

**METİN TEMELLİ PROGRAMLAMA ÖĞRETİMİ: ORTAOKUL
ÖĞRENCİLERİNİN BİLGİSAYARCA DÜŞÜNME BECERİLERİ VE
AKADEMİK BAŞARILARININ İNCELENMESİ**



Nur Faden KARAÇAM DUMAN

OCAK 2020

**METİN TEMELLİ PROGRAMLAMA ÖĞRETİMİ: ORTAOKUL
ÖĞRENCİLERİNİN BİLGİSAYARCA DÜŞÜNME BECERİLERİ VE
AKADEMİK BAŞARILARININ İNCELENMESİ**

**BAHÇEŞEHİR ÜNİVERSİTESİ
EĞİTİM BİLİMLERİ ENSTİTÜSÜ
YÜKSEK LİSANS TEZİ**

NUR FADEN KARAÇAM DUMAN

**EĞİTİM TEKNOLOJİSİ DALINDA
YÜKSEK LİSANS DERECESESİ İÇİN GEREKLİ ÇALIŞMALAR YERİNE
GETİRİLMİŞTİR.**

OCAK 2020

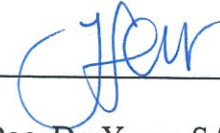
Eđitim Bilimleri Enstitüsü'nün Onayı



Doç. Dr. Enisa MEDE

Enstitü Müdürü

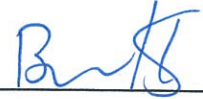
Bu tezin Yüksek Lisans derecesinde bir tez olarak gerekli çalışmaları yerine getirdiđini onaylarım.



Doç. Dr. Yavuz SAMUR

Koordinatör

Okuduđumuz bu tezin Yüksek Lisans derecesinde bir tez olarak onaylanması, düşüncemize göre, amaç ve kalite olarak tamamen uygundur.



Dr. Öğr. Üyesi Burak ŞİŞMAN

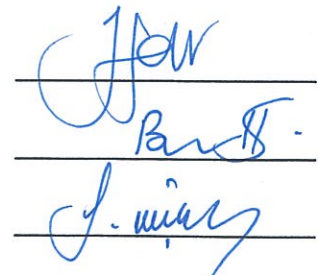
Tez Danışmanı

Komite Üyeleri

Doç. Dr. Yavuz SAMUR (BAU, BÖTE)

Dr. Öğr. Üyesi Burak ŞİŞMAN (İUC, BÖTE)

Doç. Dr. Sevda KÜÇÜK (İUC, BÖTE)



Bu tezdeki tüm bilgilerin akademik kurallara ve etik ilkelere uygun olarak elde edildiğini ve sunulduğunu; ayrıca bu kuralların ve ilkelerin gerektirdiği şekilde, bu çalışmadan kaynaklanmayan bütün atıfları yaptığımı beyan ederim.

Ad, Soyad : Nur Faden Koşan Duman

İmza : 

ÖZ

METİN TEMELLİ PROGRAMLAMA ÖĞRETİMİ: ORTAOKUL ÖĞRENCİLERİNİN BİLGİSAYARCA DÜŞÜNME BECERİLERİ VE AKADEMİK BAŞARILARININ İNCELENMESİ

Karaçam Duman, Nur Faden

Yüksek Lisans, Eğitim Teknolojisi Yüksek Lisans Programı

Tez Danışmanı: Dr. Öğr. Üyesi Burak ŞİŞMAN

Ocak 2020, 119 sayfa

Bu çalışmanın amacı, 6. sınıf öğrencilerinin Bilişim Teknolojileri ve Yazılım dersi kapsamında metin temelli programlama öğretiminin bilgisayarca düşünme becerilerine ve akademik başarılarına etkisini incelemektedir. Bu amaç doğrultusunda, blok temelli programlama öz-yeterlilik algısına sahip olan ortaokul öğrencileri ile metin temelli programlama öğretimi çalışmaları planlanmıştır. Çalışma sürecinde karma yöntemlerden açılımlayıcı sıralı desen kullanılmıştır. Çalışma grubu, İstanbul ilinde özel bir okulda, 2018-2019 eğitim-öğretim yılında öğrenim gören 138 öğrenciden oluşmaktadır. Öğrencilere araştırmacı tarafından hazırlanmış öğretim programı doğrultusunda 8 hafta boyunca programlama öğretimi gerçekleştirilmiştir. Çalışma kapsamında çoklu paradigma ve web tabanlı programlama dili olan JavaScript ve HTML dilleri ile öğretim yapılmıştır. Veri toplama aracı olarak blok temelli programlamaya ilişkin öz-yeterlilik algısı ölçeği, bilgisayarca düşünme beceri düzeyleri ölçeği, akademik başarı testi ve açık uçlu görüşme formu kullanılmıştır. Çalışmadan elde edilen bulgulara göre, öğrencilerin bilgisayarca düşünme ve problem

özme becerilerinde ön test ve son test puanları arasında anlamlı bir farklılık bulunmuştur. Araştırmada akademik başarının oldukça yüksek olduğu ancak cinsiyetin akademik başarıya anlamlı etkisinin olmadığı bulunmuştur. Ayrıca blok temelli programlamaya devam eden öğrencilerin bilgisayarca düşünme ve programlama öz-yeterlilik algılarıyla daha çok ilgili olduğu ortaya çıkmıştır. Ancak ön düşünme durumunun yalnızca bilgisayarca düşünme becerisi ile ilgili olduğu bulunmuştur. Yapılan öğrenci görüşmelerinde öğrencilerin metin temelli programlama ile komut yazmanın kendilerini motive ettiğini, etkinliklerin eğlenceli ve öğretici olduğunu belirtmişlerdir.

Anahtar Kelimeler: Programlama Öğretimi, Metin Temelli Programlama, Bilgisayarca Düşünme, Blok Temelli Programlama, Kodlama.

ABSTRACT

TEXT-BASED PROGRAMMING EDUCATION: INVESTIGATION OF COMPUTATIONAL THINKING SKILLS AND ACADEMIC ACHIEVEMENTS ON SECONDARY SCHOOL STUDENTS

Karaçam Duman, Nur Faden

Master's Thesis, Master's Program in Educational Technology

Supervisor: Assist. Prof. Burak ŞİŞMAN

January 2020, 119 pages

The aim of this study is to investigate the activities of text-based programming education in Information Technology and Software Course of 6th grades in terms' of students self-efficacy perceptions related to block-based programming, computational thinking skills and their academic achievement. For this purpose, text-based programming education activities were planned with secondary school students who had block-based programming self-efficacy perception. In the study, exploratory sequential pattern one of mixed research methods was used. The study group consisted of 138 6th grade students of a private school in İstanbul during the spring term of 2018-2019 academic year. Programming teaching was applied to students for 8 weeks in accordance with the curriculum prepared by the researcher. Multi-paradigm, JavaScript and HTML languages were used within the scope of teaching process of this study. As data collection tools; self-efficacy scale related to block-based programming, computational thinking scale related to skill levels, academic achievement test and open-ended questions interview form were used. According to

the findings, a significant difference have been found between the pre-test and post-test scores of the students' computational thinking and problem solving skills. In the research, it has been found that academic achievement has been quite high, but gender has not had a significant effect on academic achievement. In addition, it has been revealed that students who proceed to block-based programming was more interested in computational thinking and programming self-efficacy perceptions. However, it was found that the state of pre-thinking is only related to computational thinking ability. In the interviews, they have stated that the students motivate themselves with text-based programming and that the activities was fun and instructive.

Keywords: Programming Education, Text-Based Programming, Computational Thinking, Block-Based Programming, Coding.



Tüm ailem için, anneannem ve Fıstık anısına ...

TEŞEKKÜR

Bu tez çalışmasının geliştirilmesi sürecinde bilgi ve tecrübelerinden faydalandığım değerli danışmanım Dr. Öğr. Üyesi Burak ŞİŞMAN'a, tezimin savunmasında kıymetli yorum ve görüşleri için saygıdeğer jüri üyelerim Doç. Dr. Sevda KÜÇÜK ve Doç. Dr. Yavuz SAMUR'a katkılarından dolayı teşekkürlerimi sunarım.

Akademik kariyer hayatı ve tecrübesi ile örnek olan sevgili babam Zahit KARAÇAM'a, yoğun tez çalışma sürecimde yanımda olan sevgili annem Ayşe Gül KARAÇAM'a ve kıymetli yorumları ile destek olan canım kız kardeşim İrem KARAÇAM'a teşekkür ediyorum.

Akademik çalışmam sürecinde bana her anda destek olan ve hayallerimin peşinden gitmem konusunda beni motive eden sevgili eşim Recep DUMAN'a teşekkür ederim.

Akademik öğretimim sürecinde birlikte projeler geliştirdiğim ve aynı iş ortamını paylaştığım, tez geliştirme ve yazma aşamasında birbirimizi motive ettiğimiz sevgili arkadaşlarım Selin BOZKURTLAR PEÇE ve Mervenur YILDIZ'a teşekkür ederim.

