

**T.C.  
AMASYA ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ  
BİLGİSAYAR VE ÖĞRETİM TEKNOLOJİLERİ EĞİTİMİ ANABİLİM  
DALI**

**BLOK TABANLI PROGRAMLAMA ARAÇLARININ ÖĞRENCİLERİN BAŞARI,  
TUTUM VE KAYGILARI AÇISINDAN KARŞILAŞTIRILMASI**

**Yüksek Lisans Tezi**

**KEMAL BİLİR**

**AMASYA  
Temmuz 2019**

**T.C.  
AMASYA ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ  
BİLGİSAYAR VE ÖĞRETİM TEKNOLOJİLERİ EĞİTİMİ ANABİLİM  
DALI**

**BLOK TABANLI PROGRAMLAMA ARAÇLARININ ÖĞRENCİLERİN BAŞARI,  
TUTUM VE KAYGILARI AÇISINDAN KARŞILAŞTIRILMASI**

**Hazırlayan  
Kemal BİLİR**

**Tez Danışmanı  
Prof. Dr. Feda ÖNER**

**AMASYA-2019**

Kemal BİLİR tarafından hazırlanan "Blok Tabanlı Programlama Araçlarının Öğrencilerin Başarı, Tutum ve Kaygıları Açısından Karşılaştırılması" adlı tez çalışması aşağıdaki jüri tarafından OY BİRLİĞİ/OY ÇOKLUĞU ile Amasya Üniversitesi Bilgisayar ve Öğretim Teknolojileri Eğitimi Anabilim Dalında YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

**Danışman:** Prof. Dr. Feda ÖNER

Bilgisayar ve Öğretim Teknolojileri Eğitimi Anabilim Dalı, Amasya Üniversitesi

Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum/onaylamıyorum .....

**Başkan:**

Anabilim Dalı, Üniversite Adı

Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum/onaylamıyorum .....

**Üye :** Unvanı Adı SOYADI

Anabilim Dalı, Üniversite Adı

Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum/onaylamıyorum .....

Tez Savunma Tarihi: .../.../...

Jüri tarafından kabul edilen bu tezin Yüksek Lisans Tezi olması için gerekli şartları yerine getirdiğini onaylıyorum.

.....

Doç. Dr. Meryem EVECEN

Fen Bilimleri Enstitüsü Müdürü

## ETİK BEYAN

Tezimin içerdiği yenilik ve sonuçları başka bir yerden almadığımı ve bu tezi AÜ Fen Bilimler Enstitüsünden başka bir bilim kuruluşuna akademik gaye ve unvan almak amacıyla vermediğimi; tez içindeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, ayrıca tez yazım kurallarına uygun olarak hazırlanan bu çalışmada kullanılan her türlü kaynağa eksiksiz atıf yapıldığını, bu tezde sunduğum çalışmanın özgün olduğunu bildirir, aksinin ortaya çıkması durumunda aleyhime doğabilecek tüm hak kayıplarını kabullendiğimi beyan ederim. 01/07/2019

İmza  
Kemal BİLİR

## ÖZET

### BLOK TABANLI PROGRAMLAMA ARAÇLARININ ÖĞRENCİLERİN BAŞARI, TUTUM VE KAYGILARI AÇISINDAN KARŞILAŞTIRILMASI

Kemal BİLİR

Amasya Üniversitesi, Fen Bilimleri Enstitüsü

Bilgisayar ve Öğretim Teknolojileri Eğitimi Anabilim Dalı Yüksek Lisans, Temmuz/2019

Danışman: Prof. Dr. Feda ÖNER

Son yıllarda hemen hemen herkesin programlama öğrenmesi gerektiği dile getirilmektedir. Bunun nedeni 21.yüzyılda geliştirilmek istenen problem çözme ve yaratıcılık gibi becerilerin programlama ile geliştirilebileceği düşüncesidir. Özellikle programlama yapmanın geleneksel zorluk algısından kurtarılması yönüyle Blok Tabanlı Programlama (BTP) daha fazla işlevsel hale gelmektedir. BTP ise kendi içerisinde farklı öğretim yöntemleri ile sunulmakta ve bireylerin daha fazla seçeneğe sahip olmalarını sağlamaktadır. Bu çalışmada farklı yapılarıdaki BTP araçlarının öğrencilerin programlama başarılarına, programlamaya karşı tutumlarına ve programlamaya karşı kaygılarına etkileri incelenmiştir. Bu amaç doğrultusunda çalışma 2017-2018 eğitim öğretim yılı güz döneminde Amasya Üniversitesi Bilgisayar ve Öğretim Teknolojileri Bölümünde Programlama-I dersi alan 43 2.sınıf öğrencisi ile yürütülmüştür. Kontrol grubunda 10 kadın, 11 erkek öğrenci deney grubunda 10 kadın, 12 erkek öğrenci yer almıştır. Çalışmada nicel araştırma yöntemlerinden kontrol gruplu ön test-son test yarı deneysel desen kullanılmıştır. Katılımcılara uygulamadan önce algoritma başarı ön testi, programlamaya karşı tutum ön testi ve programlamaya karşı kaygı ön testi uygulanmıştır. Kontrol grubuna Scratch BTP aracı ile deney grubuna ise HackerCan BTP aracı ile dört haftalık programlama eğitimi verilmiştir. Eğitim sonunda her iki gruba da algoritma başarı son testi, programlamaya karşı tutum son testi ve programlamaya karşı kaygı son testi uygulanmıştır. Analiz sonuçlarına göre ön test algoritma başarıları benzer çıkan grupların son test başarılarında kontrol grubunun puanları yüksek olmasına rağmen aralarında anlamlı bir fark ortaya çıkmamıştır. Programlamaya karşı kaygı ve programlamaya karşı tutum ön testleri de benzer sonuçlar vermiştir. Eğitim sonunda öğrencilerin kaygılarında da azalma olduğu ortaya çıkmıştır. Ancak deney grubunun programlamaya karşı kaygısında anlamlı bir azalma olmasına rağmen kontrol grubundaki kaygı azalması anlamlı değildir. Programlamaya karşı tutum son test sonuçlarına bakıldığında ise öğrencilerin tutumlarının programlama eğitimi aldıktan

sonra olumlu yönde deđiřtiđi gözlemlenmiř ancak kontrol ve deney grupları arasında anlamlı bir farkın olmadığı gözlemlenmiřtir. Bu sonuçlar BTP eđitiminde her iki yönteme dayalı olarak düzenlenen araçların öğrenci başarısını arttırdığını, öğrencilerin tutumlarını olumlu yönde etkilediđini ve öğrencilerin kaygılarını azalttıđını göstermiřtir.

**Anahtar Kelimeler:** Algoritma, Programlama, Blok Tabanlı Programlama, Başarı, Tutum, Kaygı



## ABSTRACT

### COMPARISON OF BLOCK BASED PROGRAMMING TOOLS IN TERMS OF STUDENTS' ACHIEVEMENTS, ATTITUDES, AND LEVEL OF ANXIETY

Kemal Bilir

Amasya University, Institute Of Science

Department Of Computer Education And Instructional Technology, Master, July/2019

Supervisor: Prof. Dr. Feda ÖNER

In recent years, it is highlighted that almost everyone should learn programming. The main reason for this is the idea that the skills such as problem solving and creativity, desired to be developed in the 21st century, can be developed by programming. Block Based Programming (BBP) is becoming much more functional, especially in terms of eliminating the traditional perceptions about the difficulty of programming. Furthermore, BBP is offered through different teaching methods and enables individuals to have more options. In this study, the effects of a variety of BBP tools on students' success in programming, their attitudes towards programming and their concerns about programming were examined. For this purpose, the study was carried out with 43 students, studying at the Department of Computer Education and Instructional Technology at Amasya University, in the fall semester of 2017-2018 academic year. They were in their second year of the study and taking Programming-I course. The control group consisted of 10 female and 11 male students. The experimental group consisted of 10 female and 12 male students. In the study, employing quantitative research method, pretest-posttest quasi-experimental design with control group was used. The pre-tests on Algorithm achievement, attitudes towards programming and anxiety against programming were applied to the participants before the implementation. The control group got training by means of Scratch BBP tool and the experimental group received training with the HackerCan BBP tool for four weeks. At the end of the training, the post-tests on algorithm achievement, attitude towards programming and anxiety towards programming were carried out with both groups. According to the analysis results, there was no significant difference between the algorithm achievement post-test scores of the control group and the experimental group, although the control group had higher scores than the experimental group. At the end of the training, it was found out that there was a decrease in students' level of anxiety. However, although there was a significant decrease in the level of anxiety of the experimental group against programming,

the decrease in the level of anxiety of the control group was not significant. When the analysis results of attitudes towards programming post-test were examined, it was found out that the students' attitudes changed positively after receiving the programming education, but there was no significant difference between the control and experimental groups in this respect. These results shed a light on the fact that tools designed on the basis of both methods in BBP education increased students' achievement, positively affected students' attitudes and reduced students' level of anxiety.

**Keywords:** Algorithm, Programming, Block Based Programming, Success, Attitude, Anxiety





## ÖN SÖZ

Son yıllarda popüler olan blok tabanlı programlama ve kodlama alanları kendine yeni bir alan oluşturmuştur. Bu çalışmada bu güncel alandan yola çıkarak hazırlanmış ve daha etkili bir Blok Tabanlı Programlama eğitiminin verilmesine katkı sağlayacağı düşünülmektedir.

Kemal BİLİR



## TEŞEKKÜR

Çalışmam boyunca desteğini ve emeğini esirgemeyen, büyük bir özveri ve sabırla tez danışmanlığımı yapan, bu yolda çalışma ve tanışma fırsatı bulduğum Prof. Dr. Feda ÖNER'e şükranlarımı sunarım.

Ders ve tez aşamasında bilgilerinde ve tecrübelerinden faydalandığım çalışmama katkılar sağlayan özellikle Dr. Öğr. Üyesi Fatih SALTAN ve Doç. Dr. Recep ÇAKIR başta olmak üzere bütün hocalarıma teşekkürlerimi sunarım.

Özellikle bu günlere gelmemi sağlayan, maddi manevi desteklerini esirgemeyen anneme ve babama sonsuz teşekkürlerimi sunarım.

## İÇİNDEKİLER

ETİK BEYAN.....	iv
ÖZET .....	v
ABSTRACT.....	vii
ÖN SÖZ.....	ix
TEŞEKKÜR .....	x
TABLolar DİZİNİ.....	xiii
RESİMLER DİZİNİ .....	xv
KISALTMALAR DİZİNİ.....	xvi

### I. BÖLÜM

1. GİRİŞ .....	1
1.1. Araştırmanın Amacı .....	4
1.2. Araştırmanın Alt Problemleri .....	4
1.3. Araştırmanın Gerekçesi ve Önemi .....	4
1.4. Araştırmanın Sınırlılıkları .....	5
1.5. Araştırmanın Varsayımları .....	5
1.6. Tanımlar .....	5
1.6.1. Algoritma .....	6
1.6.2. Program Görselleştirme .....	6
1.6.3. Algoritma Animasyonları .....	6
1.6.4. BTP araçları.....	6
1.6.5. Scratch .....	6
1.6.6. HackerCan.....	6
1.6.7. Algoritma görselleştirme .....	6

### II. BÖLÜM

2. KURAMSAL ÇERÇEVE VE İLGİLİ ARAŞTIRMALAR.....	8
2.1. Algoritma .....	8
2.2. Programlama.....	8
2.3. Blok Tabanlı Programlama (BTP) .....	9
2.4. BTP Araçları .....	10
2.4.1. Alice.....	10
2.4.2. Scratch .....	11
2.4.3. Code.org.....	12
2.4.4. HackerCan.....	13
2.4.5. Blockly .....	14

2.4.6.	Kodu Game Lab .....	15
2.5.	İlgili Çalışmalar .....	16
III. BÖLÜM		
3.	YÖNTEM.....	20
3.1.	Araştırma Modeli .....	20
3.2.	Çalışma Grubu .....	20
3.3.	Veri Toplama Araçları .....	21
3.3.1.	Algoritma Başarı Testi .....	21
3.3.2.	Bilgisayar Programlama Kaygı Ölçeği.....	22
3.3.3.	Bilgisayar Programlamaya Karşı Tutum Ölçeği.....	22
3.4.	Uygulama Süreci .....	23
3.5.	Veri Analizi .....	24
IV. BÖLÜM		
4.	BULGULAR.....	25
4.1.	Farklı Blok Tabanlı Programlama Aracı Kullanılmasının Öğrencilerin Akademik Başarılarına Etkisine İlişkin Bulgular.....	25
4.2.	Farklı Blok Tabanlı Programlama Aracı Kullanımının Öğrencilerin Bilgisayar Programlamaya Karşı Tutumlarına Etkisine İlişkin Bulgular.....	28
4.3.	Farklı Blok Tabanlı Programlama Aracı Kullanımının Öğrencilerin Bilgisayar Programlamaya Karşı Kaygılarına Etkisine İlişkin Bulgular.....	32
V. BÖLÜM		
5.	TARTIŞMA.....	36
VI. BÖLÜM		
6.	SONUÇLAR ve ÖNERİLER .....	39
6.1.	Sonuçlar .....	39
6.2.	Öneriler .....	40
6.3.	Araştırmacılar İçin Öneriler: .....	40
6.4.	Öğretmenler İçin Öneriler: .....	40
7.	Kaynakça .....	41
EKLER .....		47
EK-1. Algoritma Başarı Testi .....		48
EK-2. Bilgisayar Programlamaya Karşı Tutum Ölçeği .....		53
EK-3. Programlama Kaygı Ölçeği .....		55
EK-4. Çalışma Esnasından Fotoğraflar .....		57
ÖZGEÇMİŞ .....		58

## TABLOLAR DİZİNİ

<b>Tablo 3.1.</b> Çalışma Grubu Dağılımı Tablosu.....	21
<b>Tablo 3.2.</b> Kazanımların Soru Dağılımları Tablosu.....	22
<b>Tablo 3.3.</b> Veri Toplama Ve Uygulama Aşamaları.....	24
<b>Tablo 4.1.</b> Akademik Başarı Ön test ve Son test Puanlarının Shapiro-Wilks Testi Sonuçları.....	25
<b>Tablo 4.2.</b> Deney ve Kontrol Grubu Akademik Başarı Ön test Puanları Bağımsız Örneklem t-Testi Sonuçları.....	26
<b>Tablo 4.3.</b> Deney ve Kontrol Grubu Akademik Başarı Son test Puanları Bağımsız Örneklem t-Testi Sonuçları.....	26
<b>Tablo 4.4.</b> Deney Grubu Ön test - Son test Puanları İlişkili Örneklem t-Testi Sonuçları.....	26
<b>Tablo 4.5.</b> Kontrol Grubu Ön test - Son test Puanları İlişkili Örneklem t-Testi Sonuçları.....	27
<b>Tablo 4.6.</b> Deney Grubunun Cinsiyete göre Akademik Başarı Son test Puanları Bağımsız Örneklem t-Testi Sonuçları.....	27
<b>Tablo 4.7.</b> Deney Grubunun Cinsiyete göre Akademik Başarı Son test Puanları Bağımsız Örneklem t-Testi Sonuçları.....	28
<b>Tablo 4.8.</b> Programlamaya Karşı Tutum Ön test ve Son test Puanlarının Shapiro-Wilks Testi Sonuçları.....	28
<b>Tablo 4.9.</b> Deney ve Kontrol Grubu Bilgisayar Programlamaya Karşı tutum Ön test Puanları Bağımsız Örneklem t-Testi Sonuçları.....	29
<b>Tablo 4.10.</b> Deney ve Kontrol Grubu Bilgisayar Programlamaya Karşı tutum Son test Puanları Bağımsız Örneklem t-Testi Sonuçları.....	29
<b>Tablo 4.11.</b> Deney ve Kontrol Grubunun Bilgisayar Programlamaya Karşı tutum alt boyutları Ön test Puanları Bağımsız Örneklem t-Testi Sonuçları.....	30
<b>Tablo 4.12.</b> Deney Grubu ve Kontrol Grubu Bilgisayar Programlamaya Karşı tutum alt boyutları Son test Puanları Bağımsız Örneklem t-Testi Sonuçları.....	31

<b>Tablo 4.13.</b> Programlamaya Karşı Kaygı Ön test ve Son test Puanlarının Shapiro-Wilks Testi Sonuçları.....	33
<b>Tablo 4.14.</b> Deney ve Kontrol Grubu Programlamaya Karşı Kaygı Ön test Puanları Bağımsız Örneklem t-Testi Sonuçları.....	33
<b>Tablo 4.15.</b> Deney ve Kontrol Grubu Programlamaya Karşı Kaygı Son test Puanları Bağımsız Örneklem t-Testi Sonuçları.....	33
<b>Tablo 4.16.</b> Deney Grubu Ön test - Son test Programlamaya Karşı Kaygı Puanları İlişkili Örneklem t-Testi Sonuçları.....	34
<b>Tablo 4.17.</b> Kontrol Grubu Ön test - Son test Programlamaya Karşı Kaygı Puanları İlişkili Örneklem t-Testi Sonuçları.....	34



## RESİMLER DİZİNİ

<b>Resim 2.1.</b> Alice Ekran Görüntüsü .....	11
<b>Resim 2.2.</b> Scratch Ekran Görüntüsü .....	12
<b>Resim 2.3.</b> Code.Org Ekran Görüntüsü .....	13
<b>Resim 2.4.</b> HackerCan Ekran Görüntüsü .....	14
<b>Resim 2.5.</b> Blockly Ekran Görüntüsü .....	15
<b>Resim 2.6.</b> Kodu Game Lab Ekran Görüntüsü .....	16



## KISALTMALAR DİZİNİ

**EBA:** Eğitim Bilişim Ağı

**MEB:** Milli Eğitim Bakanlığı

**TTKB:** Talim Terbiye Kurulu Başkanlığı

**BTP:** Blok Tabanlı Programlama





## I. BÖLÜM

### 1.GİRİŞ

Son yıllarda, bütün dünyada bireylerin erken yaşlarda programlamayı öğrenmeleri gerekliliği vurgulanmakta ve bu yönde adımlar atılmaktadır. Bu adımların atılmasındaki başlıca nedenlerden biri, 21.yüzyıl becerilerinin ortaya çıkardığı gerekliliktir (Sayın ve Seferoğlu, 2016). 21.yüzyıl becerileri kavramı, gelecekteki dünyaya çocukların hangi beceriler ile hazırlanması gerektiğini açıklayan kavram olarak tanımlanmaktadır (Roy Singh, 1991). 21.yy'da bireylerde olması istenilen bilgi edinme, üretken olma, kariyer planlaması yapma gibi özelliklerle beraber kodlama ve programlama öğrenimi de ön plana çıkmaktadır. Bunların yanı sıra problem çözme, algoritmik ve bilgisayarca düşünme becerileri gibi temeli programlamaya dayanan becerilerin programlama ve bilgisayar eğitimi ile kazandırılması da 21.yy becerilerinin yapı taşı oluşturmaktadır (Güneş ve Karabak, 2013; Shin, Park, ve Bae, 2013). Kodlama ve algoritma mantığının bireylerce özellikle küçük yaşlardan itibaren kavranması ile bireylerin yeni fikirler bulmalarını, yaratıcılıklarının ve üretkenliklerinin artmasını sağlayacak, karşılarına çıkan sorunlara çözüm üretebilmelerini ve iş birliği içerisinde yapacakları çalışmalar ile bireylerin gelişimine destek olacaktır (Demirer ve Sak, 2016). Alan yazına bakıldığında, programlama yapabilmek 21.yy'da bireylerin kazanması gereken birçok beceriye temel oluşturmakta ve bu becerilerin geliştirilmesini hızlandırmaktadır (Williams ve Cernochova, 2013). Günümüzde programlamanın öğrenciler ve farklı iş alanlarındaki bireyler için çok fazla önemi olduğu söylenebilir (URL-1, 2018). Artık ülkelerin kalkınma düzeyleri dijitalleşmekte ve sosyal anlamda programlama yapmanın da önemi artmaktadır (Sayın ve Seferoğlu, 2016).

