharyya, Zaffar ve Jamil (2018)	Scratch blok tabanlı programlama öğretimi üzerine öğrenci yaklaşımlarını tespit etmek	Scratch	46 ilkokul öğrencisi	Öğrenci görüşleri	Nitel	Avustralya ve Malezya olmak üzere iki ayrı ülkede öğrencilerin kendilerine verilen farklı görevleri hiçbir tecrübeleri olmadığı halde başarıyla yerine getirdikleri belirlenmiştir.
Dohn (2019)	Scratch kodlamanın öğrencilerin kodlama ve matematikteki ilgi alanlarını nasıl etkilediğini araştırmak	Scratch	44 ortaokul öğrencisi	Öğrencilerin kodlama ve matematika ilgileri	Karma	Scratch kullanımıyla birlikte kodlama ve matematik etkinliklerinin öğrencilerin ilgilerinin azaldığı fakat oyun ve animasyon kullanımının öğrencilerin ilgisini artırdığı tespit edilmiştir.
Tsai (2019)	App Inventor görsel programlama ortamı kullanılarak öğrencilerin programlama öz-yeterlik seviyelerinin ve programlama kavramlarını algılamalarının tespit edilmesi	App Inventor	180 üniversite öğrencisi	Programlama başarısı	Nicel	Düşük, orta ve yüksek öz-yeterlik seviyesindeki grupların App Inventor kullanılan ve kullanılmayan gruplarda öğrencilerin programlama başarısı incelenmiş ve App Inventor kullanılan grubun özellikle orta ve yüksek öz-yeterlik seviyesindeki gruplarda daha çok olmak üzere arttığı görülmüştür.

## Ek-10. Araştırma İzni



T.C.  
ATATÜRK ÜNİVERSİTESİ REKTÖRLÜĞÜ  
Öğrenci İşleri Daire Başkanlığı

Sayı : 88179374-300-E.1800159222  
Konu : Anket Uygulaması (Alper ÜNAL)

23.05.2018

### EĞİTİM BİLİMLERİ ENSTİTÜSÜ MÜDÜRLÜĞÜNE

Enstitünüz öğretim üyelerinden Yrd.Doç.Dr. Fatma Burcu Topu'nun danışmanlığını yürüttüğü Alper ÜNAL'ın yüksek lisans tezi kapsamında 2017-2018 Öğretim yılı II. Bahar yarıyılı boyunca Antalya İli Alanya İlçesine bağlı liselerde söz konusu uygulama çalışmasına cevaben Antalya Valiliği İl Milli Eğitim Müdürlüğünün 15.05.2018 tarihli ve 98057890-605.01-E.9496638 sayılı yazısı ekte gönderilmiştir.

Bilgilerinizi ve gereğini rica ederim.

Prof.Dr. Medine GÜLLÜCE  
Rektör Yardımcısı

Ek : 15.5.2018 tarihli 98057890-605.01-E.9496638 sayılı belge

Atatürk Üniversitesi Merkez Yerleşkesi 25240 Erzurum  
Tel: +90 442 2311601  
Elektronik Ağ: <http://www.atauni.edu.tr/#!/birim=ogrenci-isleri-daيره-baskanligi>  
Kep Adresi: [atauni@hs01.kep.tr](mailto:atauni@hs01.kep.tr)

Bilgi: Cevahir KARTA  
Faks: +90 442 2361026  
E-Posta: [odaire@atauni.edu.tr](mailto:odaire@atauni.edu.tr)



Bu belge, 5070 sayılı Elektronik İmza Kanununun 5. maddesi gereğince güvenli elektronik imza ile imzalanmıştır.  
<https://ubys.atauni.edu.tr/ERMS/Record/Confirmation/Confirmation?code=8B46FAE>





T.C.  
ANTALYA VALİLİĞİ  
İl Millî Eğitim Müdürlüğü

B-G  
G-8/1

Sayı : 98057890-605.01-E.9496638  
Konu: Anket Uygulaması

15.05.2018

ATATÜRK ÜNİVERSİTESİ REKTÖRLÜĞÜNE  
(Öğrenci İşleri Daire Başkanlığı)  
ERZURUM

İlgi :22/03/2018 tarih ve 1800096787 sayılı yazınız.

Üniversiteniz, Eğitim Bilimleri Enstitüsü Yüksek Lisans Programı Öğrencisi Alper ÜNAL'ın "Lise Öğrencilerine Blok Tabanlı Programlama Öğretiminin Kaygı, Bilişsel Yük ve Başarıya Etkisi " adlı araştırmasını, İlimiz Alanya İlçesine bağlı Liselerde uygulama isteği ile ilgili 22/03/2018 tarih ve 1800096787 sayılı yazısı, İl Millî Eğitim Müdürlüğü Araştırma Değerlendirme ve İnceleme komisyonumuz tarafından, 09/04/2018 tarihinde incelenerek "Millî Eğitim Bakanlığına Bağlı Okul ve Kurumlarda Yapılacak Araştırma, Yarışma ve Sosyal Etkinlik İzinlerine Yönelik İzin ve Uygulama Genelgesi" gereğince uygun görülmüş olup, Müdürlüğümüzün 11/04/2018 tarihli ve 7385826 sayılı onayı ve uygulanacak veri toplama araçları onaylanarak ekte gönderilmiştir.