İÇİNDEKİLER

ÖZ	iv
ABSTRACT	vi
TEŞEKKÜR	ix
İÇİNDEKİLER	x
TABLolar LİSTESİ	xv
ŞEKİLLER LİSTESİ	xix
KISALTMALAR LİSTESİ	xx
Bölüm 1	1
Giriş	1
1.1 Çalışmanın Amacı	8
1.2 Araştırma Soruları	9
1.3 Çalışmanın Önemi	9
1.4 Sayıtlılar	12
Bölüm 2	13
Alanyazın Taraması	13
Alanyazın Özeti	13
2.1 Programlama Öğretiminin Tarihçesi	13
2.1.1 Programlama ve Algoritma	13
2.1.2 Programlama Öğretiminin Tarihçesi ve Gelişimi	14
2.1.3 Dünya Ülkelerinde Programlama Öğretimi	15

2.1.4 Ülkemizde Programlama Öğretiminde Güncel Çalışmalar.....	17
2.1.5 Programlama Öğretiminde Cinsiyetin Etkisi.....	20
2.2 Blok Temelli Programlama Öğretimi.....	21
2.2.2 Blok Temelli Programlama Öğretiminde Araç Seçimi.....	22
2.3 Metin Temelli Programlama Öğretimi.....	26
2.3.1 Programlama Öğretiminde Pedagojik Yaklaşımlar.....	27
2.4 Metin Temelli Programlama Dillerinin Seçimi.....	31
2.5 Blok Temelli Programlamadan Metin Temelli Programlamaya Geçiş.....	33
2.6 Bilgisayarca Düşünme Becerileri.....	37
2.6.1 Bilgisayarca Düşünme Becerisini Geliştirme.....	41
Bölüm 3.....	43
Yöntem.....	43
3.1 Araştırma Modeli.....	43
3.1.1 Değişkenler.....	45
3.1.1.1 Bağımsız Değişken.....	45
3.1.1.2 Bağımlı Değişken.....	45
3.2 Çalışma Grubu.....	45
3.2.1 Öğrencilerin Demografik Özellikleri.....	47
3.3 Ders Materyalleri.....	48
3.3.1 Ders Planları.....	48
3.3.1.1 Uygulama Öncesi Hazırlık: Ders Planlarının Tasarımı.....	50
3.3.1.2 Çalışma Kapsamında Önemi Araştırılan Blok Temelli Programlama Ortamının Özelliklerinin İncelenmesi.....	52
3.3.1.3 Çalışma Kapsamında Kullanılan Metin Temelli Programlama Ortamının Özelliklerinin İncelenmesi.....	53
3.4 Uygulama Süreci.....	54
3.4.1 Uygulama.....	54
3.5 Verilerin Toplanması.....	56

3.5.1 Veri Toplama Araçları	56
3.5.1.1 Bilgisayarca Düşünme Beceri Düzeyleri Testi.	56
3.5.1.2 Blok Temelli Programlamaya İlişkin Öz-yeterlilik Algısı Ölçeği	57
3.5.1.3 Metin Temelli Programlamaya Yönelik Akademik Başarı Testi.....	58
3.5.2 Veri Toplama İşlemleri.....	60
3.5.2.1 Öğrenci Görüşmeleri.....	60
3.5.3 Verilerin Analizi	60
3.5.3.1 Nicel Veri Analizi	62
3.5.3.2 Nitel Veri Analizi.....	62
3.5.4 Geçerlik ve Güvenirlik.....	62
3.6 Sınırlamalar	63
Bölüm 4.....	64
Bulgular.....	64
4.1 Verilerin Normal Dağılımına Uygunluğun Belirlenmesi ve Betimsel İstatistikleri	64
4.1.1 Çalışma Gruplarının Bilgisayarca Düşünme Beceri Düzeyleri Ölçeğinin Normal Dağılıma Uygunluğunu Gösteren Veriler.....	65
4.1.2 Çalışma Gruplarının Metin Temelli Programlama Başarı Testinin Normal Dağılıma Uygunluğunu Gösteren Veriler.....	68
4.1.2 Çalışma Gruplarının Blok Temelli Programlamaya İlişkin Öz-Yeterlilik. 70	
4.2 Araştırma Problemlerine İlişkin Bulgular	73
4.2.1 Birinci Alt Probleme İlişkin Bulgular.....	73
4.2.1.1 Bilgisayar Düşünme Beceri Düzeyleri Ölçeği Betimsel İstatistikler.. 73	
4.2.1.2 Metin Temelli Programlama Başarı Testi Betimsel İstatistikler.....	75
4.2.1.3 Blok Temelli Programlamaya İlişkin Öz-Yeterlilik Algıları Ölçeği Betimsel İstatistikler.....	78
4.2.2 İkinci Alt Probleme İlişkin Bulgular	81
4.2.3 Üçüncü Alt Probleme İlişkin Bulgular	82
4.2.3.1 Üçüncü Alt Problemin Alt Boyutlarına İlişkin Bulgular	82
4.2.4 Dördüncü Alt Probleme İlişkin Bulgular.....	84

4.2.5 Beşinci Alt Probleme İlişkin Bulgular.....	94
4.2.6 Altıncı Alt Probleme İlişkin Bulgular.....	101
4.6.1 Öğrenci Görüşmelerine Ait Bulgular.....	101
4.6.1.1 Ders içi uygulama ve etkinliklere dair genel görüşme.....	102
4.6.1.2 Bilgisayarca düşünme beceri düzeylerine yönelik görüşme.....	102
4.6.1.3 Akademik başarıya yönelik görüşme.....	103
4.6.1.4 Blok temelli öz-yeterlilik algılarına yönelik görüşme.....	103
Bölüm 5.....	104
Tartışma ve Sonuçlar.....	104
5.1 Araştırmanın Birinci Alt Problemine Dair Tartışma ve Sonuçlar.....	104
5.2 Araştırmanın İkinci Alt Problemine Dair Tartışma ve Sonuçlar.....	106
5.3 Araştırmanın Üçüncü Alt Problemine ve Boyutuna Dair Tartışma ve Sonuçlar.....	107
5.4 Araştırmanın Dördüncü Alt Problemine Dair Tartışma ve Sonuçlar.....	110
5.5 Araştırmanın Beşinci Alt Problemine Dair Tartışma ve Sonuçlar.....	112
5.6 Araştırmanın Altıncı Alt Problemine Dair Tartışma ve Öneriler.....	114
5.4 Öneriler.....	116
5.4.1 Araştırmacılara Yönelik Öneriler.....	116
5.4.2 Uygulamacılara Yönelik Öneriler.....	117
5.4.3 Tasarlanan Ders Uygulamalarına Yönelik Öneriler.....	119
KAYNAKLAR.....	120
EKLER.....	137
A. 8. Hafta Bilişim Teknolojileri ve Yazılım Ders Planı.....	137
B. Bilgisayarca Düşünme Beceri Düzeyleri Ölçeği.....	140
C. Blok Temelli Programlamaya İlişkin Öz-Yeterlilik Algısı Ölçeği.....	141
D. Metin Temelli Programlama Başarı Testi.....	145

E. Metin Temelli Programlama Başarı Testi Belirtke Tablosu	155
F. Öğrenci Görüşme Formu.....	158
G. Bilgisayarca Düşünme Beceri Düzeyleri Ölçeği İzin Maili	159
H. Blok Temelli Programlamaya İlişkin Öz-Yeterlilik Algısı Ölçeği İzin Maili	160
I. Histogram ve Normal Kantil Grafikleri	161
K. Araştırmaya İlişkin Fotoğraflar	163
L. Ders Etkinliklerine İlişkin Fotoğraflar	166
ÖZGEÇMİŞ.....	167

TABLolar LİSTESİ

TABLolar

Tablo 1 Tek Grup Ön Test-Son Test Desen (Büyüköztürk vd., 2016)	43
Tablo 2 Çalışma Grubunda Yer Alan Öğrencilerin Sınıflarına göre Genel Akademik Başarı Puanları	46
Tablo 3 Çalışma Grubunda Yer Alan Öğrencilerin Cinsiyete Göre Dağılımı	47
Tablo 4 Çalışma Grubunda Yer Alan Öğrencilerin Sınıflarına Göre Dağılımı	47
Tablo 5 Araştırmanın Uygulama Süreci	55
Tablo 6 Metin Temelli Programlama Başarı Testi Pilot ve Son Haline Ait Madde İstatistikleri.....	59
Tablo 7 Araştırma Soruları ve Analiz Yöntemleri.....	61
Tablo 8 Bilgisayarca Düşünme Becerileri Ölçeğinin Ön test Sonuçlarının Cinsiyete Göre Kolmogorov-Smirnov (K-S) Ve Shapiro-Wilk Normal Dağılıma Uygunluk Testi Değerleri, Merkezi Eğilim Ve Dağılım Ölçüleri.....	65
Tablo 9 Bilgisayarca Düşünme Becerileri Ölçeğinin Son test Sonuçlarının Cinsiyete Göre Kolmogorov-Smirnov (K-S) Ve Shapiro-Wilk Normal Dağılıma Uygunluk Testi Değerleri, Merkezi Eğilim Ve Dağılım Ölçüleri.....	67
Tablo 10 Metin Temelli Programlama Başarı Testi Puanlarının Cinsiyete Göre Kolmogorov-Smirnov (K-S) Ve Shapiro-Wilk Normal Dağılıma Uygunluk Testi Değerleri, Merkezi Eğilim Ve Dağılım Ölçüleri.....	69
Tablo 11 Blok Temelli Öz-Yeterlilik Algısı Testinin Cinsiyete Göre Kolmogorov-Smirnov (K-S) Ve Shapiro-Wilk Normal Dağılıma Uygunluk Testi Değerleri,	