21.yy becerilerinden olan algoritma ise birçok alanda olduğu gibi bilgisayar biliminde de önemli yere sahiptir. Algoritma, (Futschek, 2006)'e göre tanımlanan yönergeleri tam olarak izleyerek problem çözmeye yarayan bir yöntem olarak tanımlanmaktadır. Algoritma ardışık adımlardan oluşan problemin çözümü için bu adımları anlamlı bir biçimde sıraya koyan bir yapı olarak da tanımlanabilir. Algoritma karışık bir yapıda görünmesine karşın kolay bir yapıdadır ancak bu itici algı bireylerin algoritmaya karşı önyargı ile yaklaşmasına neden olmakta ve buna bağlı olarak da bireyler programlamaya karşı da olumsuz düşünceye kapılmaktadırlar. Algoritma öğretiminin soyut bir süreç olması öğrencilerin programlama öğrenmeye karşı olumsuz tavır içinde olmasına neden olmakta ve öğrenme

sürecinde sıkıntılar yaşamasına sebep olmaktadır. Bundan dolayı algoritma öğretimini somutlaştırmak amacıyla çeşitli yazılımlar geliştirilmiştir (Erol, 2015).

Algoritma görselleştirme yazılımları bireylerin algoritmaya karşı takındıkları olumsuz tavırları ortadan kaldırmak ve algoritma yapısının daha anlaşılabilir hale gelmesini sağlamak adına geliştirilmiş kullanışlı uygulamalardır (Saygıner ve Tüzün, 2017). Algoritma görselleştirme araçları, kullanıcıya yazılım yardımıyla akış şeması oluşturma olanağı veren yazılımlardır. Bu tür araçlar kullanıcılara kendi oluşturdukları algoritma yapısını diyagramlar ile test etmelerini ve yaptıkları programın hatalarını en aza indirme olanağını sağlar (Powers, ve diğerleri, 2006). Algoritma görselleştirmeleri, bilgisayar algoritmalarının dinamik davranışını grafik olarak göstermektedir ve bu teknolojinin amacı bilgisayar bilimleri alanında çalışan öğrencilerin algoritmaları daha iyi öğrenmelerini amaçlayan sistemler olarak tanımlanmaktadır (Hundhausen ve Brown, 2008). Algoritma görselleştirmeleri her yaşta bireyin ilgisini çekmekte ve programlamaya karşı olan düşüncelerini olumlu yönde etkilemekte, yaratıcı düşünme becerilerini geliştirmektedir. Görselleştirme kapsamında, öğrenilmesi kolay ve görselliği ön planda tutan programlama ortamları geliştirilmiştir. Literatürde bu doğrultuda geliştirilen programlama ortamlarının programlama öğretimindeki etkisini ortaya çıkarmak amacıyla çok sayıda çalışma yapılmıştır (Kalelioğlu ve Gülbahar, 2014; Shin ve Park, 2014; Yükseltürk ve Altıok, 2016).

Kodlama yapan bireyler özellikle ilk yaptığı uygulamada başarılı olmaları durumunda bundan haz almaktadırlar. Bunun nedeni insandaki başarıya duyusunun ortaya çıkması ve bireyin kendine özgü bir şeyler ortaya koyması ve bu sayede üretkenlik olgusunun bireyde fark edilmesidir. İnsan başarılı oldukça daha fazlasını ister, sorunla karşılaştığında ise hırslanarak bunun üstesinden gelmeye çalışır bu noktada programlama yapmak insan gelişiminde ön plana çıkar (Yecan, Özçınar, ve Tanyeri, 2017). Bu alanda özel yeteneği ve ilgisi olan bireylerin bu yeteneklerini küçük yaşlarda fark etmelerini sağlamak önem arz etmektedir. Bu yeteneklerinin fark edilmesi için ise küçük yaşlardan itibaren bireylere fırsat verilmeli, programlama yapmalarına imkân verilmelidir. Bireylerin sadece hazır olan uygulamaları, teknolojiyi kullanmaları onları hazırcılığa alıştırıp üretkenliklerinin ortaya çıkmasına engel olmakta onların gelişimine zarar vermektedir. Günümüz şartlarını göz önüne aldığımızda tüketim toplumundan üretim toplumuna geçiş yapmalı, bireylerin üretken ve kendini geliştiren kişilikler haline gelmesini sağlamalıyız (Aytekin, Çakır, Yücel, ve Kulaöz, 2018). Alan yazına bakıldığında, küçük yaşta bireylere programlama etkisinin incelendiği bir çalışmada; programlama eğitimi alan öğrencilerin üst biliş ve yönlendirme, yaratıcılık yetenekleri ve farklı düşünme yeteneklerinin programlama eğitimi almayan öğrencilerden daha yüksek olduğu bulunmuştur (Clements ve Gullo, 1984). İlköğretim

öğrencilerinin yaratıcılık düzeylerinin Scratch programından etkilenmesi üzerine yapılan bir diğer çalışmada ise benzer sonuçlara ulaşılmış, Scratch programının öğrencilerin yaratıcılık becerisi, akıcı düşünme ve girişkenlik üzerinde anlamlı bir etkiye sahip olduğu ortaya çıkmıştır (Kobsiripat, 2015). Farklı zamanlarda yapılan bu araştırmalara bakıldığında dönemsel olarak programlama eğitiminin öğrencilere yaratıcılık, sistematik ve algoritmik düşünme gibi beceriler kazandırmada etkileri incelenmiş ve bu kazanımların programlama ile ilişkisi vurgulanmıştır. Blok Tabanlı Programlama (BTP) araçlarının kullanıldığı çalışmalarda ise öğrencilerin programlamanın temellerini daha iyi anladığı, programlama öğrenme fikrinin bu araçlar sayesinde öğrencilerde heyecana yol açtığı gözlemlenmiş ve bu sayede programlamaya karşı motivasyonları artmıştır (Calder, 2010). Başka bir çalışmada ise BTP ile eğitim gören öğrencilerin programlama eğitiminde yaptıkları uygulamaları bir bütün olarak görmelerini ve yanlışlarını anında düzeltmelerini sağlamasından dolayı öğrencilerin bilişsel becerilerinin gelişimine katkı sağladığı ifade edilmektedir (Törley, 2014).

Türkiye’de de son dönemlerde BTP araçları ile ilgili çalışmalara sıkça rastlanmaktadır. 2012-2013 eğitim öğretim yılından itibaren Türkiye’de değişen eğitim sistemiyle birlikte 5. ve 6. sınıflarda Bilişim Teknolojileri ve Yazılım dersi kazanımları kapsamında yer alan “problem çözme, programlama ve özgün ürün geliştirme” için blok tabanlı uygulamalarla çocukların eğlenerek ve istekle öğrenerek uygulama geliştirebilecekleri programlar kullanılmaktadır (Berksoy, Sözcü, Armağan, ve Arslan, 2016). Bu ders kapsamında bireylerin paylaşmaya ve iş birliği içerisinde geliştirmeye dayalı programlama ortamlarının kullanılması önerilmektedir (URL-2, 2019). Bu bağlamda yapılan çalışmalar değişen eğitim sistemini destekler niteliktedir. (Uslu, Mumcu, ve Eğin, 2018) yaptıkları çalışmada BTP eğitimi alan öğrencilerin programlama ile ilgili görüşlerini incelediklerinde, öğrencilerin programlamanın hayal güçlerini zenginleştirdiğini, bilgisayar bilimi alanındaki farkındalıklarının arttığı görülmüştür. Diğer bir çalışmada ise BTP aracının programlama kavramının öğrenilmesinde oyun vb. animasyonların hazırlanmasında öğrencilerin gelişmesine katkısı olduğu görülmektedir (Ozoran, Çağıltay, ve Topallı, 2012). Bunlar göz önüne alındığında öğrencilerin kişisel gelişimlerini de desteklemesinden ve anlama ve uygulamayı kolaylaştırması bakımından BTP araçlarının kullanılması önerilmektedir (Erümit, Benzer, Aksoy, Aksoy, ve Şahin, 2017). BTP araçları, özellikle çocuklar için birçok kolaylık sağlayan, algoritmanın yapısının kavranmasına yardımcı olan araçlardır. Çocukların sadece hazır uygulamaları kullanmak yerine kendi istek ve ihtiyaçları doğrultusunda uygulama veya oyun yapmaları bu araçlar sayesinde kolay hale gelmektedir. Bunun yanı sıra çocukların kodlama yaparken eğlenmeleri ve iş birliği içerisinde olmaları onların gelişimine büyük katkılar sağlayacaktır.

### 1.1. Araştırmanın Amacı

Araştırmanın amacı, BTP eğitiminde blok tabanlı programlamayı farklı yöntemler ile kullanan araçların öğrencilerin algoritma başarıları, programlamaya karşı kaygıları ve tutumları üzerinde olan etkilerini inceleyerek öğrencilere algoritmanın temelini öğretmek. Bu amaç doğrultusunda, oluşturulan alt problemler şu şekildedir.

### 1.2. Araştırmanın Alt Problemleri

1. Farklı blok tabanlı programlama aracı kullanılması öğrencilerin başarılarını etkilemekte midir?
2. Farklı blok tabanlı programlama aracı kullanarak programlama dersi alan öğrencilerin ön test - son test programlama başarıları arasında anlamlı bir farklılık var mıdır?
3. Farklı blok tabanlı programlama aracı kullanımı öğrencilerin bilgisayar programlamaya karşı kaygıları ve tutumlarını etkilemekte midir?
4. Farklı blok tabanlı programlama aracı kullanımı öğrencilerin bilgisayar programlamaya karşı tutum alt boyutlarını etkilemekte midir?
5. BTP eğitimindeki farklı nitelikteki araçların kullanımında erkek ve kız öğrenciler arasında başarı farklılıkları var mıdır?

### 1.3. Araştırmanın Gerekçesi ve Önemi

Programlama öğretiminde ortaya çıkan sıkıntıları gidermek amacıyla proje tabanlı, işbirlikçi ve oyun tabanlı öğrenme gibi farklı yaklaşımlar önerilmektedir. Bu yaklaşımlara ek olarak blok tabanlı görsel programlamada bulunmaktadır. Bu ortamlarda kullanıcılar, öğrenmede zorluk yaşadıkları soyut kodları, adım adım somutlaştırarak öğrenmekte ve anında dönüt alabilmektedirler (Başer, 2013). Kodların blok yapısında görsel bir halde olması, programlama yaparken unutmayı kodların mantığını öğrenmek yerine ezberlemeyi ya da programın algoritmasını oluşturmada zorluk yaşanmasını engellemektedir (Kinnunen ve Malmi, 2008). Özellikle programlama öğrenmeye yeni başlayan bireylerin programlama yapmaya olan isteğini artırarak araştırmaya yöneltmekte, sistematik ve mantıksal düşünme becerilerini geliştirmekte ve özgüvenlerini artırmaktadır (Genç ve Karakuş, 2011). Programlamanın temeli olan algoritmanın bireyler açısından zor ve sıkıcı bulunması

nedeniyle BTP araçlarının önemi artmaktadır. Öyle ki piyasada birçok BTP aracı bulunmaktadır. Öğrencilerin genellikle zor gördükleri programlama dersi bu araçlar vasıtasıyla kolay ve ilgi çekici bir hale gelmektedir. Özellikle lisans düzeyinde programlama ile ilk defa karşılaşacak öğrencilerin daha temelden ve basitleştirilerek programlama eğitimine başlamaları o öğrencilerin özgüvenlerini arttıracak ve programlamaya karşı olan olumsuz düşüncelerden sıyrılacaklardır.

Bu nedenlerden ötürü lisans düzeyindeki Bilgisayar ve Öğretim Teknolojileri öğretmen adaylarının meslek hayatlarında kullanacakları bu araçların temelini öğrenmeleri ve bu araçlar hakkında fikir sahibi olmaları önem arz etmektedir. Çalışma sonunda programlama dersi kapsamında eğitim veren eğitimcilere uygun BTP aracının hangi yapıda olduğu bilgisini vermek ve eğitim alan öğrencilerin de programlama konusunda daha olumlu düşüncelere sahip olmaları beklenmektedir.

#### **1.4. Araştırmanın Sınırlılıkları**

1. Bu araştırma 2016-2017 Eğitim Öğretim yılında Amasya Üniversitesi Bilgisayar ve Öğretim Teknolojileri bölümünde eğitim gören 2. sınıf öğrencileri ile sınırlandırılmıştır.
2. Araştırma Dört hafta ile sınırlıdır.
3. Araştırma öğrencilere uygulanan Algoritma başarı testi, Programlamaya Karşı Kaygı Ölçeği, Programlamaya Karşı Tutum Ölçeği ve görüşme Formu ölçme araçları ile sınırlıdır.

#### **1.5. Araştırmanın Varsayımları**

- 1.Çalışmaya katılan öğrencilerin ölçme araçlarındaki soruları samimiyetle ve doğru biçimde cevapladıkları varsayılmaktadır.

#### **1.6. Tanımlar**

Bu bölümde çalışmamızda yer alan kavramlar ve uygulamaların açıklamalarına yer verilmiştir.

### **1.6.1.Algoritma**

(Futschek ve Moschitz, 2010)'e göre algoritma yeni fikirler sunarak önemli bir ilerleme elde etmeyi mümkünleştiren farklı yolların ortaya çıkmasını sağlayan güçlü bir araçtır.

### **1.6.2.Program Görselleştirme**

Bu tip görselleştirme aracı, bir programın ve onun verilerinin nasıl çalıştığını grafiksel olarak sunmak için kullanılır (Bergin, ve diğerleri, 1996).

### **1.6.3.Algoritma Animasyonları**

Temel algoritmaların hangi işlemleri tamamladığını görseller yardımıyla göstermek amacıyla kullanılır (Bergin, ve diğerleri, 1996).

### **1.6.4.BTP araçları**

Programları bir ara yüz yardımıyla, sürükle–bırak biçiminde yapılandırmayı sağlayan araçlardır (Powers, ve diğerleri, 2006).

### **1.6.5.Scratch**

Algoritma ve programlama öğretiminde kullanılan, kullanımı kolay, hikâye, animasyon ve oyun tasarımına olanak tanıyan grafik tabanlı bir yazılım (Maloney, Resnick, Rusk, Silverman, ve Eastmond, 2010).

### **1.6.6.HackerCan**

HackerCan öğrencilerin ana dilde kodlamayı öğrenme ve Türkçe kod yazarak yazılım geliştirme becerisi kazandırmak için hazırlanan oyun tabanlı öğrenme modeli üzerine kurgulanmış bir platformdur.

### **1.6.7.Algoritma görselleştirme**

(Müldner, Shakshuki, Kerren, Shen, ve Bai, 2005) göre algoritma görselleştirilmesi gözle görülmeyen bir şeyin akılda canlandırılması olayıdır. (Hundhausen, Douglas, ve

Stasko, 2002) göre ise, algoritma görselleřtirmeleri, algoritmaları eylem halinde iken göstererek, bilgisayar bilimleri alanında alıřan ğrencilerin algoritmaları daha iyi ğrenmelerini amalayan sistemlerdir.



## II. BÖLÜM

### 2.KURAMSAL ÇERÇEVE VE İLGİLİ ARAŞTIRMALAR

#### 2.1.Algoritma

Algoritma kelimesi, 9. yüzyılda matematik alanında yazmış olduğu "Hisab el-cebr ve'l-mukabele" adlı kitap ile Türk Matematikçi Harezmi tarafından ortaya atılmıştır ve algoritmanın kurucusu olarak kabul edilmektedir (Kuzu, 2013). Algoritma genellikle bilgisayar ile ilgili bir kavram olarak bilinir ancak algoritma sadece bilgisayar biliminde kullanılmamakta aslında algoritmanın temeline dayanan matematik gibi birçok bilim alanında da kullanılmaktadır. Matematik biliminde problemlerin adım adım çözüm yolunu göstermek amacıyla kullanılan algoritma bilgisayar biliminde de aynı mantık ile çalışır. Programlama incelendiğinde temelinde algoritma görülmektedir. Bununla birlikte algoritmanın birtakım şekillerle ifade edilmesi ile de akış diyagramları ortaya çıkmaktadır. (Garner, 2003) Akış diyagramının doğru bir şekilde yapılabilmesi ve algoritmanın gerçekleşmesi için aşağıdaki basamakların sırasıyla uygulanması gerektiğini belirtmiştir.

- Problemin Analiz Edilmesi
- Algoritmanın Tasarlanması ve Geliştirilmesi
- Geliştirilen Algoritmanın Uygulanması

Yukarıdaki basamaklara bakıldığında öncelikle problem durumunun ne olduğu konusuna değinilmiş ve bu problemin tanımlanmasını belirtilmiştir. İkinci ve en önemli basamak olan problemin analizinden sonraki işlem olan algoritmanın tasarlanması ve geliştirilmesidir. Ortaya sağlıklı bir algoritma çıkarmak programlama yapmanın en önemli adımlarından biridir.

#### 2.2.Programlama

Programlama varlığı bilinen bir problemi çözmek için bilgisayarın anlayabileceği bir dil ile o probleme çözüm üretmek olarak tanımlanmaktadır (Roy ve Haridi, 2004). Üretilen ilk



bilgisayar ile birlikte kullanıcıların bilgisayarla ortak bir dil üzerinden iletişim kurma gereksinimi ortaya çıkmıştır. Bu ihtiyaç makine dili ile giderilmeye çalışılmış fakat bu dilin zorluğu sebebiyle daha kolay yazılıp anlaşılabilen programlama dilleri geliştirilmiştir. Programlama dili, kullanıcının bilgisayardan ne yapmasını istediğini anlatabilmek için kullandığı sözdizimleridir (Roy ve Haridi, 2004).