Araştırmanın bitiminde, sonuç raporunun bir örneğinin CD ortamında (başvuru sahibinin ekte örneği bulunan dilekçe ile) Müdürlüğümüz Ar-Ge bürosuna gönderilmesi hususunda;

Gereğini arz ederim.

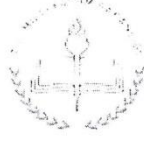
Mehmet KARAKAŞ  
Müdür a.  
Müdür Yardımcısı

EKLER:  
1- Onay ve Ekleri (12 sayfa)  
2-Dilekçe Örneği(1 sayfa)

GÜVENLİ ELEKTRONİK İMZALI  
ASLI İLE AYNI DİR  
16 Mayıs 2018  
MURAT KOYUN  
Memur

Antalya İl Millî Eğitim Müdürlüğü  
Soğuksu Mah. Hamidive Cad. MERKEZ/ANTALYA

Ayrıntılı bilgi için: Mehmet KARAKAŞ Md. Yrd.  
Tel: (0 242) 228 40 00



T.C.  
ANTALYA VALİLİĞİ  
İl Millî Eğitim Müdürlüğü

Sayı : 98057890-20-E.7385826  
Konu : Anket Uygulaması

11.04.2018

İL MİLLÎ EĞİTİM MÜDÜRLÜĞÜNE  
ANTALYA

Atatürk Üniversitesi Eğitim Bilimleri Enstitüsü Yüksek Lisans Programı Öğrencisi Alper ÜNAL'ın "Lise Öğrencilerine Blok Tabanlı Programlama Öğretiminin Kaygı, Bilişsel Yük ve Başarıya Etkisi " adlı araştırmasını, İlimiz Alanya İlçesine bağlı Liselerde uygulama isteği ile ilgili 23/03/2018 tarih ve 1800096787 sayılı yazısı, İl Millî Eğitim Müdürlüğü Araştırma Değerlendirme ve İnceleme komisyonumuz tarafından, 09/04/2018 tarihinde incelenerek "Millî Eğitim Bakanlığına Bağlı Okul ve Kurumlarda Yapılacak Araştırma, Yarışma ve Sosyal Etkinlik İzinlerine Yönelik İzin ve Uygulama Genelgesi" esaslarına uygun olduğu tespit edilmiştir.

Komisyonumuzca, "Lise Öğrencilerine Blok Tabanlı Programlama Öğretiminin Kaygı, Bilişsel Yük ve Başarıya Etkisi" isimli araştırmasını, İlimiz Alanya İlçesine bağlı Liselerde öğrenim gören öğrencilere , bahse konu Genelge ve çalışma takvimi doğrultusunda, eğitim-öğretim faaliyetlerini aksatmaksızın yapılması,

Söz konusu araştırmanın bitimine müteakip; sonuç raporunun bir örneğinin CD ortamında Müdürlüğümüz Ar-Ge bürosuna gönderilmesi kaydıyla uygulanması, Komisyonca uygun görülmüştür.

Makamlarınızca da uygun görüldüğü takdirde, Valilik Makamının 23/02/2015 tarih ve 5347 sayılı yetki devrine göre olurlarınıza arz ederim.

Mehmet KARAKAŞ  
Müdür a.  
Müdür Yardımcısı

OLUR  
11.04.2018

Yüksel ARSLAN  
Vali a.  
İl Millî Eğitim Müdürü

Antalya İl Millî Eğitim Müdürlüğü  
Soğuksu Mah. Hamidiye Cad. MERKEZ/ANTALYA  
E-posta: projeler07@meb.gov.tr

Ayrıntılı bilgi için: Mehmet KARAKAŞ Md. Yrd.  
Tel: (0 242) 238 60 00  
Faks: (0 242) 238 61 11

## ÖZ GEÇMİŞ

1985 yılında Amasya’da doğdu. İlk, orta ve lise öğrenimini Samsun’da tamamladı. 2005 yılında Samsun Ondokuz Mayıs Üniversitesi Eğitim Fakültesi Bilgisayar ve Öğretim Teknolojileri Eğitimi Bölümünde lisans eğitimine başladı ve 2009 yılında mezun oldu. 2010 yılında Diyarbakır ili Yenişehir ilçesi Şehit Öğretmen Nuriye Ak İlköğretim Okulunda Bilişim Teknolojileri Öğretmeni olarak göreve başladı. Daha sonra Bayburt ili Demirözü İlçesi YİBO’da öğretmenlik ve idarecilik görevlerinde bulundu. 2016-2017 eğitim öğretim yılında Atatürk Üniversitesi Eğitim Bilimleri Enstitüsü Bilgisayar ve Öğretim Teknolojileri Eğitimi Anabilim dalında yüksek lisans programına başladı. Halen Antalya İli Alanya İlçesi Hamdullah Emin Paşa Ortaokulu’nda görevine devam etmektedir.