Merkezi Eğilim Ve Dağılım Ölçüleri.....	70
Tablo 12 Basit Blok Temelli Öz-Yeterlilik Algısı Testinin Cinsiyete Göre Kolmogorov-Smirnov (K-S) Ve Shapiro-Wilk Normal Dağılıma Uygunluk Testi Değerleri, Merkezi Eğilim Ve Dağılım Ölçüleri.....	71
Tablo 13 Karmaşık Blok Temelli Öz-Yeterlilik Algısı Testinin Cinsiyete Göre Kolmogorov-Smirnov (K-S) Ve Shapiro-Wilk Normal Dağılıma Uygunluk Testi Değerleri, Merkezi Eğilim Ve Dağılım Ölçüleri.....	72
Tablo 14 BDBOT Ve BDBST Düzeylerine Göre Ortalama Puanlarına İlişkin Betimsel İstatistikler.....	74
Tablo 15 Cinsiyete Göre BDBOT ve BDBST Puanlarının Analizi.....	74
Tablo 16 Metin Temelli Programlama Başarı Testinin Cinsiyete Göre Ortalama Puanlarına İlişkin Betimsel İstatistikler	75
Tablo 17 Metin Temelli Programlama Başarı Testi Puanlarının Analizi	76
Tablo 18 Öğrencilerin Blok Temelli Öz-Yeterlilik Algısı Anket Sorularının Dağılımı	78
Tablo 19 BTPOT, BBTPOT, KBTPOT Cinsiyete Göre Ortalama Puanlarına İlişkin Betimsel İstatistikler.....	80
Tablo 20 MTPBT Puanlarının Cinsiyete Göre t-Testi Sonuçları.....	81
Tablo 21 BDBST Ve BDBOT Puanlarının t-Testi Sonuçları	82
Tablo 22 BDBST Ve BDBOT Puanlarının Beceri Düzeylerine Göre t-Testi Sonuçları	83
Tablo 23 Varyans-Kovaryans Matrisinin Homojenliği Testi.....	84
Tablo 24 Levene F Testi Sonuçları	85
Tablo 25 BTPOT, MTPBT ve BDBST Sonuçlarının Korelasyon Analizi	86

Tablo 26 BTPOT, MTPBT ve BDBST Puanlarının, Blok Temelli Programlama Aracını Kullanmaya Devam Etme (Scratch'e Devam Etme) Ve Aracı Sevme (Scratch Sevme) Değişkenliğine Göre MANOVA Testi Sonuçları.....	87
Tablo 27 BTPOT, MTPBT ve BDBST Bağımlı Değişkenlerinin, Scratch'e Devam Etme Bağımsız Değişkenine Göre Birbirleri Üzerindeki Etkileri.....	87
Tablo 28 BTPOT Alt Boyutuna Ait Düzey Puanlarının, Blok Temelli Programlama Aracını Kullanmaya Devam Etme (Scratch'e Devam Etme) Değişkenliğine Göre MANOVA Testi Sonuçları	89
Tablo 29 KBTPOT Ve BBTPOT Bağımlı Değişkenlerinin, Scratch'e Devam Etme Bağımsız Değişkenine Göre Birbirleri Üzerindeki Etkileri	90
Tablo 30 KBTPOT ve BBTPOT Bağımlı Değişkenlerinin, Blok Temelli Programlama Aracını Kullanmaya Devam Etme (Scratch'e Devam Etme) Değişkenine Göre Grup Dağılım Sonuçları	90
Tablo 31 BDBST Alt Boyutuna Ait Düzey Puanlarının, Blok Temelli Programlama Aracını Kullanmaya Devam Etme (Scratch'e Devam Etme) Değişkenliğine Göre MANOVA Testi Sonuçları	91
Tablo 32 BDBST Alt Boyutları Bağımlı Değişkenlerinin, Scratch'e Devam Etme Bağımsız Değişkenine Göre Birbirleri Üzerindeki Etkileri	92
Tablo 33 BDBST Alt Boyutları Bağımlı Değişkenlerinin, Blok Temelli Programlama Aracını Kullanmaya Devam Etme (Scratch'e Devam Etme) Değişkenine Göre Grup Dağılım Sonuçları	93
Tablo 34 Varyans-Kovaryans Matrisinin Homojenliği Testi.....	94
Tablo 35 Levene F Testi Sonuçları	95
Tablo 36 BTPOT, MTPBT ve BDBST Puanlarının, Ön Düşünme Durumu Değişkenliğine Göre MANOVA Testi Sonuçları	95
Tablo 37 BTPOT, MTPBT ve BDBST Bağımlı Değişkenlerinin, Ön Düşünme	

Durumu Bağımsız Değişkenlerine Göre Ortalama Ve Standart Sapma Değerleri	96
Tablo 38 BTPOT, MTPBT ve BDBST Bağımlı Değişkenlerinin, Ön Düşünme Durumu Bağımsız Değişkenlerine Göre Birbirleri Üzerindeki Etkileri	97
Tablo 39 BDBST Alt Boyutuna Ait Düzey Puanlarının, Ön Düşünme Durumu Değişkenliğine Göre MANOVA Testi Sonuçları	98
Tablo 40 BDBST Alt Boyutu Bağımlı Değişkenlerinin, Ön Düşünme Durumu Bağımsız Değişkenlerine Göre Birbirleri Üzerindeki Etkileri.....	99
Tablo 41 BDBST Alt Boyutları Bağımlı Değişkenlerinin, Ön Düşünme Durumu Değişkenine Göre Grup Dağılım Sonuçları	100
Tablo 42 Öğrenci Görüşmelerine Dair Tema ve Kategoriler.....	101

ŞEKİLLER LİSTESİ

ŞEKİLLER

Şekil 1. Programlama dili öğretim yaklaşımları	6
Şekil 2. BTY dersi ile bilgisayar bilimi derslerinin kapsamı	17
Şekil 3. İlk programlama dili seçimine yönelik pedagojik yaklaşımlar.....	28
Şekil 4. Piaget bilişsel gelişim dönemlerine göre programlama öğretimi etkinlik önerileri	30
Şekil 5. Gagne'nin öğretim durumları modeli.	50
Şekil 6. Scratch 2.0 programı arayüz görünümü.....	53
Şekil 7. Notepad++ programı arayüz görünümü.....	54
Şekil 8.1. BDBOT'nin cinsiyete göre histogram grafiği.....	66
Şekil 8.2. BDBOT'nin cinsiyete göre normal kantil grafiği.	66
Şekil 8.3. BDBST'nin cinsiyete göre histogram grafiği.	68
Şekil 8.4. BDBST'nin cinsiyete göre normal kantil grafiği.....	68
Şekil 8.9. MTPBT'nin Betimsel Histogram Grafiği.	77
Şekil 8.5. MTPBT'nin cinsiyete göre histogram grafiği.....	161
Şekil 8.6. MTPBT'nin cinsiyete göre normal kantil grafiği.	161
Şekil 8.7. BTPOT'nin cinsiyete göre histogram grafiği.	162
Şekil 8.8. BTPOT'nin cinsiyete göre normal kantil grafiği.	162

KISALTMALAR LİSTESİ

AB	:Avrupa Birliđi
APOS	:Action Process Object Shema
ATCS	:Assessment and Teaching of 21st Century Skills
BBTPOT	:Basit Blok Temelli Programlama Öz-Yeterlilik Algısı
BDBOT	:Bilgisayarca Düşünme Becerileri Ön Testi
BDBST	:Bilgisayarca Düşünme Becerileri Son Testi
BİT	:Bilgi ve İletişim Teknolojileri
BT POT	:Blok Temelli Programlama Öz-Yeterlilik Algısı Ölçeđi
BT	:Bilişim Teknolojileri
BTY	:Bilişim Teknolojileri ve Yazılım
CSTA	:Computer Science Teachers Assosition
ECT	:Exploring Computational Thinking
GUI	:Graphical User Interface
IDE	:Integrated Development Environment
ISTE	:International Society for Technology in Education
KBTPOT	:Karmaşık Blok Temelli Programlama Öz-Yeterlilik Algısı
MEB	:Milli Eğitim Bakanlığı
MTPBT	:Metin Temelli Programlama Başarı Testi
NCREL	:North Central Regional Technology in Education Consortium
OECD	:Organisation for Economic Co-operation and Development
SLO	:Nationaal Expertisecentrum Leerplanontwikkeling
SPSS	:The Statistical Package for the Social Sciences
TUI	:Text-Based User Interface

Bölüm 1

Giriş

21. yüzyılın teknoloji ile birlikte şekillenmesi, bu dönemde yetişen bireylerin günlük yaşamlarında, iş, sağlık, eğitim ve eğlence ortamlarında bilgiye hızla ulaşmaları, bilgiyi paylaşmaları ve etkin kullanmaları için birtakım becerilere sahip olmaları gerekmektedir. Eğitimin en önemli amaçlarından biri, öğrencilerin günlük hayatlarını kolaylaştırmak ve onları geleceğe hazırlamaktır (Yalçın, 2018). Bu yüzden yeni nesil öğrencilerin eğitim ve öğretim faaliyetleri bilgisayar ve teknolojik araçlar ile yürütülmektedir (Erümit ve Berigel, 2018).