(Akpınar ve Altun, 2014), programlama öğretiminin öğrencilerin düşünme becerilerini, teknoloji okuryazarlıklarını, işbirlikli çalışmalarını, bilgisayar ile öğrenme becerilerini geliştirdiğini belirtmişlerdir. (Shin ve Park, 2014) ise son yıllarda, dünyadaki bütün uzmanların bireylerin erken yaşlarda programlamayı öğrenmesi gerektiği görüşünü savunmaktadırlar. Bilgisayar programlamanın küçük yaştaki öğrencilere etkisini inceleyen (Clements ve Gullo, 1984), programlama yapan yedi yaşındaki öğrencilerin yansıtıcılık ve yaratıcılıkları ile üst biliş yetenekleri ve yönlendirme yeteneklerinin programlama yapmayan öğrencilerden daha yüksek olduğunu ortaya koymuşlardır. (Gorman ve Bourne, 1983) bir diğer araştırmada ise 12 yaşındaki öğrencilerle farklı sürelerde programlama öğrenen öğrencilerin kural öğrenmede farklı performans sergilediklerini ve haftada bir saat programlama etkinliği yapanların haftada yarım saat programlamaya katılanlara göre kural öğrenmede daha iyi olduklarını bulmuşlardır. Bu çalışmalar göz önüne alındığında programlama eğitiminin bireylerin kişisel gelişimi için önemli olduğu görülmektedir.

### 2.3. Blok Tabanlı Programlama (BTP)

BTP, programlama dilinin daha kolay ve anlaşılabilir hale getirilmiş şeklidir. BTP dilleri, bilgisayar programı geliştirmek amacıyla oluşturulan, belli bir sıralama ve akış şeması dâhilinde ortaya çıkarılan metin girdilerinden farklı olarak bloklar, düğümler, akış diyagramları gibi görsel nesnelerin daha çok “sürükle-bırak” şeklinde bir araya getirilerek oluşturulduğu dillerdir. BTP sayesinde programlamaya karşı olan zorluk algısı yıkılmış, küçük yaşlardan itibaren programlama eğitimleri vermeye başlanmıştır. Programlama dillerinin fazlaca bilgi istemesi ve algoritma kurulumlarına karşın BTP bunları sadeleştirerek ve görselleştirerek programlama yapan kişilere kolaylık sağlamaktadır. Bu kolaylık küçük yaştaki bireyler için daha belirgindir. Algoritmanın temel yapısını ve mantığını öğrenmeleri açısından BTP ortamlarının önemi fazladır. Ayrıca BTP sayesinde özellikle öğrenciler herhangi bir programlama ve algoritma eğitimi almadan da algoritma ve programlamanın yapısını anlayabilir ve başlangıç yapabilirler.

BTP'nin programlama eğitiminde kullanılmasının birçok nedeni vardır. (Kaucic ve Asic, 2011) programlama öğretiminde görsel teknolojilerin kullanıldığı araştırmaları incelemişler

ve yapılan arařtırmaların ortak bulgulara ulařtıklarına deęinmiřlerdir. Bu alıřmalardan yola ıkarak BTP teknolojilerinden yararlanmanın ğrenci motivasyonunun ve ğrencilerin programlamaya karřı tutumlarının olumlu etkilendięi grlmektedir. (Shu, 1999) da BTP kullanımının programlamaya katkı saęladığını savunmuř, grsel řemaların daha fazla anlam ierdięini, hatırlamayı ve anlamayı daha kolay hale getirdięini ve bu sayede programlamanın daha eęlenceli ve ilgi ekici olduęunu belirtmiřtir.

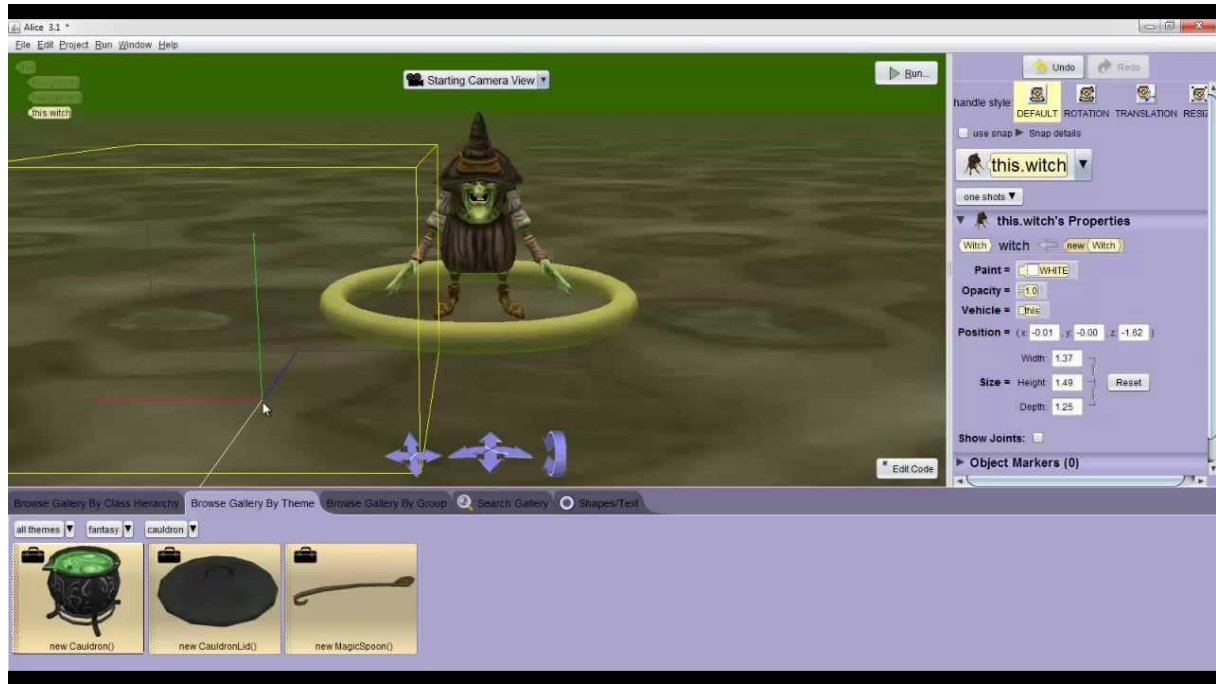
## 2.4. BTP Araları

Programları bir ara yz sayesinde, oęunlukla “srkle - bırak” řeklinde oluřturmayı saęlayan aralardır. BTP araları, programcılara programlama yaparken en ufak hatalardan bile arınık bir řekilde program geliřtirilebilmesine imkn vermektedir. BTP araları ile programlama yaparken yapılan uygulama veya yazılımın anında alıřtırılıp hızlıca n izlemesi yapılabilir ve anında dnt sistemi ile programcının varsa hatasını dzeltmesine imkan saęlamaktadır. Bu tarz BTP araları gnmzde oęalmakta ve farklı teknikler ve kendilerine has zellikleri ile karřımıza ıkmaktadırlar. Bu aralara rnek olarak Alice, Scratch, Codeorg, HackerCan, Blockly ve Kodu Game Lab verilebilir. Bu BTP aralarının ne ıkan en belirgin zellikleri programlamanın bloklar ile yapılmasıdır. Aralarda bulunan her bir blok parasının bir kodu belirtmesi ile birlikte bu blokların birleřtirilmesi sayesinde bir akıř řeması oluřur. Bu birleřimin sonucunda ise bir kod kmesi ortaya ıkmıř olur. (Weintrop ve Wilensky, 2015) Yaptıkları alıřmada BTP aralarında blok kullanımının avantajlarından bahsetmiřtir. ğrencilerin blokları gnlk dile benzer buldukları ve bundan dolayı programlama yaparken kavramları daha rahat anlayabildikleri sonucuna varmıřlardır.

### 2.4.1. Alice

Alice Carnegie Mellon niversitesi ęretim yesi Randy Pausch nclęnde niversite bnyesinde bir proje olarak geliřtirilmiřtir. Alice boyutu byk bir uygulama olması nedeniyle dięer uygulamalardan bu konuda geride kalmakta ancak grsellięi dřnldęnde uygulama boyutunun normal olduęu kanısına varılmaktadır. ğrencilerin animasyon filmi ve basit dzeyde video oyunları oluřturmalarını saęlayan uygulama temel programlamanın ęrenilmesini saęlamaktadır. Alice, geliřtirilen programın hemen alıřtırılmasını saęlayarak ğrencinin kendi yaptıęı programlamadaki iliřkiyi anlamasını ve yaptıęı programlamayı kendine zg hale getirerek bireyselleřtirmesini saęlar. Ayrıca Alice

aracı birkaç dil paketine sahip olmasına rağmen Türkçe dil paketi bulunmamaktadır. (Cooper, Dann, ve Pausch, 2000) Alice aracının bilgisayar programlamaya yeni başlayan bireylerin ilgi çekici animasyonlar ve oyunlar tasarlayabilecekleri bir ortam oluşturmaları amacıyla geliştirildiğini belirtmişleridir.



Resim 2.1. Alice ekran görüntüsü

#### 2.4.2.Scratch

Scratch, Yaşam Boyu Okul Öncesi Grubu (Lifelong Kindergarten Group) tarafından MIT Medya Laboratuvarında geliştirilmiş olan ücretsiz BTP aracıdır. Scratch aracı 2007 yılında 150'den fazla ülkede ve yaklaşık 50 dil paketi ile sunulmuştur. Bu diller içinde Türkçe de bulunmaktadır. Scratch resmi web sitesinden edinilen bilgilere göre çevrimiçi olarak 39.205.979 proje yapılmış ve bu araç yaklaşık 53 yaşındaki bireylere kadar ulaşmıştır (URL-3, 2019).

Programlamayı daha eğlenceli ve görsel hale getirmesinin yanı sıra programlamanın daha kolay ve anlaşılır yapılmasını sağlayan Scratch, programlamaya yeni başlayan bireylerin de animasyonlu hikâyeler, oyunlar, simülasyonlar vb. projeleri kolay bir şekilde oluşturabilmelerini sağlamaktadır (Yükseltürk ve Altıok, 2016). Scratch, özellikle küçük yaştaki öğrenci gruplarını hedef alan ve onların programlama eğitimine ilk girişini sağlamak amacıyla geliştirilmiş olmakla birlikte her yaş grubundan kullanıcıya hitap etmekte olan bir

BTP aracıdır. Farklı bir şekilde ifade edilecek olursa Scratch, programlamayı yeni öğrenenleri programlamanın karmaşık yapısından kurtararak onların daha çok var olan problemin çözümü ve programlamanın temelini oluşturan kavramlar üzerinde yoğunlaşmalarını sağlamaktadır (Kaucic ve Asic, 2011). Scratch sayesinde kullanıcılar üst düzey bir programlama bilgisi gerektirmeden program ara yüzü ve matematiksel – mantıksal becerilerini kullanarak programlama yapabilmektedirler.

Scratch aracında kodlar hazır bloklar halinde bulunmaktadırlar. Dolayısıyla kullanıcıların hata yapma olasılığı bulunmamaktadır. Ayrıca kod bloklarının şekilleri onların birbirleriyle nasıl birleştirileceğini kullanıcıya göstermektedir. Bu sayede kullanıcıların kod bloklarını birleştirirken anlamsız bir birleşim yapmalarına program engel olmakta ve kullanıcılar kod blokları arasındaki ilişkileri daha iyi kavrayabilmektedirler.

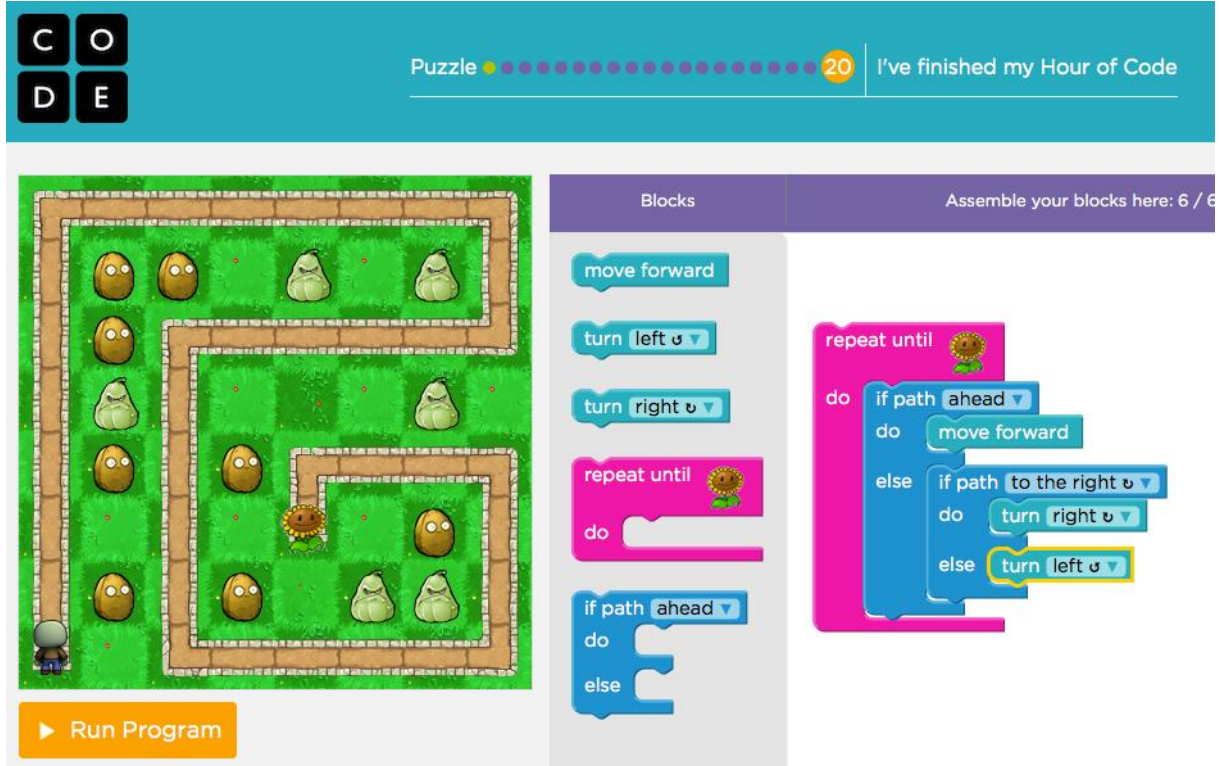


Resim 2.2. Scratch ekran görüntüsü

#### 2.4.3.Code.org

Code.org, Microsoft ve Google gibi büyük şirketler tarafından desteklenen genç yaşta kod okuryazarlığını aşılama amacıyla başlatılan proje olarak karşımıza çıkmaktadır. Hiç programlama bilgisi olmayan biri bile oyun oynayarak kodlama mantığını bu yazılım sayesinde öğrenebilir. Yazılımın ücretsiz olması ve birçok dil desteğine sahip olması öğrenciler için yazılımı cazip hale getirmektedir.

Code.org aslında kod gerektirmeden, kod blokları kullanarak küçük yaştaki bireylere, algoritmik düşünmeyi, bir işi sıraya koymayı, sıralanan işi tekrarlamayı, o işle ilgili fonksiyonlar üretme beceresi kazandırmayı amaçlamaktadır. Bu eğitimin ardından öğrenciler, kendi oyunlarını, animasyonlarını üretebilir. Ayrıca bu becerileri kazanan öğrenciler, diğer derslerinde de bu becerileri kullanarak daha başarılı olurlar (URL-4, 2017).



Resim 2.3. Code.org ekran görüntüsü

#### 2.4.4. HackerCan

Pamukkale Üniversitesi Teknokent bünyesindeki uzman ekip tarafından geliştirilmiş onaylı bir ARGE projesi olan HackerCan çocukların anadilinde kodlamayı öğrenebilmesi için %100 Türkçe olarak sunuluyor. HackerCan öğrencilerin Türkçe dilinde kodlamayı öğrenme ve Türkçe kod yazarak yazılım geliştirme becerisini kazandırmak için hazırlanan Türkiye'nin ilk eğitim platformu özelliğine sahiptir. Bu uygulamada okuma yazmayı bilen her öğrenci, kendi yaş grubu ve seviyesine göre programlama yapabilmektedir.

HackerCan, Talim Terbiye Kurulu tarafından 13 Ocak 2017'de yayınlanan Bilişim Teknolojileri ve Yazılım dersi öğretim programına giren ilk yerli yazılım olma özelliğine sahiptir. Bu yazılım ile bireylerin sayısal zeka, analitik düşünce ve problem çözme

yeteneklerinin geliştirilmesi planlanıyor. Oyun tabanlı öğrenme modeli üzerine kurgulanmış yazılım, macera serisi, çizim ve stüdyo ekranları ile altı ile dokuz yaş ve dokuz yaş üstü öğrencilerin kolaylıkla algoritma becerisi geliştireceği bir yazılım olma özelliği taşımaktadır. Eğitimciler için özel ortamlar, öğretim planları, sunumlar ve akademik materyalleri sunan HackerCan; eğitimlerin sonunda noktalama işaretlerinin ve yabancı karakterlerin en aza indirildiği bir dil öğretir.

Hem Türkçe, hem İngilizce dillerinde kullanılabilen platform öğrencilere algoritma, problem çözme yeteneği ve programlama eğitimi vermeyi hedeflemektedir. Öğrenciler almış oldukları eğitimle, istedikleri dile rahatlıkla adapte olup web ve masaüstü uygulamaları, mobil uygulamalar ve oyun geliştirebilirler (URL-5, 2019).