Bu dönemde yetişen öğrencilerin bilgiyi ezberlemesi değil; bilgiyi kullanarak yenilikçi ve fark yaratan ürünler ortaya çıkarabilmesi bilişim ve diğer sektörler için önemlidir (Baştemur-Kaya, 2018). Dolayısıyla günümüz ve gelecek meslekleri beceri odaklı ve sürekli gelişim isteyen bir hayat üzerine kurulacaktır. Bu yüzden eğitim sistemleri 21. yüzyıl becerilerini destekleyen programlar ile hazırlanmaktadır (Hamarat, 2019).

Henüz 21. yüzyılın ilk çeyreğinde, bireylerin beceri odaklı olmalarının önem arz ettiği kurumlar tarafından açıklanmıştır Her yüzyılın beceri setleri bazı kurum ve kuruluşlar tarafından betimlenmiştir. (Hamarat, 2019; Saygıner, 2017). Partnership for 21st Century Skills (P21) isimli kuruluşa göre 21. yüzyıl becerileri üç kategori olarak belirlenmiştir (P21, 2017). Bunlar; öğrenme, okuryazarlık ve gündelik hayatla ilgili becerilerdir. Computer Science Teachers Assosition (CSTA) ve International Society for Technology in Education (ISTE) ile işbirliği içinde çalışan SLO (Nationaal Expertisecentrum Leerplanontwikkeling) ve Kennisnet ile birlikte geliştirdiği 21. yüzyıl beceri modelinde temsil edilen 11 yetkinlik bulunmaktadır. Bunlar; eleştirel düşünme, yaratıcılık, problem çözme, temel Bilişim Teknolojileri (BT) becerileri, bilgi okur yazarlığı, bilgisayarca düşünme ve medya okuryazarlığı gibi becerileri içermektedir (Pijpers, 2017).

Bilginin hızla yayıldığı ve paylaşıldığı bu dönemde öğrencilerin öncelikle bilgi

okur yazarı olması gerekli görülmektedir (Polat ve Odabaş, 2008). Bilgi okuryazarlığı becerisinden sonra teknolojiyi kullanarak bilginin oluşturulması, iletilmesi, saklanması ve paylaşılması için gerekli olan beceri bilgi ve iletişim teknolojileri (BİT) okuryazarlığıdır. BİT okuryazarlık becerisi, 21. yüzyılın temel eğitim hedeflerinden biri olarak görülmektedir (Günay ve Şişman, 2019). Bu yüzyılda yetişen öğrencilerin teknoloji ile iç içe olmalarından dolayı bilgilerini, düşünme becerileri kazanmalarını desteklemek büyük önem taşımaktadır (Doğan, 2015). Bu kapsamda, 21. yüzyıl öğrencilerinin sahip olmaları gereken becerilerden biri olduğu ifade edilen bilgisayarca düşünme giderek önem kazanmaktadır (Wing, 2006).

Bilgisayarca düşünme, 1960'lı yıllardan itibaren bilgisayar bilimleri tarihinde çok önemli bir yere sahip olmuştur (Denning, 2009). Bilgisayarca düşünme, temel kavramlarla oluşturulan bir analitik düşünme şekli olarak problem çözmek için sistemleri tasarlamayı ve insan davranışlarını anlamayı amaçlayan bir yaklaşım sergilemektedir (Wing, 2006). Başlangıçta, algoritmik bir beceri olarak ifade edilen bu kavram, ilerleyen teknolojik gelişmeler ile yayılmıştır. Bilgisayarca düşünme, günümüzde çeşitli üst düzey ve temel becerilerin birleşimi olarak ifade edilmektedir (Denning, 2009; Wing, 2006).

Google Exploring Computational Thinking ([ECT], 2016) bilgisayarca düşünmeyi, problem çözme sürecindeki mantıksal veriyi analiz etme ve algoritmalar kullanarak çözüm üretme olarak tanımlanmaktadır. CSTA (2017) göre yeni bilgilerin örgütlenmesi sırasında problem çözmek ve sistem tasarlamak için bir düşünme şeklidir. Bu tanımlar incelendiğinde, bilgisayarca düşünmenin birçok alt beceriler ile var olduğu anlaşılmaktadır. ISTE (2019) göre algoritmik düşünme, yaratıcı düşünme, işbirlikli öğrenme ve iletişim becerileri gibi alt beceriler olmadan bilgisayarca düşünmenin net bir şekilde anlatılamayacağı açıklanmıştır. Bu araştırmalara göre bilgisayarca düşünmenin daha iyi anlaşılması için alt becerilerin önemli olduğunu belirtmek gerekir.

Wing (2006) bilgisayarca düşünmeyi 6 adım ile tarif etmiştir; (a) büyük zor problemlerin görece kolay ve çözülebilir problemler olarak yeniden formüle edilmesi, (b) özinelemeli düşünme, (c) soyutlama ve çözümlenme, (d) değişmezleri kullanarak sistemin davranışını tarif etme, (e) buluşsal muhakemeyi kullanma, (f) bilgisayar bilimcisi gibi düşünmek ve çoklu seviyedeki soyutlamaları düşünmek olarak belirtmiştir. Bu becerilerin her seviyeden öğrenci için gerekli olduğu savunulmaktadır.

Bilgisayar bilimi ise sadece araç kullanımını değil; matematik ve enformatik kavramlarını uygulama süreçlerini de içermektedir (Kert, 2018a). Ancak bilgisayar bilimi sadece yapay süreçleri incelemeyi, doğal süreçleri de düşünmek gerekmektedir. Bu yüzden bilgisayarca düşünme, bilgisayar bilimini açıklamaya yetmemektedir çünkü bilgisayarca düşünme, bilgisayar biliminin temel pratiklerinden biridir; sadece bilgisayar bilimine özgü değildir (Denning, 2013, 2017).

ISTE ve CSTA birlikte çalışarak bilgisayarca düşünmenin ne olduğunu belirlemek ve sınırlamak için problem çözme sürecini tanımlamışlardır. Onların standartlarına göre algoritmik düşünme, soyutlama, problemi bileşenlerine ayırma, otomasyon, problem çözme, veri analizi ve temsili, model ve simülasyonlar gibi kavram ve beceriler ele alınmıştır (ISTE, 2011).

Bilgisayar biliminde, bilgisayarın programlanmasından daha önemli olan ise soyutlama yeteneğini kullanan BT uzmanlarının yetiştirilmesidir. Bu uzmanların en önemli becerisi ise düşünme becerilerini kullanarak bilgisayar programlarını alt bileşenlerine ayrıştırabilmesidir (Lu ve Fletcher, 2009; Wing, 2006). Dolayısıyla, bilgisayarca düşünme kavramlarının gelişmesi sürecinde programlama çok etkili bir araçtır (Bocconi, Chiocciariello, Dettori, Ferrari ve Engelhardt, 2016; Lu ve Fletcher, 2009).

Bilgisayarca düşünmenin temelleri her ne kadar Wing (2006) ile anılsa da bu terimi ilk kullanan Seymour Papert'tır (Papert ve Harel, 1991). Papert (1980) geometrik problemlerin çözümünde bilgisayarların kullanılması için problemlerin çözümlerinde bilgisayarca düşünmenin açıklanması ve bilginin nasıl yapılandırıldığı üzerine çalışmalar yapmıştır. Programlama aracılığı ile öğrenciler matematiksel ve mantıksal düşünme becerilerini geliştirebilmektedir. Dubinsky (1995)'e göre birçok insan programlama öğrenmeye isteklidir fakat belli bir grubun matematikte başarılı olduğu görülmüştür. Papert ve arkadaşları tarafından geliştirilen LOGO programlama dili ve SET programlama diline bağlı geliştirilen ISETL, matematik eğitiminde Piaget'nin yansıtıcı soyutlama kavramını temel alarak bilgisayarca düşünme becerisinin nasıl geliştiğine yönelik araştırmalarda kullanılmıştır. Yakın dönemde ise Action Process Object Shema (APOS) kuramı ile bilgisayarca düşünme süreçleri açıklanmaktadır (Çetin ve Toluk-Uçar, 2018). APOS kuramı matematiksel kavramların anlamlandırılması sürecinin inşa edilmesidir (Gölbucaklı, 2016).

Ülkemizde bilgisayarca düşünme becerileri Bilişim Teknolojileri dersi programlarına çeşitli güncellemelerle eklenmiştir (Milli Eğitim Bakanlığı [MEB], 2016; Saritepeci ve Durak, 2017). Programlardaki güncellemeler göz önüne alındığında programlama eğitiminin ön plana çıktığı görülmektedir. (Bocconi vd., 2016). Bu duruma bağlı olarak, teknoloji temelli üretimin artması için programlama ile bilgisayarca düşünme desteklenmektedir (Boechler, Artym, DeJong, Carbonaro ve Stroulia, 2014; Lee, Martin ve Apone, 2014). Programlama, aynı zamanda problemlerin çözümünde, problemlerin analizini ve ayrıştırılmasını da ele almaktadır (Bocconi vd., 2016). Problem çözümü, bilgi-işleme birimi tarafından problemin nasıl çözülmesi gerektiğini çözen kişi tarafından bilgisayara anlatılması üzerine yoğunlaşmaktadır. Bilgisayarların yeteneklerine göre problemi farklı bakış açıları ile çözümlememize yardımcı olur ve aslında bilgiyi analiz etmemizde yardımcıdır (Barr ve Stephenson, 2011).

Programlama 1950 yıllarından itibaren farklı tanımlamalarla günümüze kadar gelmiştir. Blackwell (2002) göre, matematiksel formülleştirmeler bilgisayar terimlerine göre yapılırken, matematiksel tanımların bilgisayarın kendi işleyebileceği kodlamalar halinde şekillenmesi olarak tanımlamıştır. İnsan ve bilgisayarın arasındaki iletişimi sağlaması için gerekli olan kodlamaların uygun sıralama sistemi ile doğru bir şekilde yazma işlemi programlamadır. Psychology of Programming kaynağında entelektüel doğanın insanların işlerinde yardımcı olacağı ve zaman zaman da yerini alacağı makina davranışlarının tasarımı şeklinde ifade edilmiştir (Hoc, Green, Samurçay ve Gilmore, 1990).

Günümüzde çok sayıda programlama dili bulunsa da, ilk programlar makine dili ile geliştirilmekteydi. Makine dili ile programlama yaparken, işlemci düzeyinde kodların anlaşılabilmesi için herhangi bir derleyici programa ihtiyaç duyulmaktadır. Makine dili ile programlama, bilgisayar işlemcisi düzeyinde hız açısından avataj sağlasa da hızlı kod yazmayı zorlaştırmaktadır. 1960'lardan sonra programlama dillerinde çeşitlenmeler olmuştur. Güncel teknolojiye kullanılan birçok programlama dili, Pascal ve C gibi programlama dilleri temel alınarak geliştirilmiştir (Gülbahar ve Karal, 2018). Bu yüzden de yazım dili olarak birbirinden farklı programlama dilleri ortaya çıkmıştır (Baştemur-Kaya, 2018). İlk yüksek seviyeli derleyiceye sahip olan programlama dilleri Fortran ve Algol, yazılmış kodları derleyerek ve makine diline çevirerek çalıştırmışlardır (Metcalf, 1985). Bu programlama dilleri daha sonradan

geliştirilen Cobol, Basic, Logo, C ve Pascal gibi dillerin temellerini oluşturmuşlardır. Basic programlama dili, genel amaçlı ve kullanıcı dostu bir programlama dilidir ve o dönemde öğrencilerin farklı alan derslerinde de bilgisayar kullanabilmeleri için geliştirilmiştir. Günümüzde daha çok nesne yönelimli bir dil olarak da kullanılmaktadır (Erümit ve Berigel, 2018).

İnternetin buluşu ile programlama tarihinde yeni bir dönem başlamıştır. İnternet programcılığının ortaya çıkışı ile eşzamanlı ve dağıtık programlamaya başlanmıştır. İnternet programcılığında, birden fazla programlama dili, paradigması ve aracı beraber kullanılmaktadır. İnternet tarayıcı ile çalışan bu programlarda nesne yönelimli ve kendi derleyicisine sahip dillerin (Delphi, C# vb.) yanı sıra kendi sunucuna sahip diller (ASP, PHP vb.) ve sadece yorumlayıcıya sahip Script (betik) diller (JavaScript, VBScript v.) birlikte kullanılabilir. Bu dillerin ortak noktası ise fonksiyonel bir yapıya sahip olmalarıdır. Söz diziminin sade ve esnek yapısı ile Python en popüler programlama dillerinden biri haline gelmiştir (Erümit ve Berigel, 2018).

1994 yılında ilk web tarayıcısının yayınlanması ile bu ortamda yapılabileceklerin seviyesi de artmaya başlamıştır. Aynı yıl içerisinde HTML'in ötesine geçen C tabanlı bir Script dili olan PHP sunucu taraflı bir dil olarak günümüzde pek çok programcı tarafından halen dinamik web siteleri oluşturmak için kullanılmaktadır. 1995 yılında Brenda Eich tarafından JavaScript dili geliştirilmiştir. Özellikle web tarayıcısına entegre olan Script dili web geliştiricileri için ilk çıktığı zamandan itibaren önemli bir dil haline gelmiştir. Günümüzde HTML ve CSS ile birlikte web temel teknolojilerini oluşturmaktadır (Gülbahar ve Karal, 2018).

Günümüz programlama dilleri ve araçları; bilgisayar işlemcisine aynı anda eşzamanlı erişebilme, fonksiyonel ve statik programlama, Script kullanımı ve sezgisel kod tamamlama gibi özellikler sayesinde yaygınlaşmıştır. Bu durumda bilişim alanları uzmanları ve programcı sayısı artmış ve açık kaynak kodlu programlama dilleri ile programlama ortamlarının da gelişimi artmıştır. Dolayısıyla bu alanlarda nitelikli uzmanların ihtiyacı da artmaktadır (Erümit ve Berigel, 2018).

Tüm ülkeler eğitim sistemlerinde, program geliştirebilecek ve anlayabilecek nesiller için kendi müfredatlarının içeriklerini geliştirmektedirler. Sadece uzman yetiştirmekle kalmayıp, bilişsel becerilere katkısı olduğu gerekçesi ile anaokulundan üniversiteye kadar programlama öğretimi verilmektedir. Programlama öğrenimi

sürecinde, öğrencilerin soyutlama ve düşünme becerilerinin geliştiğini, olay ve olguların ne olduğundan daha çok nasıl geliştiğini sorguladıklarını ve bilişsel gelişim gösterdiklerini belirtmiştir (Papert, 1980). Programlama becerilerinin ve bilgisayarca düşünme unsurlarının kazandırılması amacıyla çeşitli eğitsel programlama dilleri seçilmektedir (Şendurur, 2018).

Programlama dili öğretiminde doğru pedagojik yaklaşım ve öğretim tasarımı, programlama dili aracı, öğretmen ve öğrenci alan yeterliliği, hedefe odaklı ders plan ve içerikleri ile verimliliği artıracak önemli unsurlar arasında yer almaktadır. Düşünme becerilerinin gelişmesi için programlama öğretimindeki teknik unsurlar dışında, alan öğretmenlerinin farkındalıklarının oluşmuş olması gerekmektedir. Öğrencilerin sadece bir kod satırını yazması öğrenme hedefi olmamalıdır. Programlama öğretimi disiplinlerarası bir boyutta, derin bakış açısı ve bilgisayarca düşünme becerileri hedeflenerek yapılmalıdır (Kert, 2018a).

Programlama öğretiminde ortaya konulan kazanımların temelinde problem çözme ve düşünme becerileri, yaş grubunun bilişsel gelişim dönemlerine göre öğretim hedeflerine uygun olarak seçilmesi gerekmektedir (Kert, 2018b). Programlama dili öğretim yaklaşımları aşağıda Şekil 1’de yer almaktadır.

PEDAGOJİ			
Öğrenme Modelleri ve Öğretimsel Teknikleri	Bağlam	Programlama Dilleri	Öğrenci Katılımı
Öğrenme modelleri ve öğretim programı yapıları	Fiziksel programlama pedagojisi Oyun geliştirme pedagojisi	Blok ve Metin temelli programlama pedagojisi	Eşli programlama
Öğretimsel teknikler ve öğretim stratejileri	Bilgisayarsız uygulama pedagojisi Öğretim programları arası pedagoji	Diğer programlama dili fırsatları	Diğer öğrenci katılım uygulamaları

Şekil 1. Programlama dili öğretim yaklaşımları (Waite, 2017).