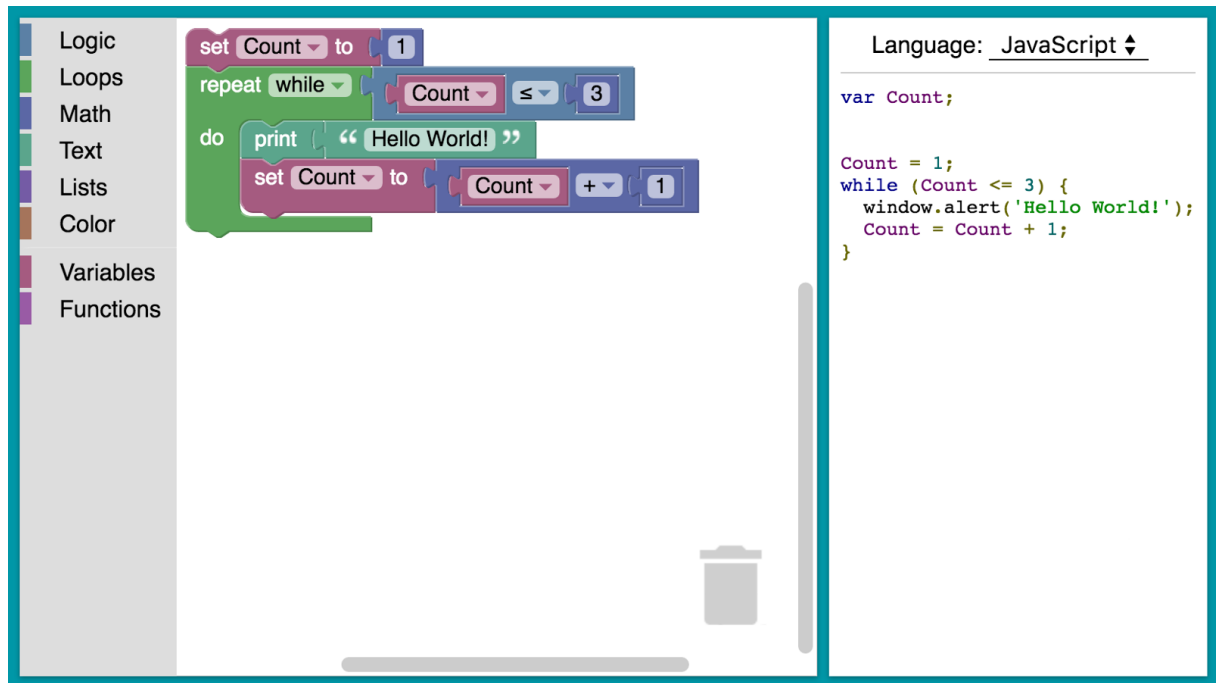


Resim 2.4. HackerCan ekran görüntüsü

### 2.4.5. Blockly

Blockly aracı Google tarafından geliştirilmiş olan, kullanıcıyı söz dizimi ile programlama yapmanın zorluklarından kurtarıp kod blokları vasıtasıyla programlama yapmaya yarayan bir ortamdır. Blockly açık kaynak kodlu bir yazılım olması sebebiyle kullanıcılar, Blockly ortamı üzerinde istedikleri değişiklikleri yapabilmekte ve kendi istekleri doğrultusunda yeni özellikler ekleyebilmektedir. Blockly aracının Scratch ve Alice araçlarından farkı ve en önemli özelliklerinden olan Blockly de tasarlanan kod bloğunun JavaScript, Python, Dart ve

XML dillerinde söz dizimsel olarak yazılan kodların görülebilmesidir. Kullanıcılar görsel olarak hazırladıkları kod blokları sayesinde program hazırlamakla kalmayıp, hazırladıkları programın sözdizimine de erişebilmektedirler. Bu sayede kullanıcılar metin tabanlı programlama dillerine daha hızlı adapte olmakla birlikte BTPyı da daha aktif bir şekilde kullanabilmektedirler. Bu özelliğinden dolayı Blockly diğer BTP araçlarına göre daha gerçekçi bir programlama yapmayı sağlayan bir araç olarak görülmektedir (Mahadevan, Freeman, ve Magerko, 2016).



Resim 2.5. Blockly ekran görüntüsü

#### 2.4.6.Kodu Game Lab

Kodu Game Lab oyun tasarlamak amacıyla düşünülmüş basit ve görsel bir programlama dilidir. Genel olarak çocuklar ve programlamayı eğlenerek öğrenmek isteyen herkes için tasarlanmıştır. Programlama kullanıcı arayüzü sayesinde gerçekleştirilmektedir. Nesne tabanlı ve basit bir arayüz bulunmaktadır. Arayüz sayesinde sayfalar arasında geçiş, geniş bir yardım menüsü ve diğer özelliklerin tanımı bulunmaktadır. Türkçe dil desteği bulunan Kodu Game Lab her yaş gurubuna uygun olmakla birlikte bilgisayar programlama mantığını kavratmayı amaçlamaktadır. Kodu Game Lab, teknik ayrıntılardan daha çok programlama mantığını öğretmek için kullanılabilir (URL-6, 2017).



Kodu Game Lab’da kullanılan programlama dili semantik bir yapıdadır; isim, fiil ve sıfat üzerine kurulu olan dil, “topa yaklaşınca yok et” gibi basit düzeyde cümleler kurarak kullanıcıların olaylar kurgulayabilmesini sağlıyor. Kodu Game Lab’ı diğer araçlardan ayıran özelliği görsel ara yüzü ve semantik yapıya dayalı programlama ilkelerine uyumlu olmasıdır.



Resim 2.6. Kodu Game Lab ekran görüntüsü

## 2.5. İlgili Çalışmalar

Literatürde BTP ile ilgili birçok çalışma bulunmaktadır özellikle son yıllarda bu çalışmaların sayısı artmıştır. Taranan çalışmalarla BTP aracı kullanmanın etkileri incelenmiştir. (Saltan, 2017), Yaptığı çalışmada çevrimiçi algoritma görselleştirmelerinin programlama dersine girişteki başarılarına etkisini araştırmıştır. Çalışma grubunu 40 Bilişim Teknoloji öğrencisi oluşturmaktadır ve çalışma dört hafta sürmüştür. Nitel ve nicel veri toplama yöntemi kullanılan araştırmada öğrenciler çevrimiçi algoritma görselleştirmenin programlama dersini daha ilgi çekici hale getirdiğini algoritmik düşünme becerisini geliştirdiğini belirtmişlerdir.

(Portelance, Strawhacker, ve Bers, 2016), Anaokulunda yaptıkları çalışmada 62 öğrencinin katılımı ile altı haftalık bir Scratch eğitimi verilmiştir. Bu eğitimde temel



programlama eğitimini Scratch aracı ile öğrenen öğrenciler blok kodlama ile programlama öğrenmişlerdir. Katılımcıların yaşlarından dolayı programlama eğitiminde zorlandıkları görülmekle birlikte Scratch aracının ilgi çekici ve eğlenceli olduğu sonucuna varılmıştır.

(Du, Wimmer, ve Rada, 2016), “Kod Saati” aracını kullanarak yaptıkları çalışmada öğrencilerin programlamaya karşı tutumlarını ve programlama başarılarını incelemişlerdir. Code.org altyapısını kullanan “kod saati” aracı ile yapılan çalışma üniversite öğrencileri ile yürütülmüştür. Çalışma sonunda elde edilen bulgulara göre öğrencilerin programlamaya yönelik tutumları önemli ölçüde olumlu etkilenirken programlama başarıları aynı ölçüde etkilenmemiştir.

(Patan, 2016), yaptığı yüksek lisans tezinde , “Kodlama Saati” programının tasarlanması, geliştirilmesi ve değerlendirilmesi konusunun araştırılması ve okul öncesi düzeyindeki dört ve beş yaş çocuklarının bilişimsel düşünme becerilerinin geliştirilmesi amaçlanmıştır. Proje 2013-2014 yılında geliştirilmiş ve 2014-2015 eğitim öğretim yılında uygulanmıştır. Bilişimsel düşünmenin geliştirilmesi ve eğitsel oyun gibi çeşitli etkinlikler ile Bilişim Teknolojileri dersinde ipad uygulamalarının kullanımı gerçekleştirilmiştir. Proje de kodable, code.org, Flipboom ve Srtachjr araçları kullanılmıştır. Elde edilen bulgulara göre öğrencilerin uygulamada başarılı oldukları ancak bazı döngü komut seviyelerinde zorlandıkları görülmüştür.

(Yiğit, 2016), yaptığı yüksek lisans tezinde, BTP araçları ile programlama öğretiminin öğrencilerin programlama başarılarına ve programlamaya karşı tutumlarına etkisini incelemiştir. Çalışma 2015-2016 Eğitim-Öğretim yılı güz döneminde bir üniversitenin Bilgisayar ve Öğretim Teknolojileri Eğitimi Bölümü ikinci sınıf öğrencileri üzerinde Sekiz hafta boyunca yürütülmüştür. Deney grubundaki öğrencilere Blockly aracı ile kontrol grubundaki öğrencilere ise geleneksel olarak kabul edilen Python aracı ile eğitim verilmiştir. Araştırma sonucunda BTP eğitimi alan öğrencilerin geleneksel programlama eğitimi alan öğrencilere göre programlamayı öğrenmede daha başarılı oldukları sonucuna varılmıştır.

(Papatğa, 2016), Doktora tezi kapsamında yaptığı çalışmada Scratch aracı ile öğrencilerin okuduklarını anlama becerilerinin gelişme düzeylerini incelemiştir. Araştırma 2015-2016 öğretim yılı güz döneminde bir ilkokulda 15 hafta boyunca gerçekleştirilmiştir. Çalışma nitel ve nicel verilerle değerlendirilmiştir. Araştırma sonucunda Scratch aracını kullanan öğrencilerin okuduğunu anlama becerilerinin geliştiği gözlenmiştir.

(Yükseltürk ve Altıok, 2016), Yaptıkları bu çalışmada Bilişim Teknolojileri öğretmen adaylarının Scratch aracı kullanımlarına yönelik algılarının incelemişlerdir. Çalışma 2013.-2014 öğretim yılında 159 öğretmen adayı ile gerçekleştirilmiştir. Çalışma karma desen

olarak düzenlenmiş ve çalışma sonucunda Bilişim Teknolojileri öğretmen adaylarının Scratch aracı ile programlama yapmaları motivasyonlarının artmasına ve kullanım kolaylığı konusundaki algılarının olumlu yönde etkilendiği belirlenmiştir.

(Saltan, 2016), Algoritma görselleştirmesi ile yaptığı çalışmada öğretmen adaylarının öğrenme ve motivasyonlarına katkısı bakımından Algoritma Görselleştirme algılarını ve öğretmenlik seviyelerinin uygunluğunu incelemiştir. Çalışma 218 üniversite öğrencisi ile yürütülmüştür. Araştırmada nitel ve nicel yöntemler kullanılmıştır. Araştırma sonucunda öğretmen adayları algoritma görselleştirmenin, öğrenmeyi daha eğlenceli hale getirdiği, motivasyonu arttırdığı ve algoritmik düşünmeyi sağladığını belirtmişlerdir.

(Topal, 2015), Yaptığı yüksek lisans tezinde öğrencilerin problem çözme becerilerinden olan mantık yürütme ve matematiksel düşünme becerilerini incelemiştir. Bu bağlamda, 2013 - 2014 eğitim öğretim yılında Türkiye'nin güneyinde yer alan bir ildeki 260 ortaokul 6. sınıf öğrencisi ile yürütülen çalışmada öğrencilerin algoritma becerileri ölçülmüştür. Nitel ve nicel yöntemler kullanılan araştırmada öğrencilerin standart bir algoritma kullanarak çözebildikleri ve çözemedikleri soruların başarı ortalamalarına bakılmıştır. Elde edilen bulgulara göre öğrenciler standart bir algoritmaya dayalı problemlerde daha başarılı olmalarına rağmen problem çözmeye genel olarak elde edilmek istenen sonuca ulaşamamışlardır.

(Doğan, 2015), Bu araştırmada bilgisayar oyunu geliştirmenin eleştirel düşünme becerilerine ve algoritma başarılarına etkisini belirlemeyi amaçlamıştır. Araştırma bir ortaokulun 6. sınıflarında öğrenim gören 54 öğrenci ile altı hafta boyunca devam ettirilmiştir. Bilişim teknolojileri ve yazılım dersi kapsamında yapılan çalışma algoritma konusunda deney ve kontrol grupları oluşturulmuş ve farklı yöntemlerle ders işlenmiştir. Deney grubunda oyun geliştirme konusu işlenirken kontrol grubunda geleneksel yöntemle ders işlenmiştir. Araştırmada nitel ve nicel verilerden yararlanılmış ve edinilen bulgulara göre deney grubundaki öğrencilerin eleştirel düşünme becerilerinin ve algoritma başarılarının olumlu düzeyde etkilendiği görülmüştür.

(Erol, 2015), yaptığı doktora tezi çalışmasında, Scratch ile programlama öğretiminin öğrencilerin motivasyon ve programlama başarısına etkisini incelemeyi amaçlamıştır. Bu çalışmanın örneklemini Mehmet Akif Ersoy Üniversitesi, Eğitim Fakültesi, Bilgisayar ve Öğretim Teknolojileri bölümünde eğitim gören 52 ikinci sınıf öğrencisi oluşturmuştur ve çalışma 14 hafta devam etmiştir. Çalışma iki bölümde yapılmış ve ilk yedi haftalık süreçte kontrol grubuna geleneksel programlama eğitimi verilmiş, deney grubuna ise Scratch programı ile programlama eğitimi verilmiştir. Uygulamanın ikinci yedi haftalık bölümünde ise hem kontrol hem de deney grubuna C# programlama dili eğitimi verilmiştir her iki grupta

aynı yöntemle ders işlemiştir. Araştırmanın sonucunda ise öğrencilerin motivasyonlarında ve başarılarında deney grubu lehine anlamlı bir farklılık ortaya çıkmıştır. Çalışmanın nitel verilerine bakıldığında ise kontrol grubundaki öğrencilerin programlama eğitimi sürecini zor ve sıkıcı buldukları, uygulamada aktif rol alamamaları gibi etkenlerin motivasyonlarını düşürdüğünü belirtmişlerdir. Deney gurubundaki katılımcıların ise Scratch ile programlama eğitiminin eğlenceli ve kolay olduğunu bu sayede motivasyon ve başarılarının artış gösterdiğini belirtmişlerdir.

(Çatlak, Tekdal, ve Baz, 2015), yaptıkları doküman inceleme çalışmasında Scratch aracının programlama öğretiminde kullanımını incelemişlerdir. Bu çalışmada farklı veri tabanları üzerinde tarama yapılmış ve 32 makale incelenmiştir. Bu makalelerden elde edilen bulgulara göre Scratch aracının programlama öğretimini zevkli ve daha anlaşılır hale getirdiği ve programlama eğitiminde daha etkili olduğu görülmüştür. Scratch aracının görselliği ön planda tutmasından dolayı özellikle ilköğretim okullarında öğretim programına eklenmesi gerekliliği vurgulanmıştır

(Olgun, 2014) yaptığı yüksek lisans tezinde, ortaokul öğrencileri ile ilgili bir çalışma gerçekleştirmiştir bu kapsamda programlama eğitimi almamış olan 6. sınıf öğrencilerinin düşünme stillerine programlamanın etkisi araştırılmıştır. Araştırmada Düşünme Stilleri Envanteri kullanılmış programlama eğitimi ile ilgili yurt içi ve yurt dışı çalışmalar incelenmiş ve öğrenciler üzerinde farklı etkilerinin olduğu belirtilmiştir. Scratch BTP aracı ile yapılan çalışmada öğrencilerin düşünme stillerinde gelişim gösterdikleri görülmüştür.

## III. BÖLÜM

### 3.YÖNTEM

Bu bölümde araştırma modeli, araştırma grubu, veri toplama araçları ve verilerin değerlendirilmesi konuları yer almaktadır.

#### 3.1.Araştırma Modeli

Çalışmada nicel araştırma yöntemlerinden ön test-son test kontrol gruplu yarı deneysel desen kullanılmıştır. Bu desende bağımlı değişken ile ilgili ölçümler hem deneysel çalışmanın öncesinde hem de sonrasında uygulanır ve değişkenler arasında oluşan neden sonuç ilişkileri test edilir (Karasar, 2005).

Katılımcılara uygulamadan önce algoritma başarı ön testi ve programlamaya karşı kaygı ön testi ve programlamaya karşı tutum ön testi uygulanmıştır. Kontrol grubuna Scratch BTP aracı ile deney grubuna ise HackerCan BTP aracı ile dört haftalık programlama eğitimi verilmiştir. Eğitim sonunda her iki gruba da algoritma başarı son testi ve programlamaya karşı kaygı son testi ve programlamaya karşı tutum son testi uygulanmıştır.

#### 3.2.Çalışma Grubu

Araştırmanın çalışma grubunu Orta Karadeniz'de bir üniversitenin Bilgisayar ve Öğretim Teknolojileri Eğitimi bölümünde öğrenim gören 43 2. Sınıf öğrencisi oluşturmaktadır. Çalışma grubunda yer alan öğrenciler yansız atama ile iki gruba ayrılmıştır. Kontrol grubu öğrencileri 11 kadın ve 10 erkek olmak üzere toplam 21 kişi. Deney grubu ise 10 kadın 12 erkek toplam 22 öğrenciden oluşmaktadır. Çalışmanın 2. Sınıflara yapılmasının sebebi bu sınıftaki bireylerin henüz BTP araçları ile ilgili eğitim almamış olmalarıdır.

Tablo 3.1. Çalışma Grubu Dağılımı Tablosu

Gruplar	Kadın	Erkek	Toplam
Deney	10	12	22
Kontrol	11	10	21
Toplam	21	22	43

### 3.3. Veri Toplama Araçları

Araştırmanın başında ve sonunda öğrencilere algoritma becerilerini ölçen algoritma başarı testi, programlamaya karşı kaygı testi ve programlamaya karşı tutum ölçeği uygulanmıştır.

#### 3.3.1. Algoritma Başarı Testi

Algoritma konusundaki başarıları ölçmek amacıyla araştırmacı tarafından geliştirilen "Algoritma Başarı Testi" ile veriler toplanmıştır. Çoktan seçmeli 20 soru bulunan testte doğru yanıtlara bir puan yanlış yanıtlara ve boş bırakılan sorulara sıfır puan verilmiştir. Testten alınabilecek en yüksek puan 20'dir.

Algoritma başarı testinin geliştirilmesi aşamasında ilk olarak algoritma konusunda yer alan altı kazanıma yönelik 36 soru oluşturulmuştur. Uzman görüşleri alınarak testin geçerliliği artırılmaya çalışılmıştır. Bu doğrultuda oluşturulan başarı testinin pilot çalışması daha önce algoritma dersi almış Bilgisayar ve Öğretim Teknolojileri Öğretmenliği bölümünde olan 3. Sınıf, 4. Sınıf ve mezunları olmak üzere toplam 105 kişiye uygulanmıştır. Yapılan pilot uygulamada elde edilen verilerin analizleri sonucunda ayırt edicilikleri düşük olan ve güçlük indeksleri çok yüksek veya çok düşük olan sorular testten çıkarılarak soru sayısı 20'ye düşürülmüştür. Son halini alan testin KR-20 güvenirlik katsayısı 0,84 olarak bulunmuştur.

Algoritma Başarı Testinde bulunan soruların değerlendirmede ele alınan 6 kazanım alanına yönelik dağılımları Tablo 3.2' de gösterilmektedir.