Pedagojik yaklaşımlar; öğrenme modelleri ve öğretimsel teknikleri, bağlam, kullanılan programlama dilleri ve öğrenci katılım temaları altında ele alınmıştır (Waite, 2017). En çok tercih edilen yaklaşım olarak “araç odaklı yaklaşımlar”

programlama öğretiminde önemli bir yer tutmaktadır. Bu yaklaşıma göre, öğrenme sürecinde kullanılan araca göre pedagojik yaklaşımlar farklılık gösterebilir (Kert, 2018b). Programlama öğretiminde araç odaklı yaklaşımlar dört farklı uygulamadan oluşmaktadır. Bunlar; bilgisayarsız programlama, blok temelli programlama, metin temelli programlama ve fiziksel programlama olarak ele alınmaktadır (Kert, 2018b).

Bilişsel gelişim sürecine uygun programlama aracı seçilmezse, erken yaşta programlama öğretimi bilişsel yükü artırmakta ve programlama dilinin öğrenimi zorlaştırmaktadır (Kert, 2018b). Programlama ile ilk kez tanışan birçok öğrenci metin temelli programlama dilinin söz dizimi kavramları ve kuralları, fonksiyon yapıları veya uygulamalarına yönelik problemlere blok temelli programlama ortamları ile çözüm bulunmaktadır (Kurihara, Sasaki, Wakita ve Hosobe, 2015). Blok temelli programlama, söz dizimi komutların yerine hazır görsel bloklar yapısında programlama terimlerinin sürükle-bırak ya da yap-boz yoluyla bir araya getirilmesiyle oluşan programları ifade etmektedir (Yükseltürk ve Üçgül, 2018). Dolayısıyla, programlama öğretimine yeni başlayanlar için basit arayüz kullanımı olan oyun tabanlı araçlar olan blok temelli programlama dilleri önerilmektedir (Kalelioğlu ve Gülbahar, 2014). Blok temelli programlama dilleri, programlama öğretimine giriş için seçilmektedir. Ancak, bu araçlar geleneksel metin temelli programlama dillerinde yazılım geliştirmek için yeterli kavramlara sahip değildir (Kandemir, 2018b; Kert, 2018b). Blok temelli programlama dili ile temel programlama becerileri oluşmuş öğrencilerin, metin temelli programlama araçları ile program geliştirebilecekleri belirtilmiştir (Dorling ve White, 2015).

Metin temelli programlama öğretiminde, programlama dili seçilirken söz dizimi yapıları ve kurallarına dikkat edilmesi gerekmektedir. Seçilen programlama dili bilişsel gelişim sürecine uygun olursa öğrenme kazanımlarına ulaşılabilmektedir (Kandemir, 2018b). Programlama ile ilk tanışan öğrencilerin hazır kod blokları kullanarak hedef odaklı problemleri çözdükleri programlama araçlarının motivasyonlarını olumlu etkilemektedir (Maloney, Resnick, Rusk, Pepler ve Kafai, 2008). Bu nedenle blok temelli programlama araçları öğrenme kazanımlarına ulaşmak için metin temelli programlama öğretiminden önce tercih edilmektedir. Blok temelli araçların kullanımında problem çözme ve düşünme becerilerinin desteklendiği ders etkinlikleri hazırlanması gerekmektedir (Yükseltürk ve Altıok, 2018). Weintrop ve Wilensky (2015)'e göre blok temelli programlama araçlarının kısıtlı yapıda olması

yaratıcılık ve problem çözme becerilerini olumsuz etkileyebilmektedir. Bu nedenle bilgisayarca düşünme becerilerinin geliştirilmesi sürecinde metin temelli programlama araçları kullanılabilir (Nouri, Zhang, Manilla ve Noren, 2019).

Metin temelli programlama öğretiminde programlama dilinin giriş-çıkış işlemlerinin kolay, söz diziminin okunabilir yapısı, küçük-dikey kümeler özelliği ve diğer programlama dillerinin yapılarından açık olarak ayırık olmasına dikkat edilmelidir (Vujosevic-Janicic ve Tosic, 2008). Bu özelliklerin yanında hedef kitlenin yaşına, bilişsel seviyesine, bilgisayarca düşünme ve programlama becerisine yönelik kazanımlara göre programlama ortamı ve dili değişebilmektedir (Kandemir, 2018b). Kaplumbağa grafiklerini temel alan kütüphanesi ve programlama kavramları ile Small Basic bilgisayarca düşünme ve programlama becerileri için kullanılmaktadır (Papert ve Harel, 2017). Özellikle son yıllarda nesne tabanlı ve betik bir dil olan Python ve olan JavaScript ile dinamik web uygulamalar geliştirmek oldukça popülerdir (Aytekin, Çakır, Yücel ve Kulaözü, 2018; Baştuğ, 2019). Bu programlama dilleri, temel programlama ve bilgisayarca düşünme becerilerini desteklerken öğrencilerin kariyer planları için de faydalı olmaktadır (Jenkins, 2004). Metin temelli programlama dilleri ile programlama becerileri kazanılırken problem çözme ve soyutlama becerileri de gelişmektedir (Kandemir, 2018b). Aynı zamanda geniş çaplı projeler için elverişli programlama dilleri olmalarından dolayı hem ilk ve orta öğretimde, hem de yüksek öğretim programlarında metin temelli programlama dillerinden faydalanılmaktadır (Şendurur, 2018).

1.1 Çalışmanın Amacı

Bu çalışmanın amacı, 6. sınıf öğrencilerinin Bilişim Teknolojileri ve Yazılım (BTY) dersi kapsamında metin temelli programlama öğretiminde bilgisayarca düşünme becerilerine ve akademik başarılarına etkisini incelemektedir. Bu amaç doğrultusunda metin temelli programlama öğretiminde öğrencilerin blok temelli programlama öz-yeterlilik algısının bilgisayarca düşünme becerilerine ve akademik başarılarına etkisi araştırmanın temelini oluşturmaktadır.

Yapılan bu çalışmanın, ortaokul düzeyindeki öğrencilerin metin temelli programlama öğretimine yönelik olarak yapılacak diğer araştırmacılara yol göstermesi, kaynaklık etmesi ve öğrencilerin bilgisayarca düşünme becerilerine ve akademik başarılarına doğru yönde katkı sağlanması için bulguların paylaşılması, bu

yolda harcanan emek ve zamanın daha nitelikli kullanılması hedeflenmiştir.

1.2 Araştırma Soruları

1. Ortaokul öğrencilerinin blok temelli programlama öz-yeterlilik algıları, bilgisayarca düşünme becerileri ve akademik başarıları ne düzeydedir?
2. Metin temelli programlama öğretiminde öğrencilerin akademik başarıları cinsiyete göre anlamlı farklılık göstermekte midir?
3. Metin temelli programlama öğretiminde, öğrencilerin bilgisayarca düşünme becerilerine göre ön test ve son test puanları arasında anlamlı farklılık var mıdır?
 - a. Bilgisayarca düşünme becerileri alt boyutlarına göre ön test ve son test puanları arasında anlamlı farklılık var mıdır?

Ortaokul öğrencilerinin,

4. Blok temelli programlama öz-yeterlilik algıları, bilgisayarca düşünme becerileri ve akademik başarıları, blok temelli programlama aracını kullanmaya devam etme durumuna göre anlamlı farklılık göstermekte midir?
5. Blok temelli programlama öz-yeterlilik algıları, bilgisayarca düşünme becerileri ve akademik başarıları, ön düşünme düzeyine göre anlamlı farklılık göstermekte midir?
6. Tasarlanan ders etkinliklerine, bilgisayarca düşünme becerilerine ve blok temelli programlama öz-yeterlilik algılarına yönelik betimsel görüşleri nelerdir?

1.3 Çalışmanın Önemi

21. yüzyılda yaşanan teknolojik dönüşümler ile birlikte günlük hayatımızda bilgi kullanımının hızlanması, yetişen bireylerin sahip olması gereken becerileri de etkilemiştir. Öğrencilerin günlük hayatlarını kolaylaştırmak ve geleceğe hazırlamak ise eğitimin amaçlarından biridir (Yalçın, 2018). Dolayısıyla belirli kuruluşlar tarafından beceri odaklı veyahut becerileri destekleyici olarak güncellenen öğretim programları yayınlanmaktadır. P21, CSTA, ISTE, SLO ve Kennisnet gibi kuruluşlar, hazırladıkları öğretim programlarında özellikle bilgisayarca düşünme ve programlama becerisine dikkat çekerek bu becerilerin BİT okuryazarlığı ile geliştirileceğini ifade etmişlerdir (P21, 2017; CSTA, 2009; ISTE, 2019; SLO, 2019). Buna göre bilgisayarca

düşünme kavramlarının gelişmesi sürecinde programlama etkili bir araçtır (Bocconi vd., 2016; Lu ve Fletcher, 2009). Wing (2006), bilgisayarca düşünmenin yalnızca bilgisayar bilimlerine ait olmadığını da belirtmiştir. Ancak bilgisayarca düşünme, bilgisayar biliminin temel pratiklerinden biridir. Bilgisayarca düşünmenin alt boyutları ele alındığında soyutlama, algoritmik düşünme ve problem çözme gibi becerilerin bilgisayar bilimi programlarında yer aldıklarını görmek mümkündür.