Tablo 3.2. Kazanımların soru dağılımları

Temel kavramlar hakkında bilgi sahibidir.	1,2,3,4,5,7,8,15,19
Değişken yapısını çözümler.	1,6,7,11,12,14,15,17,19
Döngü yapısını günlük hayattaki problemlerde kullanır.	5,10,11,12,13,20
İşlem Sırasını takip ederek sonuca ulaşır.	11,13,14,16,17,18,20
Mantıksal – Matematiksel problemleri ayırt eder.	3,7,9,10,11,12,13,14,15,16,18,19
Koşul yapısını kavrayarak günlük hayata aktarır.	8,10,11,12,14,17

### 3.3.2.Bilgisayar Programlama Kaygı Ölçeği

Öğrencilerin programlamaya karşı kaygılarını ölçmek amacıyla (Demir, 2015) tarafından Türkçeye uyarlanan ilk olarak (Cheung, Koon, Cheng, ve Choo, 1990) tarafından geliştirilen “bilgisayar programlama kaygı ölçeği”, “computer programming anxiety scale” kullanılmıştır. Ölçek 19 maddeden oluşmakta ve her bir madde katılımcılar tarafından “Hiçbir zaman”, “Nadiren”, “Bazen”, “Sıklıkla” ve “Her zaman” şeklinde değerlendirilmektedir. Ölçek tek faktörlü olmakla birlikte Cronbach-Alpha güvenilirlik katsayısı 0,93 olarak bulunmuştur. Bu ölçüme göre ölçeğin oldukça güvenilir olduğu görülmektedir. Ölçeğin geçerlik katsayısı Kaiser-Meyer-Olkin (KMO) katsayısı ve Sphericity testi aracılığı ile ölçülmüştür. KMO katsayısının 0,80’den büyük olması faktörü analizinin iyi olması anlamına gelmektedir. Ölçeğin KMO katsayısı 0,911 düzeyinde bulunmuştur.

### 3.3.3.Bilgisayar Programlamaya Karşı Tutum Ölçeği

Öğrencilerin programlamaya karşı tutumlarının ölçülmesi amacıyla uygulama öncesinde ve sonrasında (Başer, 2013) tarafından geliştirilen Bilgisayar Programlamaya Karşı Tutum Ölçeği kullanılmıştır.

Ölçekte 38 madde bulunmaktadır. Her bir madde katılımcılar tarafından "Kesinlikle Katılmıyorum", "Katılmıyorum", "Karasızım", "Katılıyorum" ve "Kesinlikle Katılıyorum" şeklinde değerlendirilmektedir. Yapılan analizlerde ölçek maddelerinin "Programlamada kendine güven ve güdülenme", "Programlamanın faydası", "Programlamada başarıya karşı tutum" ve "Programlamada başarının sosyal algısı" olarak 4 boyut haline toplanmıştır. Alt boyutların Cronbach-Alpha güvenilirlik katsayıları sırasıyla 0,944, 0,920, 0,926, 0,618 olup

ölçeğin tamamının Cronbach-Alpha güvenilirlik katsayısı ise 0,953 olup güvenilir bir ölçek olarak değerlendirilebilmektedir (Hair, Black, Babin, ve Anderson, 2010).

### 3.4. Uygulama Süreci

Uygulama aşamasında öğrenciler iki gruba ayrıldı kontrol grubunda Scratch uygulaması kullanılırken deney grubunda HackerCan uygulaması kullanıldı. Uygulama süreci dört hafta her iki gruba ayrı olmak üzere yürütüldü.

Bir Hafta her iki grupta da kendi uygulamalarını kullanarak “Algoritma ve Programlamaya Giriş” konusu işlendi. Algoritmanın tanımı, akış şeması ifadesi ve sembolleri anlatıldı ve örnek uygulamalar yapıldı.

2. Hafta “Değişken yapıları” ile ilgili eğitimler verildi ve örnek uygulamalar yapıldı. Kontrol grubunda Scratch ile değişken oluşturma ve kullanma kazanımlarına yönelik basit bir hesap makinesi etkinliği gerçekleştirildi. Bu uygulamada iki değişken tanımlanmış ve “sayi1”, “sayi2” olarak tanımlanmış ardından bu değişkenler arasında toplama ve çıkarma gibi işlemler yapılmıştır. Deney grubunda ise HackerCan ile değişken uygulaması yapılmış ve karakterin hedefe ulaşması için gitmesi gereken yollar ve nasıl gitmesi gerektiğini belirten değişkenler uygulanarak gerçekleştirilmiştir.

3. Hafta “Koşul yapıları” ile ilgili eğitimler verildi ve örnek uygulamalar yapıldı. Kontrol grubunda Scratch ile koşul yapısı etkinliği gerçekleştirilmiştir. Bu etkinlikte uygulamadan bir sayı üretmesini ve kullanıcının aklından tutacağı sayının uygulama tarafından tutulan sayı ile eşleşip eşleşmediği durumuna bakılmıştır. Eğer kullanıcının tuttuğu sayı büyük veya küçük ise uygulama kullanıcıyı yönlendirerek doğru tahmini yapmasını sağlamaktadır. Deney grubunda ise HackerCan ile koşul yapısı uygulaması yapılmış ve burada ilk bölümden itibaren karşımıza çıkan koşullar uygulandı ve karakter bir bölgeye uğramadan çıkış bölümüne geçemediği için koşulu sağlaması gerekiyordu ve bu sayede koşullar gerçekleştirilmiştir.

4. Hafta “döngü yapıları” ile ilgili eğitimler verildi ve örnek uygulamalar yapıldı. Kontrol grubunda Scratch kullanılarak döngü yapısını kavramak amacı ile bir sayaç eklenmiş ve sayacın döngü yapısı ile aktif hale getirilip çalıştırılması sağlanmıştır. Deney grubunda ise HackerCan ile döngü yapısı uygulaması yapılmış ve burada 14. bölümden itibaren döngü yapılarının olduğu bölümler gelmekte ve karakterin sürekli aynı adımı tekrar etmesi ile döngü sağlanmış olur.

Tablo 3.3. veri toplama ve uygulama aşamaları

<b>1. Hafta</b>	Ön test ve ölçeklerin uygulanması
<b>2. Hafta</b>	Algoritma ve programlamaya giriş
<b>3. Hafta</b>	Değişken yapıları ile ilgili eğitim ve örneklerin verilmesi
<b>4. Hafta</b>	Koşul yapıları ile ilgili eğitim ve örneklerin verilmesi
<b>5. Hafta</b>	Döngü yapıları ile ilgili eğitim ve örneklerin verilmesi
<b>6. Hafta</b>	Son test ve ölçeklerin uygulanması
<b>7. Hafta</b>	Gruplara eğitimi verilen uygulamaların değiştirilerek tanıtılması

Kontrol grubu ile yapılan uygulamalar bilgisayarlara kurulu olan Scratch aracı ile gerçekleştirilmiş olup her kazanıma uygun ve bütünlüyci bir şekilde uygulanmıştır. Deney grubu ile gerçekleştirilen uygulamalar HackerCan web sitesinde çevrimiçi olarak sürdürülmüş ve uygulamadaki seviyeler kazanım düzeylerine erişinceye kadar devam ettirilmiştir. Bu uygulamada kullanıcıların hata yapması sonucunda uygulama dönüt vererek kullanıcının yanlışını bulmasını sağlamaktadır. Ayrıca web sitesinde sınıf oluşturulmuş ve her öğrencinin hangi düzeyde olduğu eğitmen tarafından gözlenmiştir.

### 3.5. Veri Analizi

Çalışmada nicel veriler toplanmıştır. Nicel veriler algoritma başarı testi, programlamaya karşı tutum ve programlamaya karşı kaygı ölçekleri ile elde edilmiştir. Ölçeklerden toplanan verilerin analizi SPSS 20 ile yapılmıştır. Ölçekler eğitimden önce ve sonra ön test - son test olarak uygulanmıştır. Grupların ön test puanlarına bağımsız örneklem t-testi ile bakılarak farklılık olup olmadığına bakılmıştır. Çalışma sonunda son testler uygulanarak gruplar arasındaki farklar bağımsız örneklem t-testi ile analiz edilmiştir. Grupların kendi içindeki değişime bakmak için ise ilişkili örneklem t-testi ile analizler yapılmıştır.



## IV. BÖLÜM

### 4.BULGULAR

#### 4.1.Farklı Blok Tabanlı Programlama Aracı Kullanılmasının Öğrencilerin Akademik Başarılarına Etkisine İlişkin Bulgular

Programlama eğitiminde farklı yapılardaki BTP araçlarının akademik başarıya etkisini belirlemek amacıyla öğrencilerin ön test başarı puanları arasında farklılık olup olmadığı incelenmiştir. Ön test ve Son test puanlarına bakılarak yapılacak analizleri belirlemek amacıyla ön test ve son test başarı puanlarının normal dağılım gösterip göstermediği Shapiro-Wilks testi ile incelenmiştir. Elde edilen bulgular Tablo 4.1’de verilmiştir.

Tablo 4.1. Akademik Başarı Ön test ve Son test Puanlarının Shapiro-Wilks Testi Sonuçları

Testler	Gruplar	Shapiro-Wilks	P
Ön Test	Deney	0,93	0,18
Ön Test	Kontrol	0,95	0,45
Son Test	Deney	0,94	0,20
Son Test	Kontrol	0,96	0,54

Tablo 4.1 incelendiğinde Deney ön test ve son test başarı puanları ile Kontrol ön test ve son test başarı puanlarının normal dağılım özelliği gösterdiği görülmektedir ( $p>0,05$ ) (Büyüköztürk, 2016).

Deney ve Kontrol ön test başarı puanları arasında anlamlı farklılık olup olmadığı puanların normal dağılım özelliği göstermesinden dolayı Bağımsız Örneklem t-testi (Independent Samples t-test) ile analizleri yapılmıştır.

Tablo 4.2. Deney ve Kontrol Grubu Akademik Başarı Ön test Puanları Bağımsız Örneklem t-Testi Sonuçları

	Grup	N	$\bar{X}$	S	sd	t	P
Akademik Başarı Ön test	Deney	22	14,54	2,28	41	0,75	0,45
	Kontrol	21	14,00	2,42			

Analiz sonuçlarına göre deney grubunun ön test başarı puanlarının ( $\bar{X}=14,54$ ) kontrol grubunun ön test başarı puanlarından ( $\bar{X}=14,00$ ) yüksek olduğu görülmektedir. Tablo 4.2 incelendiğinde deney ve kontrol grubu akademik başarı ön test puanları arasında anlamlı bir farklılık olmadığı ve böylelikle grupların denk olduğu görülmektedir  $t(41)=0,75$ ,  $p>0,05$ .

Tablo 4.3. Deney ve Kontrol Grubu Akademik Başarı Son test Puanları Bağımsız Örneklem t-Testi Sonuçları

	Grup	N	$\bar{X}$	S	sd	t	P
Akademik Başarı Son test	Deney	22	15,40	2,51	41	0,50	0,62
	Kontrol	21	15,76	2,07			

Deney ve Kontrol son test başarı puanları arasında anlamlı farklılık olup olmadığı puanların normal dağılım özelliği göstermesinden dolayı Bağımsız Örneklem t-testi ile analizleri yapılmıştır. Analiz sonuçlarına göre kontrol grubunun son test başarı puanları ( $\bar{X}=15,76$ ) deney grubunun ön test başarı puanlarından ( $\bar{X}=15,40$ ) yüksek olduğu görülmektedir. Tablo 4.3 incelendiğinde deney ve kontrol grubu akademik başarı son test puanları arasında anlamlı bir farklılık olmadığı görülmektedir  $t(41)=0,50$ ,  $p>0,05$ .

Tablo 4.4. Deney Grubu Ön test - Son test Puanları İlişkili Örneklem t-Testi Sonuçları

	Başarı Testi	N	$\bar{X}$	S	sd	t	P
Deney	Ön test	22	14,54	2,28	21	2,27	0,34
	Son test	22	15,40	2,51			

Deney grubundaki öğrencilerin ön test ve son test puanları arasındaki farkın anlamlı olup olmadığının incelenmesi amacıyla İlişkili Örneklem T-testi(Paired Samples t-test) sonuçlarına bakılmıştır.

Tablo 4.4 incelendiğinde Deney grubundaki öğrencilerin akademik başarı ön test ve son test puanları arasında anlamlı farklılık olmadığı görülmektedir  $t(21)=2,27$ ,  $p>0,05$ . Deney grubundaki öğrencilerin uygulama öncesi başarı puanlarının ortalaması ( $\bar{X}=14,54$ ), uygulama sonrası başarı puanlarının ortalamasından ( $\bar{X}=15,40$ ) yüksek olduğu görülmektedir.

Tablo 4.5. Kontrol Grubu Ön test - Son test Puanları İlişkili Örneklem t-Testi Sonuçları

	Başarı Testi	N	$\bar{X}$	S	sd	t	p
Kontrol	Ön test	21	14,00	2,42	20	3,01	0,07
	Son test	21	15,76	2,07			

Kontrol grubundaki öğrencilerin ön test ve son test puanları arasındaki farkın anlamlı olup olmadığının incelenmesi amacıyla İlişkili Örneklem T-testi sonuçlarına bakılmıştır.

Tablo 4.5 incelendiğinde Kontrol grubundaki öğrencilerin akademik başarı ön test ve son test puanları arasında anlamlı farklılık olduğu görülmektedir  $t(20)=3,01$ ,  $p<0,05$ . Kontrol grubundaki öğrencilerin uygulama öncesi başarı puanlarının ortalaması ( $\bar{X}=14,00$ ), uygulama sonrası başarı puanlarının ortalamasından ( $\bar{X}=15,76$ ) yüksek olduğu görülmektedir.

Tablo 4.6. Deney Grubunun Cinsiyete göre Akademik Başarı Son test Puanları Bağımsız Örneklem t-Testi Sonuçları

	Cinsiyet	N	$\bar{X}$	S	sd	t	p
Akademik Başarı Son test	Kadın	10	14,90	1,66	20	0,86	0,40
	Erkek	12	15,83	3,06			

Kadın ve erkek son test başarı puanları arasında anlamlı farklılık olup olmadığı puanların normal dağılım özelliği göstermesinden dolayı Bağımsız Örneklem t-testi ile analizleri yapılmıştır. Analiz sonuçlarına göre erkeklerin son test başarı puanları ( $\bar{X}=15,83$ ) kadınların son test başarı puanlarından ( $\bar{X}=14,90$ ) yüksek olduğu görülmektedir. Bu farkın anlamlı olup olmadığı ise bağımsız örneklem t-testi sonucu ile anlaşılır. Tablo 4.6 incelendiğinde kadın ve erkek akademik başarı son test puanları arasında anlamlı bir farklılık olmadığı görülmektedir  $t(20)=0,86$ ,  $p>0,05$ .

Tablo 4.7. Deney Grubunun Cinsiyete göre Akademik Başarı Son test Puanları Bağımsız Örneklem t-Testi Sonuçları

	Cinsiyet	N	$\bar{X}$	S	sd	t	p
Akademik Başarı Son test	Kadın	10	15,30	2,00	19	0,97	0,34
	Erkek	11	16,18	2,13			

Kadın ve erkek son test başarı puanları arasında anlamlı farklılık olup olmadığı puanların normal dağılım özelliği göstermesinden dolayı Bağımsız Örneklem t-testi ile analizleri yapılmıştır. Analiz sonuçlarına göre erkeklerin son test başarı puanları ( $\bar{X}=16,18$ ) kadınların son test başarı puanlarından ( $\bar{X}=15,30$ ) yüksek olduğu görülmektedir. Bu farkın anlamlı olup olmadığı ise bağımsız örneklem t testi sonucu ile anlaşılır. Tablo 4.7 incelendiğinde kadın ve erkek akademik başarı son test puanları arasında anlamlı bir farklılık olmadığı görülmektedir ( $t(19)=0,97, p>0,05$ ).

#### 4.2. Farklı Blok Tabanlı Programlama Aracı Kullanımının Öğrencilerin Bilgisayar Programlamaya Karşı Tutumlarına Etkisine İlişkin Bulgular

Öğrencilerin programlamaya karşı tutumları incelenmiş ve farklılık olup olmadığına bakılmıştır. Programlamaya karşı tutum ön test ve son test puanlarına bakılarak yapılacak analizleri belirlemek amacıyla ön test ve son test tutum puanlarının normal dağılım gösterip göstermediği Shapiro-Wilks testi ile incelenmiştir. Elde edilen bulgular Tablo 4.8'de verilmiştir.

Tablo 4.8. Programlamaya Karşı Tutum Ön test ve Son test Puanlarının Shapiro-Wilks Testi Sonuçları

Testler	Gruplar	Shapiro-Wilks	p
Ön Test	Deney	,95	,39
Ön Test	Kontrol	,98	,98
Son Test	Deney	,98	,96
Son Test	Kontrol	,96	,54

Tablo 4.8 incelendiğinde deney grubu ön test ve son test tutum puanları ile kontrol grubu ön test ve son test tutum puanlarının normal dağılım özelliği gösterdiği görülmektedir ( $p>0,05$ ). Deney ve Kontrol ön test ve son test tutum puanları arasında anlamlı farklılık olup olmadığı puanların normal dağılım özelliği göstermesinden dolayı Bağımsız Örneklem t-testi ile analizleri yapılmıştır.

Tablo 4.9. Deney ve Kontrol Grubu Bilgisayar Programlamaya Karşı tutum Ön test Puanları Bağımsız Örneklem t-Testi Sonuçları

	Grup	N	$\bar{X}$	S	sd	t	p
Programlamaya Karşı Tutum Ön test	Deney	22	3,80	0,44	41	1,07	0,28
	Kontrol	21	3,95	0,48			

Deney ve Kontrol grubu ön test tutum puanları arasında anlamlı farklılık olup olmadığı puanların normal dağılım özelliği göstermesinden dolayı Bağımsız Örneklem t-testi ile analizleri yapılmıştır. Analiz sonuçlarına göre deney grubu ön test tutum puanlarının ( $\bar{X}=3,80$ ) kontrol grubunun ön test tutum puanlarından ( $\bar{X}=3,95$ ) düşük olduğu görülmektedir. Buna karşın Tablo 4.9 incelendiğinde deney ve kontrol gruplarının programlamaya karşı tutum ön test puanları arasında anlamlı bir farklılık olmadığı görülmektedir  $t(41)=1,07$ ,  $p>0,05$ .

Tablo 4.10. Deney ve Kontrol Grubu Bilgisayar Programlamaya Karşı tutum Son test Puanları Bağımsız Örneklem t-Testi Sonuçları

	Grup	N	$\bar{X}$	S	sd	t	p
Programlamaya Karşı Tutum Son test	Deney	22	3,91	0,44	41	0,33	0,74
	Kontrol	21	3,85	0,72			

Deney ve Kontrol grubu son test tutum puanları arasında anlamlı farklılık olup olmadığını Bağımsız Örneklem t-testi ile analizleri yapılmıştır. Analiz sonuçlarına göre deney grubunun son test tutum puanları ( $\bar{X}=3,91$ ) kontrol grubunun son test tutum puanlarından ( $\bar{X}=3,85$ ) yüksek olmasına rağmen Tablo 4.10 incelendiğinde, deney ve kontrol grubu programlamaya karşı tutum son test puanları arasında anlamlı bir farklılık olmadığı görülmektedir  $t(41)=0,33$ ,  $p>0,05$ .

Deney ve Kontrol grubu Bilgisayar Programlamaya Karşı tutum Kendine güven ve güdülenme alt boyutu ön test puanları arasında anlamlı farklılık olup olmadığı puanların normal dağılım özelliği göstermesinden dolayı Bağımsız Örneklem t-testi ile analizleri yapılmıştır.

Tablo 4.11. Deney ve Kontrol Grubunun Bilgisayar Programlamaya Karşı tutum alt boyutları Ön test Puanları Bağımsız Örneklem t-Testi Sonuçları

	Grup	N	$\bar{X}$	S	sd	t	p
Kendine güven ve güdülenme	Deney	22	3,55	0,52	41	0,00	0,99
	Kontrol	21	3,55	0,60			
Programlamanın Faydası	Deney	22	3,97	0,40	41	1,10	0,27
	Kontrol	21	4,17	0,70			
Programlamada başarıya karşı tutum	Deney	22	3,97	0,78	41	1,63	0,10
	Kontrol	21	4,34	0,70			
Programlamada Başarının Sosyal Algısı	Deney	22	4,21	0,69	41	1,47	0,14
	Kontrol	21	4,49	0,53			

Analiz sonuçlarına göre iki grubunda ön test Kendine güven ve güdülenme alt boyutu puanlarının ( $\bar{X}=3,55$ ) aynı olduğu görülmektedir. Tablo 4.11 incelendiğinde Deney ve kontrol grubunun programlamaya karşı tutum kendine güven ve güdülenme alt boyutu ön test puanları arasında anlamlı bir farklılık olmadığı görülmektedir  $t(41)=0,00$ ,  $p>0,05$ .

Deney ve Kontrol grubu Bilgisayar Programlamaya Karşı tutum Programlamanın Faydası alt boyutu ön test puanları arasında anlamlı farklılık olup olmadığı puanların normal dağılım özelliği göstermesinden dolayı Bağımsız Örneklem t-testi ile analizleri yapılmıştır. Analiz sonuçlarına göre kontrol grubunun ön test Kendine güven ve güdülenme alt boyutu puanı ( $\bar{X}=4,17$ ) deney grubunun puanından ( $\bar{X}=3,97$ ) yüksek olmasına rağmen bu fark anlamlı değildir.  $t(41)=1,10$ ,  $p>0,05$ .

Deney ve Kontrol grubunun Bilgisayar Programlamaya Karşı tutum Programlamada başarıya karşı tutum alt boyutu ön test puanları arasında anlamlı farklılık olup olmadığı puanların normal dağılım özelliği göstermesinden dolayı Bağımsız Örneklem t-testi ile analizleri yapılmıştır. Analiz sonuçlarına göre kontrol grubunun ön test programlamada

başarıya karşı tutum alt boyutu puanı ( $\bar{X}=4,34$ ) deney grubunun puanından ( $\bar{X}=3,97$ ) yüksek olmasına rağmen bu fark anlamlı değildir.  $t(41)=1,63$ ,  $p>0,05$ .

Deney ve Kontrol grubu bilgisayar programlamaya karşı tutum programlamada başarının sosyal algısı alt boyutu ön test puanları arasında anlamlı farklılık olup olmadığı puanların normal dağılım özelliği göstermesinden dolayı Bağımsız Örneklem t-testi ile analizleri yapılmıştır. Analiz sonuçlarına göre kontrol grubunun ön test programlamada başarının sosyal algısı alt boyutu puanı ( $\bar{X}=4,49$ ) deney grubunun puanından ( $\bar{X}=4,21$ ) yüksek olmasına rağmen bu fark anlamlı değildir.  $t(41)=1,47$ ,  $p>0,05$ .

Deney ve Kontrol Bilgisayar Programlamaya Karşı tutum Kendine güven ve güdülenme alt boyutu son test puanları arasında anlamlı farklılık olup olmadığı puanların normal dağılım özelliği göstermesinden dolayı Bağımsız Örneklem t-testi ile analizleri yapılmıştır.

Tablo 4.12. Deney Grubu ve Kontrol Grubu Bilgisayar Programlamaya Karşı tutum alt boyutları Son test Puanları Bağımsız Örneklem t-Testi Sonuçları

	Grup	N	$\bar{X}$	S	sd	t	p
Kendine güven ve güdülenme	Deney	22	3,62	0,57	41	0,62	0,53
	Kontrol	21	3,48	0,82			
Programlamanın Faydası	Deney	22	4,18	0,51	41	0,92	0,36
	Kontrol	21	3,99	0,79			
Programlamada başarıya karşı tutum	Deney	22	4,10	0,62	41	0,33	0,74
	Kontrol	21	4,19	0,97			
Programlamada Başarının Sosyal Algısı	Deney	22	4,18	0,64	41	2,33	0,02
	Kontrol	21	4,58	0,48			

Analiz sonuçlarına göre deney grubunun son test Kendine güven ve güdülenme alt boyutu puanlarının ( $\bar{X}=3,62$ ) kontrol grubunun puanlarından ( $\bar{X}=3,48$ ) yüksek olduğu görülmektedir. Tablo 4.12 incelendiğinde deney ve kontrol grubu programlamaya karşı tutum Kendine güven ve güdülenme alt boyutu son test puanları arasında anlamlı bir farklılık olmadığı görülmektedir  $t(41)=0,62$ ,  $p>0,05$ .

Deney ve kontrol grubu Bilgisayar Programlamaya Karşı tutum Programlamanın Faydası alt boyutu son test puanları arasında anlamlı farklılık olup olmadığı puanların normal dağılım özelliği göstermesinden dolayı Bağımsız Örneklem t-testi ile analizleri yapılmıştır. Analiz sonuçlarına göre deney grubunun son test Kendine güven ve güdülenme alt boyutu puanı ( $\bar{X}=4,18$ ) kontrol grubunun puanından( $\bar{X}=3,99$ ) yüksek olmasına rağmen bu fark anlamlı değildir.  $t(41)=0,92$ ,  $p>0,05$ .

Deney ve Kontrol Bilgisayar Programlamaya Karşı tutum Programlamada başarıya karşı tutum alt boyutu son test puanları arasında anlamlı farklılık olup olmadığı puanların normal dağılım özelliği göstermesinden dolayı Bağımsız Örneklem t-testi ile analizleri yapılmıştır. Analiz sonuçlarına göre kontrol grubunun son test programlamada başarıya karşı tutum alt boyutu puanı ( $\bar{X}=4,19$ ) deney grubunun puanından( $\bar{X}=4,10$ ) yüksek olmasına rağmen bu fark anlamlı değildir.  $t(41)=0,33$ ,  $p>0,05$ .

Deney ve kontrol grubu bilgisayar programlamaya karşı tutum programlamada başarının sosyal algısı alt boyutu son test puanları arasında anlamlı farklılık olup olmadığı puanların normal dağılım özelliği göstermesinden dolayı Bağımsız Örneklem t-testi ile analizleri yapılmıştır. Analiz sonuçlarına göre kontrol grubunun son test programlamada başarının sosyal algısı alt boyutu puanı ( $\bar{X}=4,58$ ) deney grubunun puanından( $\bar{X}=4,18$ ) yüksek olduğu görülmektedir. Tablo 4.12 incelendiğinde deney ve kontrol grubu programlamaya karşı tutum programlamada başarının sosyal algısı alt boyutu son test puanları arasında kontrol grubu lehine anlamlı bir farklılık olduğu görülmektedir  $t(41)=2,33$ ,  $p<0,05$ .

#### **4.3. Farklı Blok Tabanlı Programlama Aracı Kullanımının Öğrencilerin Bilgisayar Programlamaya Karşı Kaygılarına Etkisine İlişkin Bulgular**

Öğrencilerin programlamaya karşı kaygılarının incelenmesi amacıyla programlamaya karşı kaygıları arasında farklılık olup olmadığına bakılmıştır. Programlamaya karşı kaygı ön test ve son test puanlarına bakılarak yapılacak analizleri belirlemek amacıyla ön test ve son test kaygı puanlarının normal dağılım gösterip göstermediği Shapiro-Wilks testi ile incelenmiştir. Elde edilen bulgular Tablo 4.13'de verilmiştir.



Tablo 4.13. Programlamaya Karşı Kaygı Ön test ve Son test Puanlarının Shapiro-Wilks Testi Sonuçları

Testler	Gruplar	Shapiro-Wilks	p
Ön Test	Deney	0,92	0,07
Ön Test	Kontrol	0,96	0,67
Son Test	Deney	0,94	0,24
Son Test	Kontrol	0,96	0,69

Tablo 4.13 incelendiğinde Deney ön test ve son test kaygı puanları ile Kontrol ön test ve son test kaygı puanlarının normal dağılım özelliği gösterdiği görülmektedir ( $p>0,05$ ). Deney ve kontrol grubu ön test ve son test kaygı puanları arasında anlamlı farklılık olup olmadığı puanların normal dağılım özelliği göstermesinden dolayı Bağımsız Örneklem t-testi ile analizleri yapılmıştır.

Tablo 4.14. Deney ve Kontrol Grubu Programlamaya Karşı Kaygı Ön test Puanları Bağımsız Örneklem t-Testi Sonuçları

	Grup	N	$\bar{X}$	S	sd	t	p
Programlamaya Karşı Kaygı Ön test	Deney	22	2,66	2,28	41	0,77	0,93
	Kontrol	21	2,67	2,42			

Deney ve Kontrol ön test kaygı puanları arasında anlamlı farklılık olup olmadığı puanların normal dağılım özelliği göstermesinden dolayı Bağımsız Örneklem t-testi ile analizleri yapılmıştır. Tablo 4.14 incelendiğinde deney ve kontrol grubu programlamaya karşı kaygı ön test puanları arasında anlamlı bir farklılık olmadığı görülmektedir  $t(41)=0,77$ ,  $p>0,05$ .

Tablo 4.15. Deney ve Kontrol Grubu Programlamaya Karşı Kaygı Son test Puanları Bağımsız Örneklem t-Testi Sonuçları

	Grup	N	$\bar{X}$	S	sd	t	p
Programlamaya Karşı Kaygı Son test	Deney	22	2,34	2,28	41	0,76	0,45
	Kontrol	21	2,52	2,42			

Deney ve Kontrol son test kaygı puanları arasında anlamlı farklılık olup olmadığı puanların normal dağılım özelliği göstermesinden dolayı Bağımsız Örneklem t-testi ile analizleri yapılmıştır. Tablo 4.15 incelendiğinde deney grubunun programlamaya karşı kaygısı( $\bar{X}=2,34$ ) kontrol grubunun programlamaya karşı kaygısından( $\bar{X}=2,52$ ) düşük olmasına rağmen iki grubun programlamaya karşı kaygı son test puanları arasında anlamlı bir farklılık olmadığı görülmektedir  $t(41)=0,77$ ,  $p>0,05$ .

Tablo 4.16. Deney Grubu Ön test - Son test Programlamaya Karşı Kaygı Puanları İlişkili Örneklem t-Testi Sonuçları

	Kaygı Testi	N	$\bar{X}$	S	sd	t	p
Deney	Ön test	22	2,66	0,61	21	4,21	0,00
	Son test	22	2,34	0,61			

Deney grubundaki öğrencilerin ön test ve son test kaygı puanları arasındaki farkın anlamlı olup olmadığının incelenmesi amacıyla İlişkili Örneklem t-testi sonuçlarına bakılmıştır.

Tablo 4.16 incelendiğinde deney grubundaki öğrencilerin programlamaya karşı kaygı ön test ve son test puanları arasında anlamlı farklılık olduğu görülmektedir  $t(21)=4,21$   $p<0,05$ . Deney grubundaki öğrencilerin uygulama öncesi kaygı puanlarının ortalaması ( $\bar{X}=2,66$ ), uygulama sonrası kaygı puanlarının ortalamasından( $\bar{X}=2,34$ ) yüksek olduğu görülmektedir.

Tablo 4.17. Kontrol Grubu Ön test - Son test Programlamaya Karşı Kaygı Puanları İlişkili Örneklem t-Testi Sonuçları

	Kaygı Testi	N	$\bar{X}$	S	sd	t	p
Kontrol	Ön test	21	2,67	0,77	20	1,41	0,17
	Son test	21	2,52	0,89			

Kontrol grubundaki öğrencilerin ön test ve son test kaygı puanları arasındaki farkın anlamlı olup olmadığının incelenmesi amacıyla İlişkili Örneklem t-testi sonuçlarına bakılmıştır.

Tablo 4.17 incelendiğinde Kontrol grubundaki öğrencilerin programlamaya karşı kaygı ön test ve son test puanları arasında anlamlı farklılık olmadığı görülmektedir  $t(20)=1,41$   $p>0,05$ . Kontrol grubundaki öğrencilerin uygulama öncesi kaygı puanlarının ortalaması ( $\bar{X}=2,67$ ), uygulama sonrası kaygı puanlarının ortalamasından ( $\bar{X}=2,52$ ) yüksek olduğu görülmektedir. Fakat bu fark anlamlı değildir.



## V. BÖLÜM

### 5.TARTIŞMA

Bu araştırma BTP araçlarının programlama eğitiminde farklı yapıdaki araçların kullanımının öğrencilerin algoritma başarılarına, tutumlarına ve kaygılarına olan etkilerini konu almaktadır. Araştırma kontrol gruplu yarı deneysel olarak yürütülmüştür ve bu bölüm öğrencilerin BTP deneyimlerinin değerlendirilmesini ele almaktadır.

Araştırma bulgularına göre Scratch ve HackerCan gruplarının programlama başarı ön test puanlarının benzer olduğu görülmektedir. Bu sonuca göre Scratch ve HackerCan grubu öğrencilerinin birbirlerine denk gruplardan oluştuğu söylenebilir. Scratch ve HackerCan gruplarının programlama başarı son test sonuçlarında ise Scratch grubu öğrencileri HackerCan grubu öğrencilerine göre daha yüksek puan almalarına rağmen aradaki farkın anlamlı olmadığı görülmektedir. Elde edilen bu sonuca göre Scratch ile BTP yapan öğrenciler daha başarılı olmuşlardır. Bu sonucun ortaya çıkmasını öğrencilerin Scratch ile daha serbest programlama yapabiliyor olmalarına bağlayabiliriz.

HackerCan grubunun programlama başarı ön test ve son test sonuçlarına bakıldığında başarının yükselmesine rağmen arada anlamlı bir fark olmadığı görülmektedir. Ancak Scratch grubunun programlama başarı ön test ve son test sonuçlarına bakıldığında Scratch grubu öğrencilerinin uygulama sonrası puanlarının anlamlı bir şekilde arttığı görülmekte ve bu durum kullanılan BTP aracının etkili olduğunu ortaya koymaktadır. Bu bulgu alan yazında bulunan birçok çalışma tarafından desteklenmektedir. (Ozoran, Çağıltay, ve Topallı, 2012) özel bir üniversitede öğrenim gören öğrenciler üzerinde yaptıkları çalışmada Scratch kullanımının öğrencilerin programlama dersindeki başarılarını artırdığını belirtmişlerdir (Erol, 2015). (Rizvi ve Humphries, 2012) ise üniversite öğrencilerinin Scratch aracı ile programlamayı öğrenmelerini araştırdıkları çalışmada önceden Scratch eğitimi alan öğrencilerin almayanlara göre programlama dersindeki başarılarının daha yüksek olduğunu ortaya koymuşlardır. Bu sonuçlara bakıldığında araştırmamızda elde edilen bulguları destekler nitelikte oldukları görülmektedir.

HackerCan ve Scratch grubu öğrencilerinin cinsiyete göre programlama başarı son test sonuçlarına bakıldığında kadın ve erkek öğrencilerin puanlarının benzer olduğu görülmektedir. Bu sonuçlar göz önüne alındığında grupların cinsiyete göre farklılaşmaları

benzer düzeydedir. Bu bulgu alan yazın tarafından da desteklenmektedir. Örneğin (Olgun, 2014) 120 adet 6. sınıf öğrencisi ile yaptığı yüksek lisans tezinde programlamanın öğrencilerin düşünme stilleri üzerindeki etkisini incelemiştir ve düşünme stillerinin kız ve erkek öğrenciler tarafından benzer biçimde olduğunu ortaya koymuştur. Bir başka çalışmada (Uslu, Mumcu, ve Eğin, 2018), BTP eğitiminin ortaokul öğrencilerinin bilgi işlemsel düşünme becerilerine etkilerini araştırdıkları çalışma 6. Sınıfta öğrenim gören 55 öğrencisi ile yürütülmüştür bu çalışmaya göre kız ve erkek öğrencilerin BTP eğitimi öncesinde ve sonrasında bilgi işlemsel düşünme becerilerinin farklılaşmadığını belirtmektedirler.

Araştırmanın bir diğer bulgusu olan programlamaya karşı tutum ön test puanları HackerCan ve Scratch grupları arasında anlamlı bir farklılık oluşturmamaktadır. HackerCan ve Scratch grubu öğrencileri programlamaya karşı düşük ve benzer tutum içerisinde bulunmaktadırlar. Programlamaya karşı tutum son test sonuçlarına bakıldığında ise HackerCan grubunda bulunan öğrencilerin tutumlarında artış olmasına rağmen Scratch grubunda bulunan öğrencilerin programlamaya karşı tutumlarında bir düşüş görülmektedir. Bu sonuçlara baktığımızda öğrencilerin programlamaya yönelik olan düşük tutumlarının HackerCan kullanıldığında olumlu olarak etkilendiği Scratch kullanıldığında ise olumsuz olarak etkilendiği sonucuna varırız. Araştırmada elde edilen bu sonuçları destekler nitelikte olan çalışmalar alan yazında mevcuttur. (Yiğit, 2016), üniversite öğrencileri ile yaptığı yüksek lisans tezinde BTP eğitiminin öğrencilerin programlamaya karşı tutumlarını olumlu yönde geliştirdiği sonucuna varmıştır. (Kalelioğlu ve Gülbahar, 2014) 5. Sınıfta öğrenim gören 49 ortaokul öğrencisi ile yaptıkları çalışmada Scratch aracının programlama dersinde kullanımının programlamaya karşı ve programlama dersine karşı öğrencilerin tutumlarını iyileştirdiğini belirtmişlerdir. Bir başka çalışmada ise elde ettiğimiz bulguların tersi olarak bir sonuca varılmış buna göre, (Fesakis ve Serafeim, 2009) Scratch aracının öğrencilerin programlamaya karşı tutumlarına etkisini incelemek amacı ile 35 üniversite öğrencisi ile yapmış oldukları çalışmada Scratch aracının kullanımının öğrencilerin programlama dersi hakkındaki tutumlarını olumlu yönde etkilediği sonucuna varmışlardır.