Programlama, insanların kodları belirli sıralama sistemi ile bilgisayara tanıtmaya işlemidir. Bu durum bir insanın başka birine yaptırmak istediği eylemi doğru sırada vermesi ve doğru sözcükler kullanması ile eş değerdir (Erümit ve Berigel, 2018). Programlama dillerini geliştiren alan uzmanları sadece problemi çözmeye yönelik dili geliştirirler; bu sırada yaratılan yeni dili öğrenecek olanlar göz ardı edilebilir (Kandemir, 2018a). Özellikle programlama öğretiminde acemi olan öğrencilerin programlama komutlarının arasında kayboldukları görülmektedir (Yıldız-Durak, 2018). Bu sorunların çözümü için programlama araçlarının çeşitliliği artırılmıştır (Trees, 2010).

Günümüzde bilişsel gelişim dönemlerine göre çeşitli programlama dilleri bulunmaktadır (Kandemir, 2018b). Çeşitli programlama dillerinin, algoritmik ve eleştirel düşünme, problem çözme becerileri gibi niteliklerin üzerinde olumlu etkileri olduğu belirtilmektedir (Wang, Mei, Lin, Chiu ve Lin, 2009). Papert (1980) bilgisayarca düşünmenin alt becerilerini, programlama ile kolaylıkla verebileceğine inanmaktaydı. Bu sebeple programlama öğretimi planlanırken, bilgisayarca düşünme ve programlama beceri kavramlarını kapsayan ders planlarının ve öğretim programlarının hazırlanması ve uygulanması önemli görülmektedir. Erken yaşlarda bu becerilerin kazandırılması amacıyla programlama öğretimine işlem öncesi dönemde başlanmaktadır (Kandemir, 2018b). Bu doğrultuda programlama öğretiminde acemiler için basit arayüze sahip, görselleştirilmiş kod yapısı olan ve oyun tasarımı odaklı programlama araçları seçilmektedir (Kalelioğlu ve Gülbahar, 2014). Programlama kavramlarının soyut ve kuramsal kavramlarının öğretiminde zorlanması nedeniyle blok temelli programlama araçları seçilmeye başlanmıştır (Yükseltürk ve Üçgül, 2018). Bu araçlar, başlangıç seviyesindeki acemi programlamacılar için programlama öğretiminde motivasyon sağlamaktadır (Maloney vd., 2008; Kandemir, 2018a, 2018b). Buna rağmen blok temelli programlama araçlarında, programlama kavramlarının ve kod yapılarının kısıtlı olması sebebiyle problem çözme becerisini

olumsuz etkileyebilmektedir (Weintrop ve Wilensky, 2015).

Günlük hayat problemlerinin çözümünde ise metin temelli programlama dilleri daha aktif bir alandadır (Kandemir, 2018a). Bilgisayarca düşünme ve metin temelli programlamada soyutlama daha ön plandadır; soyutlama becerisi problem çözmenin temelini oluşturmaktadır (Bayman ve Mayer, 1988; Papert, 1980; Wing, 2006). Ancak metin temelli programlama dilinin söz dizimi kuralları ve hedef kitlenin bilişsel gelişim dönemine uygun olması pedagojik olarak önem taşımaktadır (Kandemir, 2018b; Kert, 2018b). Blok temelli programlama öğretimi ile temel programlama kavramları ve becerilerinin gelişmesinden sonra metin temelli programlama dili ile bilişsel yük oluşmadan daha sağlıklı program geliştirebileceklerdir (Kert, 2018b; Dorling ve White, 2015).

Sonuç olarak bilgisayarca düşünme becerileri ile programlama öğretimi arasında bir ilişki vardır (Papert, 1980). Bilgisayarca düşünme becerilerinin geliştirilmesi için programlama öğretimine önem verilmesi gerekmektedir. Ülkemizde, BTY derslerinin öğretim programları gereği programlama öğretimi ortaokul düzeylerinde yapılmaktadır (MEB, 2017). Son beş yıldır ise ders dışı kulüp etkinlikleri, kursları ve maker kulüpleri gibi ortamlarda programlama öğretimi verilmektedir. Bununla birlikte ülkemizde her yaş seviyesinde programlama öğretimine yönelik Alanyazında çalışmaların bulunduğu söylenebilmektedir. Ancak bu çalışmaların yoğunlukla blok temelli programlama öğretiminde yapıldığını söylemek mümkündür. Metin temelli programlama öğretimi ile ilgili Alanyazın çalışmaları henüz başlangıç aşamasındadır. Bilgisayarca düşünme becerileri alanında ise Alanyazın çalışmaları yeni olmakla birlikte gün geçtikçe artmaktadır.

Bilgisayarca düşünme ve programlama becerisine katkı sağladığı bilinen metin temelli programlama dillerinin, soyut işlemler dönemine uygun pedagojik yaklaşım ve öğretim programı ile hazırlanan etkinliklere fazlasıyla ihtiyaç duyulmaktadır. Bu doğrultuda araştırma sürecinde metin temelli programlama uygulamalarının öğrenciler üzerindeki etkilerini belirlemek için hazırlanmış olan ders planları, bilişsel gelişim dönemine uygun ve bilgisayarca düşünme alt becerilerini geliştirmeye yönelik süreci sunmaktadır. Böylelikle araştırma süreci boyunca, metin temelli programlama öğretimine yer verilmiş aynı zamanda bilgisayarca düşünme becerilerini geliştirmeye yönelik katkı sağlayacağı düşünülmekte olan 8 farklı ders planı hazırlanmıştır. Aynı zamanda, BTY dersi kapsamında programlama ile ilgili kazanımlara yönelik metin

temelli programlama ile uygulamalar geliřtirmesi ve bunu öğretim planı řeklinde sunması bakımında önemli kılmaktadır. Bu çalıřma ile blok temelli programlama öğretiminden sonra planlanacak olan metin temelli programlama öğretiminin, bilgisayarca düşünmeye ve akademik başarılarına etkilerini ortaya koyarak ilgili Alanyazında etkili bir yer tutacaktır. Ortaya çıkan sonuçların bu alanda çalıřma yapmayı hedefleyen arařtırmacılara yol gösterecek önemli bir kaynak olduđu düşünölmektedir.

1.4 Sayıtlar

Bu çalıřmanın sayıtları ařađıdaki gibidir:

1. Çalıřmada kullanılan ölçek ve testlerin geçerli ve güvenilir olduđu,
2. Çalıřma grubunda bulunan öğrencilerin, blok temelli öz-yeterlilik algılarını ve hazırbulunluřluklarını belirlemek amacıyla “Blok Temelli Öz-Yeterlilik Algısı Ölçeđi”ni, bilgisayarca düşünme becerileri düzeylerini tespit etmek amacıyla yapılan “Bilgisayarca Düşünme Becerileri Testi”ni ve metin temelli programlama konusundaki akademik başarılarını öğrenmek amacıyla yapılan “Metin Temelli Programlama Başarı Testi”ni samimiyetle cevapladıkları,
3. Kontrol altına alınamayan deđiřkenlerin arařtırma grubundaki öğrencileri eřit bir řekilde etkilediđi,
4. Arařtırma için seçilen yöntem, arařtırmanın amacına, konusuna ve problemine uygun olduđu varsayılmıřtır.
5. Bu arařtırmadaki öğrenciler 2017-2018 eğitim-öđretim yılında blok temelli programlama araçlarının kullanımını bildikleri varsayılmıřtır.
6. 2018-2019 eğitim-öđretim birinci döneminde 2 ders saati eđitsel metin temelli programlama dili CodeCombat ile eđitsel hedef odaklı metin temelli programlama öđretimi aldıkları varsayılmıřtır.

Bölüm 2

Alanyazın Taraması

Alanyazın Özeti

Bu bölümde programlama ve algoritma, programlama öğretiminin tarihçesi ve gelişimi, dünya ülkelerinde programlama öğretimi, ülkemizde programlama öğretimindeki gelişmeler, blok temelli programlama, metin temelli programlama, blok temelli programlamadan metin temelli programlamaya geçiş, bilgisayarca düşünmeye yönelik araştırmalar olmak üzere altı bölüm bulunmaktadır.

2.1 Programlama Öğretiminin Tarihçesi

2.1.1 Programlama ve Algoritma. Programlama, belirli kurallara uygun olarak hazırlanmış algoritmalarla, problemi çözebilmek için bir araya gelen komutlar, kelimeler, aritmetikler sonucu oluşmuştur (Erdoğan ve Erdoğan, 2017). Programlama bir problem çözme mantığı için algoritma oluşturmak olarak tanımlanırken, kodlama bu mantığın söz dizimine (koda) çevirilmesi olarak tanımlanabilmektedir (Kandemir, 2018a).