Araştırmanın son ölçeği olan programlamaya karşı kaygı ölçeği ön test sonuçlarına bakıldığında Scratch ve HackerCan grubu öğrencileri arasında kaygı bakımından anlamlı bir farklılık görülmemektedir. Programlamaya karşı kaygı son test sonuçlarına göre Scratch ve HackerCan gruplarının programlamaya karşı kaygıları benzerlik göstermekte ve aralarında anlamlı bir farklılık bulunmamaktadır. HackerCan grubu programlamaya karşı kaygı ön test ve son test puanları karşılaştırıldığında uygulama öncesi kaygısı yüksek olan öğrencilerin uygulama sonrasında programlamaya karşı olan kaygılarının anlamlı bir

şekilde azaldığı görülmektedir. Scratch grubunda bulunan öğrencilerin programlamaya karşı kaygı ön test ve son test puanları arasındaki farka bakıldığında öğrencilerin uygulama sonrası kaygılarının azaldığı görülmektedir. Ancak aradaki farkın anlamlı olmadığı ortaya çıkmıştır. Alan yazında da benzer sonuçlar bulunmaktadır. (Uslu, Mumcu, ve Eğin, 2018), yaptıkları çalışmada BTP eğitiminden sonra öğrencilerin kaygılarının azaldığını belirtmişlerdir. (Demir, 2015), 87 üniversite öğrencisi ile yaptığı doktora tezinde Scratch aracının öğrencilerin programlama başarılarını ve kaygılarına etkisini incelemiştir ve Scratch aracı kullanımının programlamaya yönelik kaygıyı azalttığı sonucuna varmıştır. (Fesakis ve Serafeim, 2009) yaptıkları çalışmada Scratch aracı gibi eğitsel programlama dillerinin stres ve kaygı düzeyini azalttığını belirtmişlerdir.



## VI. BÖLÜM

### 6.SONUÇLAR ve ÖNERİLER

#### 6.1.Sonuçlar

Araştırmanın bu bölümünde bulgular göz önüne alınarak elde edilen sonuçlara yer verilmektedir. BTP eğitiminin farklı araçlarla verildiği çalışmada kontrol grubu ve deney grubu öğrencilerinin programlama başarılarının artmış olduğu görülmektedir. Kontrol grubunda kullanılan Scratch öğrencilerin programlama başarılarını daha fazla arttırmakta olduğu görülmektedir. Bunun nedeni Scratch uygulamasında kullanıcıların daha serbest ve özgür olmaları ve bu sayede daha yaratıcı ürünler ortaya koyabilmeleridir. Araştırmadan elde edilen bu sonuca göre BTP eğitiminin her iki BTP aracının programlama eğitiminde kullanımının olumlu sonuçlar meydana getireceği söylenebilir.

BTP eğitimi alan öğrencilerin programlamaya karşı tutumları olumlu yönde değişim göstermektedir. Çalışmada ortaya çıkan bu sonuca göre bireylerdeki programlama yapma zorluğu algısı ve programlamaya karşı olan düşük tutum düzeyi BTP araçları ile olumlu yönde değişim göstermektedir. Yine programlama başarısında olduğu gibi kontrol grubunda kullanılan Scratch aracı tutum düzeyinin artmasında daha etkili olmuştur. Bu durum BTP araçlarının kullanıcıyı daha özgür kıldığı zaman daha iyi motive ettiği ve tutumlarını daha fazla etkilediği söylenebilir.

Araştırmada öğrencilerin programlamaya karşı kaygıları incelenmiş ve uygulama öncesi ve sonrası kaygıları göz önüne alındığında öğrencilerin kaygılarında azalma görülmektedir. Programlamaya karşı kaygı düzeyinin deney grubu öğrencilerinde daha fazla azalmaya uğramış olduğu görülmektedir. Deney grubunun kullanmış olduğu HackerCan uygulamasında programlama yapmanın seviye seviye olması ve verilen yönlendirmelere göre yapılmasının kaygıyı azalttığı söylenebilir.

Araştırmadaki bütün bulgular göz önüne alındığında BTP eğitiminin bireyler tarafından ilgi çekici olarak bulunduğu, programlamaya karşı olan “zor” ve “sıkıcı” gibi kalıp yargıların yıkıldığı görülmüştür. Bireyler BTP ile özellikle serbest olarak programlama yapmaları sağlandığında daha başarılı ve istekli olabilmektedirler.

## 6.2. Öneriler

### 6.3. Arařtırmacılar İin Öneriler:

1. alıřma kısıtlı sürede ve sayıda bir örneklem grubu ile yapılmıřtır. Aynı alıřma farklı öđrenci grubu ve aynı BTP araçları ile yapılarak arařtırılabilir ve sonuçlar karşılařtırılabilir.
2. Farklı yapılardaki BTP araçlarının etkileri arařtırılabilir.
3. BTP araçlarının bilgisayarca düşünme becerilerine olan etkisi arařtırılabilir.
4. BTP araçlarının bilgisayar programlamaya karşı motivasyona olan etkisi arařtırılabilir.
5. BTP eğitimi kullanılan araçların genellikle küçük yařtaki bireylere hitap etmesinden dolayı ilk ve orta öğretim düzeyindeki öğrenciler üzerinde uygulanabilir.

### 6.4. Öğretmenler İin Öneriler:

1. BTP araçlarını kullanacak eğitimcilerin kullanacağı aracı seçerken dikkatli davranması gerekmektedir. Öğrenci grubunun özelliklerine göre bir araç seçimi yapılmalıdır.
2. BTP araçlarını kullanarak programlama eğitimi verecek olan öğretmenlerin yapacağı etkinlikleri önceden planlaması ve kazanımlara göre uygulaması daha etkili olabilir.
3. BTP araçları özellikle programlamanın temelini kavratmada kullanılmalarından dolayı farklı derslere da entegre edilerek öğretilir.



## 7.Kaynakça

- Akpınar, Y., ve Altun, A. (2014). Bilgi Toplumu Okullarında Programlama Eğitimi Gereksinimi. *İlköğretim Online*, 13(1), 1-4.
- Aytekin, A., Çakır, F. S., Yücel, Y. B., ve Kulaözü, İ. (2018). Geleceğe Yön veren Kodlama Bilimi ve Kodlama Öğrenmede Kullanılabilecek Bazı Yöntemler. *Avrasya Sosyal ve Ekonomi Araştırmaları Dergisi*, 5(5), 24-41.
- Başer, M. (2013). Bilgisayar Programlamaya Karşı Tutum Ölçeği Geliştirme Çalışması. *The Journal of Academic Social Science Studies*, 6(6), 199-215.
- Bergin, J., Martinez, M. P., Brodlie, K., Goldweber, M., McNally, M., Naps, T. ve Wilson, J. (1996). An overview of visualization: its use and design Report of the Working Group on Visualization. *Acm Sigcse Bulletin*, 28, 192-200.
- Berksoy, İ., Sözcü, Ö. F., Armağan, E., ve Arslan, A. (2016). Algoritma ve Programlama Eğitiminde Scratch Programı Üzerine Yapılan Çalışmaların Analizi. Akademik Bilişim Konferansı, 201-207, Adnan Menderes Üniversitesi, Aydın.
- Büyüköztürk, Ş. (2016). *Sosyal Bilimler İçin veri Analizi El Kitabı* (22. baskı). Ankara: Pegem Akademi.
- Calder, N. (2010). Using scratch: an integrated problem-solving approach to mathematical thinking. *Australian Primary Mathematics Classroom*, 15(4), 9-14.
- Cheung, K. C., Koon, K. W., Cheng, S. K., ve Choo, M. L. (1990). *Meaningful Measurement in the Classroom Using the Rasch Model: Some Exemplars*.
- Clements, D. H., ve Gullo, D. F. (1984). Effects of Computer Programming on Young Children's Cognition. *Journal of Educational Psychology*, 76(6), 1051-1058.
- Cooper, S., Dann, W., ve Pausch, R. (2000). Alice: a 3-D tool for introductory programming concepts. *Journal of Computing Sciences in Colleges*, 15(5), 107-116.
- Çatlak, Ş., Tekdal, M., ve Baz, F. Ç. (2015). Scratch Yazılımı İle Programlama Öğretiminin Durumu: Bir Doküman İnceleme Çalışması. *Journal of Instructional Technologies & Teacher Education*, 4(3), 13-25.
- Demir, F. (2015). Programlama Öğretiminde Eğitsel Programlama Dilinin Farklı Kullanımlarının Programlama Başarısı ve Kaygısına Etkisi. Doktora tezi. Atatürk Üniversitesi, Erzurum

- Demirer, V., ve Sak, N. (2016). Dünyada Ve Türkiye'de Programlama Eğitimi Ve Yeni Yaklaşımlar. *Eğitimde Kuram ve Uygulama Dergisi*, 12(3), 521-546.
- Doğan, U. (2015). Ortaokul Öğrencilerinde Bilgisayar Oyunu Geliştirme Sürecinin Öğrencilerin Eleştirel Düşünme Becerilerine ve Algoritma Başarılarına Etkisi. Yüksek lisans tezi. Yıldız Teknik Üniversitesi, İstanbul.
- Du, J., Wimmer, H., ve Rada, R. (2016). "Hour of Code": Can It Change Students' Attitudes toward Programming? *Journal of Information Technology Education*, 15, 52-73.
- Erol, O. (2015). Scratch İle Programlama Öğretiminin Bilişim Teknolojileri Öğretmen Adaylarının Motivasyon ve Başarılarına Etkisi. Doktora tezi. Anadolu Üniversitesi, Eskişehir.
- Erümit, A. K., Benzer, A. İ., Aksoy, D. A., Aksoy, A., ve Şahin, G. (2017). Algoritmik Düşünme İçin Programlama Öğretimi Adımları. *Eğitim Teknolojileri Okumaları*, 1-18. Ankara: TOJET
- Fesakis, G., ve Serafeim, K. (2009). Influence of the Familiarization with "Scratch" on Future Teachers' Opinions and Attitudes about Programming and ICT in Education. *ACM SIGCSE Bulletin*, 41(3), 258-262.
- Futschek, G. (2006). Algorithmic Thinking: The Key for Understanding Computer Science. *The Bridge between Using and Understanding Computers*, (159-168). Berlin.
- Futschek, G., ve Moschitz, J. (2010). Developing Algorithmic Thinking by Inventing and Playing Algorithms. *Constructionism*, (1-10). Paris.
- Garner, S. (2003). Learning Resources and Tools to Aid Novices Learn Programming. *Informing science & information technology education joint conference*, (213-222). Pori.
- Genç, Z., ve Karakuş, S. (2011). Tasarımla öğrenme: Eğitsel bilgisayar oyunları tasarımında Scratch kullanımını. *5th International Computer & Instructional Technologies Symposium*, (981-987). Elazığ.
- Gorman, H., ve Bourne, L. E. (1983). Learning to think by learning LOGO: Rule learning in third-grade computer programmers. *Bulletin of the Psychonomic Society*, 21(3), 165-167.
- Güneş, A., ve Karabak, D. (2013). Ortaokul Birinci Sınıf Öğrencileri İçin Yazılım Geliştirme Alanında Müfredat Önerisi. *Eğitim ve Öğretim Araştırmaları Dergisi*, 2(3), 175-181.

- Hair, J. F., Black, W. C., Babin, B. J., ve Anderson, R. E. (2010). *Multivariate Data Analysis* (7. Baskı). New Jersey: Prentice Hall, Upper Saddle River.
- Hundhausen, C. D., ve Brown, J. L. (2008). Designing, visualizing, and discussing algorithms within a CS 1 studio experience: An empirical study. *Computers & Education*, 50, 301-326.
- Hundhausen, C. D., Douglas, S. A., ve Stasko, J. T. (2002). A Meta-Study of Algorithm Visualization Effectiveness. *Journal of Visual Languages and Computing*, 13(3), 259-290.
- Kaleliođlu, F., ve Glbahar, Y. (2014). The Effects of Teaching Programming via Scratch on Problem Solving Skills: A Discussion from Learners' Perspective. *Informatics in Education*, 13(1), 33-50.
- Karasar, N. (2005). *Bilimsel Arařtırma Yntemleri*. Ankara: Nobel Yayıncılık.
- Kaucic, B., ve Asic, T. (2011). Improving Introductory Programming with Scratch? Proceedings of the 34th MIPRO International Convention, (1095-1100), Opatija.
- Kinnunen, P., ve Malmi, L. (2008). CS minors in a CS1 course. In Proceeding of the Fourth international Workshop on Computing Education Research, (79-90). Sydney.
- Kobsiripat, W. (2015). Effects of the media to promote the scratch programming capabilities creativity of elementary school students. *Procedia - Social and Behavioral Sciences*, 174, 227-232.
- Kuzu, A. (2013). *Harezmi: Sıfır Rakamını Bulan Deha*. İstanbul: Hiperlink.
- Mahadevan, A., Freeman, J., ve Magerko, B. (2016). An interactive, graphical coding environment for EarSketch online using Blockly and Web Audio API. *Web Audio Conference*, Atlanta.
- Maloney, J., Resnick, M., Rusk, N., Silverman, B., ve Eastmond, E. (2010). The Scratch Programming Language and Environment. *Transactions on Computing Education*, 10(4), 16.
- Mldner, T., Shakshuki, E., Kerren, A., Shen, Z., ve Bai, X. (2005). Using Structured Hypermedia to Explain Algorithms. *Proceedings of the 3rd IADIS International Conference e-Society* (499-503). Qawra: IADIS Press.
- Olgun, K. B. (2014). Programlamanın Ortaokul đrencilerinin Dřnme Stilleri zerine Etkisi. Yksek lisans tezi. İstanbul niversitesi, İstanbul.

- Ozoran, D., Çağıltay, N. E., ve Topallı, D. (2012). Using Scratch in Introduction To Programming Course For Engineering Students. *2nd International Engineering Education Conference*, (125-132), Antalya
- Papatğa, E. (2016). Okuduğunu Anlama Becerilerinin Scratch Programı Aracılığıyla Geliştirilmesi. Doktora tezi. Anadolu Üniversitesi, Eskişehir.
- Patan, B. (2016). Okul Öncesi Kodlama Öğretim Programının Geliştirilmesi. Yüksek lisans tezi. Bahçeşehir Üniversitesi, İstanbul.
- Portelance, D. J., Strawhacker, A. L., ve Bers, M. U. (2016). Constructing The Scratch Jr Programming Language In The Early Childhood Classroom. *International Journal of Technology and Design Education*, 26(4), 489-504.
- Powers, K., Cooper, S., Goldman, K. J., Carlisle, M., McNally, M., ve Proulx, V. (2006). Tools for Teaching Introductory Programming: What Works? *Computer and Information Science Education*, 560-561.
- Rizvi, M., ve Humphries, T. (2012). A Scratch-based CS0 Course for At-risk Computer Science Majors. *Frontiers in Education Conference Proceedings* (1-5). Washington: IEEE.
- Roy Singh, R. (1991). *Education for the twenty-first century: Asia-Pacific perspectives*. Bangkok: Unesco Principal Regional Office for Asia and the Pacific.
- Roy, P. V., ve Haridi, S. (2004). *Concepts, Techniques, and Models of Computer Programming*. Cambridge: MIT press.
- Saltan, F. (2016). Looking at Algorithm Visualization through the Eyes of Pre-service ICT Teachers. *Universal Journal of Educational Research*, 4(2), 403-408.
- Saltan, F. (2017). The Impact of Online Algorithm Visualization on ICT Students' Achievements in Introduction to Programming Course. *Journal of Education and Learning*, 6(1), 184-192.
- Saygıner, Ş., Tüzün, H. (2017). Programlama eğitiminde yaşanan zorluklar ve çözüm önerileri. *Uluslararası Bilgisayar ve Öğretim Teknolojileri Sempozyumu*, (78-90), Malatya.
- Sayın, Z., Seferoğlu, S. S. (2016). Yeni Bir 21. Yüzyıl Becerisi Olarak Kodlama Eğitimi ve Kodlamanın Eğitim Politikalarına Etkisi. Adnan Menderes Üniversitesi, *Akademik Bilişim Konferansı*, (3-5), Aydın.

- Shin, S., ve Park, P. (2014). A Study on the Problem Solving Ability of Primary Students through Mathematics Logic Solving with Scratch. *International Information Institute*, 17(10), 5277-5282.
- Shin, S., Park, P., ve 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), 247-249.
- Shu, N. C. (1999). Visual programming: Perspectives and approaches. *IBM Systems Journal*, 38(4), 199-221.
- Sykes, E. R. (2007). Determining the Effectiveness of the 3D Alice Programming Environment at the Computer Science 1 Level. *Journal of Educational Computing Research*, 36(2), 223-244.
- Topal, A. (2015). Ortaokul 6. Sınıf Öğrencilerinin Standart Bir Algoritmaya Çözülebilir ve Çözülemez Problemlerde Kullandıkları Matematiksel Düşüncelerinin İncelenmesi. Yüksek lisans tezi. Gaziantep Üniversitesi, Gaziantep.
- Törley, G. (2014). Algorithm visualization in teaching practice. *Acta Didactica Napocensia*, 7(1), 1-17.
- URL-1, <https://ec.europa.eu/digital-single-market/coding-21st-century-skill> European Commission. 02 Mayıs 2019
- Url-2, <http://ttkb.meb.gov.tr> Bilişim Teknolojileri Ve Yazılım Dersi Programı 02 Mayıs 2019
- Url-3, [www.scratch.mit.edu](http://www.scratch.mit.edu) Scratch 13 Mart 2019
- Url-4, <https://kodlamayabasla.com/platformlar/web-siteler/code-org-nedir.html> Kodlamaya Başla 24 Nisan 2019
- Url-5, <http://www.hackercan.com/tr/genel-ozellikler> Hacker Can 13 Mart 2019
- Url-6, <http://www.eba.gov.tr/haber/1382225534> Eğitim Bilişim Ağı 24 Nisan 2019
- Uslu, N. A., Mumcu, F., Ve Eğin, F. (2018). Görsel Programlama Etkinliklerinin Ortaokul Öğrencilerinin Bilgi-İşlemsel Düşünme Becerilerine Etkisi. *Ege Eğitim Teknolojileri Dergisi*, 2(1), 19-31.
- Weintrop, D., ve Wilensky, U. (2015). To Block or not to Block, That is the Question: Students' Perceptions of Blocks-based Programming. *Proceedings of the 14th International Conference on Interaction Design and Children*, (199-208). Medford.

- Williams, L., Ve Cernochova, M. (2013). Literacy from Scratch. *Proceedings of the 10th IFIP World Conference on Computers in Education*, (17-27).
- Yecan, E., Özçınar, H., Ve Tanyeri, T. (2017). Bilişim Teknolojileri Öğretmenlerinin Görsel Programlama Öğretimi Deneyimleri. *İlköğretim Online*, 16(1), 377-393.
- Yiğit, M. F. (2016). Görsel Programlama Ortamı İle Öğretimin Öğrencilerin Bilgisayar Programlamayı Öğrenmesine Ve Programlamaya Karşı Tutumlarına Etkisinin İncelenmesi. Yüksek lisans tezi. Ondokuz Mayıs Üniversitesi, Samsun.
- Yükseltürk, E., ve Altıok, S. (2016). Bilişim Teknolojileri Öğretmen Adaylarının Programlama Öğretiminde Scratch Aracının Kullanımına İlişkin Algıları. *Mersin Üniversitesi Eğitim Fakültesi Dergisi*, 12(1), 39-52.
- Yükseltürk, E., ve Altıok, S. (2016). BT Öğretmen Adayları Tarafından Scratch Görsel Programlama Aracı ile Geliştirilen Eğitsel Oyunların İncelenmesi. *SDU International Journal of Educational Studies*, 3(1), 59-66.



## EK-1. Algoritma Başarı Testi

Değerli katılımcı bu test algoritma bilginizi ölçmek amacı ile hazırlanmış olup Vereceğiniz yanıtlar gizli kalacak ve sadece bilimsel amaçlı kullanılacaktır. Katılımınız için teşekkür ederim.  
KEMAL BİLİR

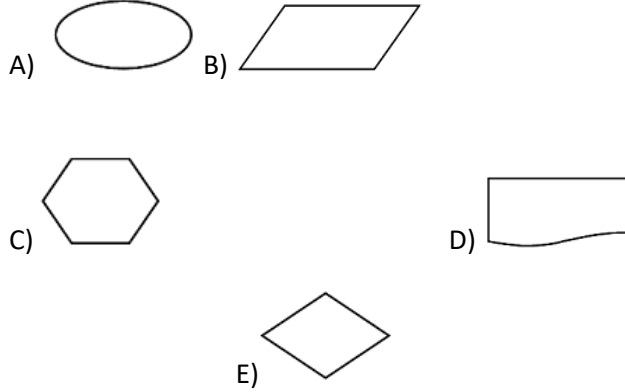
Okul Numaranız:  
Grubunuz:

- 1) Girdiğimiz değerleri alan veya programın çalışmasıyla bazı değerlerin atandığı veri tutucularına ne denir?
  - A) Döngü
  - B) Algoritma
  - C) Değişken
  - D) Koşullu yapı
  - E) Alt program
  
- 2) Aşağıdakilerden hangisi bir veri çıkışı işlemidir?
  - A) Cep telefonundan mesaj yazma
  - B) Fotoğraf makinesi ile fotoğraf çekinme
  - C) Mouse ile tıklama yapma
  - D) Mikrofon ile ses kayıt etme
  - E) Cep telefonu bataryasının azalma sinyali vermesi
  
- 3) Spor seçmelerinde seçmeye katılacak kişilerin boyları (B) 1.80 cm'den az olmamalı ve kilosu (K) 110 kg'ı geçmemelidir şartlarının temel gösterimi aşağıdakilerden hangisidir?
  - A)  $K \leq 110$  VE  $B \leq 1.80$
  - B)  $K \leq 110$  VEYA  $B \leq 1.80$
  - C)  $K \geq 110$  VE  $B \leq 1.80$
  - D)  $K \leq 110$  VE  $B \geq 1.80$
  - E)  $K \geq 110$  VEYA  $B \geq 1.80$
  
- 4) Yandaki sembol akış diyagramında hangi görevde kullanılır?
  - A) Programa veri girişi
  - B) Aritmetik işlemler ve her türlü atama işlemi
  - C) Karar verme
  - D) Ekran ve yazıcıya bilgi çıkışı
  - E) Birleştirici veya bağlantı noktalarını temsil etmek





- 5) Program içinde belli noktaların art arda tekrar edileceğini gösteren akış diyagramı sembolü aşağıdakilerden hangisidir?



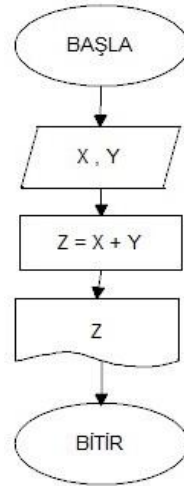
- 6) Yandaki algoritması verilen programa sırasıyla "Can" , "15" ifadeleri girilmiştir. Algoritmanın ekran çıktısı aşağıdakilerden hangisidir?

- A) Merhaba 15 Can  
B) Can 15 Merhaba  
C) 15 Can Merhaba  
D) Merhaba Can 15  
E) Can Merhaba 15

```

Başla
A=Yaz("Adınızı giriniz")
Oku(A)
B=Yaz("Yaşınızı giriniz")
Oku(B)
C="Merhaba"
Yaz(C,A,B)
Bitir

```



- 7) Yukarıdaki şekildeki programın açıklaması aşağıdakilerden hangisidir?
- A) Ekranda girilen iki sayının ortalamasını bulan bir program  
B) Sabit iki sayının toplamını bulan bir program  
C) Ekranda girilen iki sayının toplamını bulan bir program  
D) Sabit iki sayının ortalamasını bulan bir program

- E) Ekranda girilen iki sayının çarpımını bulan bir program
- 8) Klavyeden kilo bilgisi giriliyor. Kilo:60kg'dan az ise "Kilonuz Az" ; 60-100kg arasında ise "Kilonuz Normal" ; 100kg dan fazla ise "Kilonuz Fazla" çıktısını ekranda görüntüleyen programın içerisinde aşağıdaki programlama yapılarından hangisi bulunur?
- A) Döngü yapısı  
B) Değişken yapısı  
C) İç içe döngü yapısı  
D) Sayaç yapısı  
E) İç içe şart yapısı

- 9) Ahmet aklından bir sayı tutuyor. Bu sayının karesini alıyor, 6 çıkartıyor, 5 ile çarpıyor, 2 çıkartıyor 8'e bölüyor. Sonuç 6 ise, Ahmet'in aklından tuttuğu sayı kaçtır?

- A) 1  
B) 2  
C) 3  
D) 4  
E) 5

- 10) Yan tarafta algoritması verilen program ile elde edilmek istenen sonuç nedir?

- A) a ve b'yi yazdırmak  
B) a'yı yazdırmak  
C) b'yi yazdırmak  
D) İki sayıdan küçük olanı yazdırmak  
E) İki sayıdan büyük olanı yazdırmak

Başla  
a ve b sayısını girin  
a>b ise a'yı yaz değilse b'yi yaz  
Bitir

- 11) Yan tarafta algoritması verilen program ile elde edilmek istenen sonuç nedir?

- A) 100'den küçük sayıları yazdırmak  
B) 1'den 100'e kadar olan sayıları toplamak  
C) 0 yazdırmak  
D) s değerini yazdırmak  
E) s+t değerini yazdırmak

Başla  
t=0 ve s=0 al  
s=s+1 işlemini yap  
s<100 ise t=t+s ve 3.adıma git  
t değerini yazdır  
Bitir.

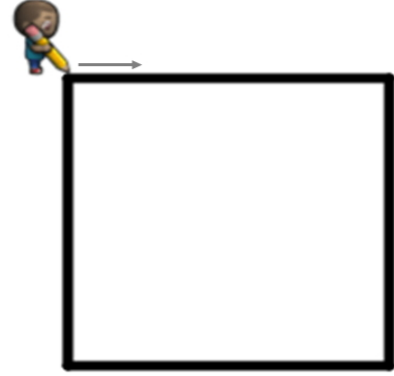
- 12) Yan tarafta algoritması verilen programda x yerine 4 yazılırsa programın çıktısı aşağıdakilerden hangisi olur?

- A) 4>0  
B) x=4  
C) negatif  
D) pozitif  
E) x=0

Başla  
x sayısını gir  
x>0 ise pozitif yaz değilse negatif  
Bitir.

- 13) Şekildeki çocuğun ok yönünde giderek kare çizmesini sağlamak için takip edilmesi gereken adımlar aşağıdakilerden hangisinde doğru verilmiştir?(Kare'nin her bir kenarı 5cm'dir.)

- A) 5cm ilerle sağa dön(4 defa)
- B) 4cm ilerle sola dön (4 defa)
- C) 5cm ilerle sola dön (5 defa)
- D) 5cm ilerle sola dön (4 defa)
- E) 4cm ilerle sağa dön (4 defa)



- 14) Yan tarafta algoritması verilen programda programın çıktısının gaz olabilmesi için t değerinin aşağıdakilerden hangisi olması gerekmektedir?

- A) 0
- B) -80
- C) 80
- D) 90
- E) 150

Başla  
t değerini girin  
t<0 "Kati" yaz  
t>0 ve t<100 "Sıvı" yaz  
t>100 "Gaz" yaz  
Bitir.

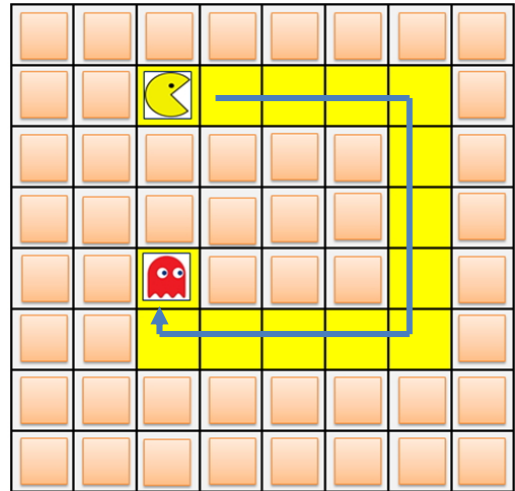
- 15) Yandaki programın çıktısı nedir?

- A) 15
- B) 25
- C) 70
- D) 80
- E) 375

a=75  
b=5  
yaz(a/b)

- 16) Pacman'ın hayaleti yakalaması için hangi adımları takip etmesi gerekmektedir?

- A) 4 birim sağa 5 birim aşağıya 5 birim sola ilerle
- B) 4 birim ilerle sağa dön(3defa) 1 birim ilerle
- C) 5 birim ilerle sola dön(4 defa) 1 birim ilerle
- D) 4 birim ilerle sağa dön(4 defa)
- E) 4 birim ilerle sağa dön(3defa) 2 birim ilerle



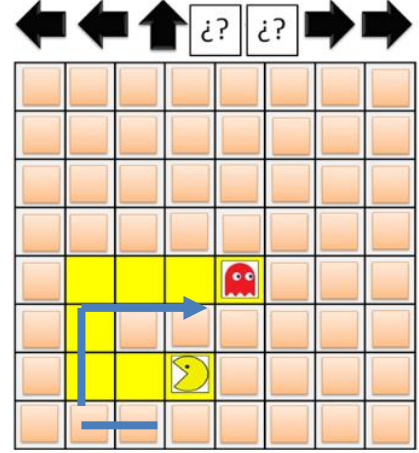
17) Yandaki algoritmada sayi1=8, sayi2=15 değeri girilirse algoritmanın ekran çıktısı ne olur?

- A) "buyuktur"
- B) "YAZ"
- C) YAZ "kucuktur"
- D) Ekranı bir şey yazmaz
- E) "kucuktur"

```
BASLA
YAZ "iki sayı giriniz"
OKU sayi1 , sayi2
EGER sayi1<sayi2 YAZ "buyuktur"
DEĞİLSE YAZ "kucuktur"
BİTİR
```

18) Yandaki şekilde Pac-man'ın hayaletle ulaşması için verilen adımlardan soru işareti ile belirtilen adımlara sırasıyla aşağıdakilerden hangisi gelmelidir?

A)	→ ↓
B)	↓ ←
C)	← ↑
D)	↑ →
E)	↑ ↑



19) Aşağıdaki algoritmanın çıktısı hangi seçenekte doğru olarak verilmiştir?

- A) 113
- B) 23
- C) 133
- D) 4290
- E) 123

```
BAŞLA
a=55
b=78
yaz(a+b)
BİTİR
```

20) Klavyeden girilen iki sayıyı toplayıp ortalamasını ekrana yazan programın algoritması yanda verilmiştir. Buna göre adım4'e aşağıdakilerden hangisi gelmelidir?

- A)  $ortalama = (sayi1 + sayi2) / 2$
- B) "birinci sayı – ikinci sayı"
- C) Sayıları ekrana yaz
- D)  $Ortalama(sayi1 + sayi2)$
- E) Ekrana yaz sayi1 + sayi2

```
Adım1: Başla
Adım2: "Birinci sayıyı giriniz" , sayi1
Adım3: "ikinci sayıyı giriniz" , sayi2
Adım4:
Adım5: Ekrana yaz, ortalama
Adım6: Bitir
```

## EK-2. Bilgisayar Programlamaya Karşı Tutum Ölçeği

Okul numaranız:

Cinsiyetiniz: KadınErkek

Değerli katılımcı, ölçekte yer alan sorulara verdiğiniz yanıtlar sadece bilimsel amaçlı kullanılacaktır. Tüm soruları dikkatle okuyarak cevabınızı ifadenin karşısındaki seçeneklerden sizin için en uygun olanı işaretleyerek belirtiniz.

		Kesinlikle Katılmıyorum	Katılmıyorum	Kararsızım	Katılıyorum	Kesinlikle Katılıyorum
1	Bilgisayar programlama problemlerini çözmede kendime güveniyorum.					
2	Programlamayı öğreneceğimden eminim.					
3	Daha zor programlama problemlerinin üstesinden geleceğimi düşünüyorum.					
4	Programlama derslerinde iyi notlar alabilirim.					
5	Söz konusu programlama olunca kendime güvenim çoktur.					
6	Programlamada iyi değilim.					
7	İleri programlama derslerinde başarılı olacağımı sanmıyorum.					
8	Bilgisayar programlamayı iyi yapan birisi değilim.					
9	Çok çalışmama rağmen programlama bana inanılmaz zor geliyor.					
10	Birçok dersin üstesinden geliyorum fakat programlama problemlerinde becerim yok.					
11	En başarısız olduğum ders bilgisayar programlamadır.					
12	Programlama derslerinde mükemmel öğrenci olarak algılanmak beni mutlu eder.					
13	Programlama derslerinde göze çarpan bir öğrenci olmak bana gurur verir.					
14	Programlama derslerinde en yüksek notları almak beni mutlu eder.					

15	Programlama yarışmalarında bir ödül almak benim için muhteşemdir.					
16	Programlama yarışmasında birinci olmak beni sevindirir.					
17	Programlama derslerinde akıllı bir öğrenci sayılmak benim için büyük bir şeydir.					
18	Programlama ile ilgili bir yarışmada bir ödül kazanmak bana cazip gelmez.					
19	Programlama derslerinde yüksek not alırsam onu önemsemem.					
20	Eğer programlama dersinde bir kere yüksek not alırsam bir daha istemem.					
21	Programlama derslerinde başarılı bir öğrenci olursam insanlar beni daha az sever.					
22	İnsanların programlama derslerinde akıllı öğrenci olduğumu düşünmelerini istemem.					
23	Sonraki çalışmalarında programlamaya ihtiyacım olacak.					
24	Programlamaya çalışıyorum çünkü onun ne kadar gerekli olduğunu biliyorum.					
25	Programlamayı bilmek iş bulmama yardımcı olacak.					
26	Programlama değerli ve yararlıdır.					
27	İş hayatımda programlamayı birçok şekilde kullanacağım.					
28	İş hayatım için programlamanın önemi yok.					
29	İş hayatımda programlamanın önemi olmayacak.					
30	İş hayatımda programlamayı seyrek kullanacağım.					
31	Programlama dersleri almak zaman kaybıdır.					
32	Okul sonrasında programlamanın çok az kullanım yeri olacağını tahmin ediyorum.					
33	Bir programlama problemi ile karşılaşırsam onu çözünceye kadar uğraşırım.					
34	Bir program üzerinde çalışmaya başlarsam durmakta zorlanırım.					
35	Programlama dersinde cevapsız bir soru kalırsa, daha sonra hakkında düşünmeye devam ederim.					
36	Hemen anlayamadığım bilgisayar programlama problemleri beni kamçılar					
37	Programlama probleminin zor olması ilgimi çekmez.					
38	İnsanların program yazarken çok zaman harcamaları ve bundan zevk almalarını anlamıyorum.					

### EK-3. Programlama Kaygı Ölçeği

Aşağıdaki maddeler programlamaya yönelik kaygıyı ölçmek amacı ile hazırlanmış ve bilimsel bir araştırmada kullanılacaktır. Lütfen her maddeyi dikkatle okuyup size yansıtma düzeyine göre işaretleme yapınız.

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

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

17.	Bilgisayar programımda hata olduđu her zaman kendimi gergin hissediyorum.					
18.	Arkadaşlarım benim anlamadığım programlama konularında konuştuğunda sıkıntı yaşıyorum.					
19.	Bilgisayarla çalışma konusunda her zaman kendimi güvensiz hissediyorum.					





#### EK-4.Çalışma Esnasından Fotoğraflar



## ÖZGEÇMİŞ

### KİŞİSEL BİLGİLER

Adı-Soyadı : Kemal BİLİR  
Doğum tarihi : 04.12.1991  
Doğum yeri : Amasya

### EĞİTİM DURUMU

Lisans Öğrenimi : Tokat Gaziosmanpaşa Üniversitesi

### BİLİMSEL FAALİYETLERİ

#### a) Yayınlar (-SCI -Diğer)

- Bilir, K., Korkmaz, Ö. ve Çakır R. (2018) BT Öğretmenlerinin Derslerinde Gerçekleştirdiği Etkinlikler, Bu Etkinliklerin Öğretim Programına Uygunluğu ve Gerekçeleri. *Mersin Üniversitesi Eğitim Fakültesi Dergisi*. 14(3), 1138-1155.
- Saltan, F., Türkyılmaz, T., Karaçaltı, C. ve Bilir, K., (2018) Güncel Eğitim Teknolojilerinin Fen Eğitiminde Kullanılması: Tarama Çalışması. *Çukurova Üniversitesi Eğitim Fakültesi Dergisi*. 47(1), 308-336.

#### b) Bildiriler (-Uluslararası –Ulusal)

- Çakır, R., Karaçaltı, C. Kılıç, M. ve Bilir, K. (2017) Mobil Destekli Öğretimin Öğrencilerin Yabancı Dil Kelime Öğrenimi Üzerindeki Etkisi, 5. Uluslararası Eğitim Teknolojileri ve Öğretmen Eğitimi Semineri, İzmir.
- Çakır, R., Aygül, F., Kılıç, M., Karaçaltı, C. ve Bilir, K. (2017) Uzaktan Eğitimde Ders Veren Öğretim Elemanlarının Bireysel Yenilikçi Özelliklerinin İncelenmesi, 5. Uluslararası Eğitim Teknolojileri ve Öğretmen Eğitimi Semineri, İzmir.
- Saltan, F., Aygül, F., Türkyılmaz, T., Bilir, K. ve Karaçaltı, C.(2017) İngilizce Kelime Öğretiminde Flaş kart mı? Artırılmış Gerçeklik mi?, 11. Uluslararası Bilgisayar ve Öğretim Teknolojileri Sempozyumu, Malatya.
- Bilir, K., ve Öner, F. (2018) Görsel Programlama Araçlarının Programlama Öğretimindeki Etkisinin Başarı ve Kaygı Açısından İncelenmesi. 17. Uluslararası Sınıf Öğretmenliği Sempozyumu. Ankara

### İŞ DENEYİMİ

Çalıştığı Kurumlar ve Yıl : Milli Eğitim Bakanlığı 2019-....

### İLETİŞİM

E-posta Adresi : kemalbilir05@gmail.com