Algoritma, ilk olarak 9. yüzyılda İbn Musa El Harezmi tarafından cebir alanındaki matematiksel işlemler ile anılmıştır. Algoritmalar bir problemin çözümünü methodlaştırarak çözmeyi sağlar. Bu method, genellikle kullanılan programlama dilinden bağımsızdır ve birçok programlama diline de eşit seviyede uygun olması muhtemeldir. Algoritmalar, bilgisayar biliminin kavramıdır; bilgisayar programının alandaki çalışma mekanizmalarıdır (Sedgewick ve Wayne, 2011).

Bilgisayar ve matematik bilimi için bütün problemlerin algoritmik bir çözümü olmayabilir. Algoritmik çözümü imkansız problemler vardır; fakat bilgisayarın bilgi işleme birimi tarafından işletilebilecek bir algoritma üretilirse, adımları gerçekleştirmeyi bilen sistem otomasyon programlarını çalıştırabilir (Gülbahar, 2018).

Algoritmik düşünme becerisi ise programlamanın temel kavramlarından biri olarak görülmektedir. Brown (2015)'e göre algoritmaları anlama, uygulama, değerlendirme ve üretme becerisidir. Bu yüzden, algoritmik düşünmenin gelişmesi için programlama dilinin özelliklerine çok fazla odaklanmadan, algoritmaların tasarlanmasına yeterli zaman verilmelidir. Bu sayede öğrenciler, sözde kod kullanarak ve seviyelerine uygun algoritmalar geliştirerek programlama ve algoritmik düşünme becerilerini geliştirebilirler (Futschek, 2006).

Computer Science: The Discipline adlı bildiride belirtilen “Standart Alan Endişeleri” bölümünde, dijital bilgisayarların merkezi rolde olduklarını ve bilgisayarda çalışacak her program için bilişimci ya da programcıların dört disipline ihtiyacı olduğu saptanmıştır. Bunlar; algoritmik düşünme, gösterme (temsil etme), programlama ve tasarım olarak belirtilmiştir (Grover, 2018). Bu standartlara göre bu bireylerin programlama geliştirmeleri, algoritmik düşünme ve gösterme (temsil etme) disiplinlerine sahip olmaları ve bilgisayarların belli algoritma ile çalışmasını sağlayacak programlar yazmaları beklenir. Bu beceri, farklı programlama dillerinin güçlü yanlarının ve sınırlılıklarının çalışma şeklini bilmeyi, program geliştirme araçlarını test etmeyi, hata ayıklamayı ve bilgisayarların işletim birimlerine hakim olmayı içerir (Denning, 1997).

2.1.2 Programlama Öğretiminin Tarihçesi ve Gelişimi. 21. yüzyılın ilk çeyreğinde olmamıza rağmen dijital dünyanın hayatımıza olan devamlı etkisi görülmektedir. Günümüzde 20. yüzyıl becerilerini taşıyan bireylerin, sadece sanayinin etkin olduğu becerilere sahip olması, belirli kurul ve kuruluşlar tarafından yeterli olmayacağı açıklanmıştır. Bunlardan biri, 2002 yılında Amerika Birleşik Devletleri'nde (ABD) iş dünyası, eğitim bilimcileri ve politikacıları bir araya getiren P21 oluşumudur.

P21 gibi, dünyanın birçok yerinde farklı kuruluşlar 21. yüzyıl becerilerini açıklığa kavuşturmak için çeşitli girişimlerde bulunmuşlardır. EnGauge, North Central Regional Technology in Education Consortium (NCREL) tarafından desteklenerek çalışmalarını yürütmektedir. Assessment and Teaching of 21st Century Skills (ATCS) beceri temelli müfredat çalışmaları yapmaktadır. ISTE, teknoloji dönüşümlü eğitim öğretim çağını standartlaştırmaktadır. Avrupa Birliği (AB) ve Organisation for Economic Co-operation and Development (OECD) ise bu becerileri betimlemek için girişimde bulunmuşlardır. Kuruluşların yaptığı araştırmalar sonucunda, 21. yüzyıl

becerilerini içeren ortak noktalar problem çözme, analitik ve eleştirel düşünme, yaratıcılık ve işbirlikli çalışma gibi bilgisayarca düşünmenin temelinde yer alan bulgulardır. Bunun yanında bilgi ve iletişim teknolojileri okuryazarlığı becerisi ise dijital ortamda bilgi edinme ve paylaşma, insanlarla iletişimde bulunma, programlama becerisi edinme ve programlama aracını kullanma olarak tanımlanmıştır (Hamarat, 2019).

Bu yüzyılda, teknoloji günlük hayatımızın her faaliyet alanında kullanılmaktadır. Dijital yerliler olarak adlandırılan yeni nesil öğrencilerin de eğitim öğretim süreçleri bilgisayar ve teknoloji araçlarıyla çerçevelenmiş olarak yürütülmektedir (Gülbahar ve Karal, 2018). Bu bağlamda, programlama öğretiminin K12 öğretim seviyelerinde yaygınlaşmaya başlamıştır. Programlama öğretiminin küçük yaş seviyelerinde başlaması, programlama öğretimini daha zor ve karmaşık hale getirebilmektedir (Kert ve Uğraş, 2009). Programlama öğrenmek ise oldukça zor bir beceridir (Helminen ve Malmi, 2010). Programlama öğrenme, problem çözme, mantıksal düşünme ve soyutlama gibi becerileri zorlamaktadır (Bayman ve Mayer, 1988; Papert, 1980). Daha verimli ders müfredatlarının geliştirilmesi için gün geçtikçe araştırmalar artmaktadır (Bennedsen, 2008). 1980 ve 1990'lı yıllar arasında kişisel bilgisayarların teknolojik olarak gelişmesi ve yaygınlaşması ile programlama öğretiminin alanyazın süreci hızlanmıştır (Raadt, 2008).

2.1.3 Dünya Ülkelerinde Programlama Öğretimi. 21. yüzyılda gelişen beceriler arasında yer alan yaratıcılık, işbirliklilik, problem çözme, eleştirel düşünme ve özellikle “algoritmik düşünme” ise programlama becerisi ile eş değer olarak görülmektedir (Balanskat, Engelhardt, Licht, 2018). Dünya ülkeleri, bu beceri standartlarını belirlerken eğitim müfredatlarına da bu becerileri almaya çalışmaktadır. Ülkeler, programlama öğretimini tek ders ya da başka dersler ile entegre ederek öğretmeye başlamıştır. Birçok ülkede yaygınlaşmaya başlayan programlama dersleri, sadece okullarda değil, okul dışında etkinliklerde ve kulüp çalışmalarında da oldukça yaygınlaşmıştır (Gülbahar ve Karal, 2018).

Programlama öğretimi, genel olarak ortaöğretim düzeyindeki müfredatlarda yaygın olarak görülüyor; fakat yakın zamanda yapılan çalışmalarda ilköğretim ve okul öncesinde de programlama öğretimine ilginin arttığı görülmektedir (Angeli, Voogt, Fluck, Webb, Cox, Malyn-Smith, ve Zagami, 2016). İsveç ve Norveç ülkelerinde yayınlanmış programlama müfredatında 2020 yılında tüm pilot uygulamaların

tamamlanıp, okullarda hazırlanan pilot müfredatların uygulanmaya başlanması ön görülmüştür. Hazırlanan müfredatta okul öncesi ile başlayarak algoritmanın nasıl verileceği planlanmıştır. 1-3 yaş arası bilgisayarsız etkinliklerle algoritmanın temeli olan adım adım yönergeleri oluşturma ve takip etme; 4-6 yaş arasında algoritma oluşturma ve blok temelli programlamada yaratıcılığını kullanma ve 7-9 yaş arasında ise programlamayı öğrenme, matematiksel problem çözme ve algoritmaların nasıl yapılarını bilme ve oluşturma olarak belirlenmiştir (Balanskat vd., 2018).

European Schoolnet araştırma raporunda 21 Avrupa Birliği ülkesinden 16'sında programlama öğretimi, ders ya da kulüp etkinlikleri olarak yer almaktadır (Morvan, 2018). Amerika Birleşik Devletleri'nde ortak bir müfredat bulunmamasına rağmen 31 eyalet, 120'den fazla şehir ve bölgede bilgisayar bilimleri dersi K12 seviyesinde bulunmaktadır. Avustralya'da 2014 yılında başlatılan STEM müfredatı ile 3 yaşından itibaren blok temelli programlama öğretimi zorunlu hale gelmiştir. İngiltere'de 2014 yılı itibariyle programlama öğretimi ilkökul ve ortaokul düzeylerinde zorunlu hale geldikten sonra, 5-6 yaşındaki öğrencilere algoritma mantığı ve nasıl kullanılacağı gösterilmektedir (Gülbahar ve Karal, 2018). Finlandiya'da, 1980'li yıllarda bilgisayar bilimleri dersi lise düzeyinde verilmeye başlanmıştır; fakat 2016 yılında Milli Eğitim Bakanlığı programlama öğretimini müfredatın önemli bir parçası haline getirmiştir (Finnish National Board of Education, 2016).

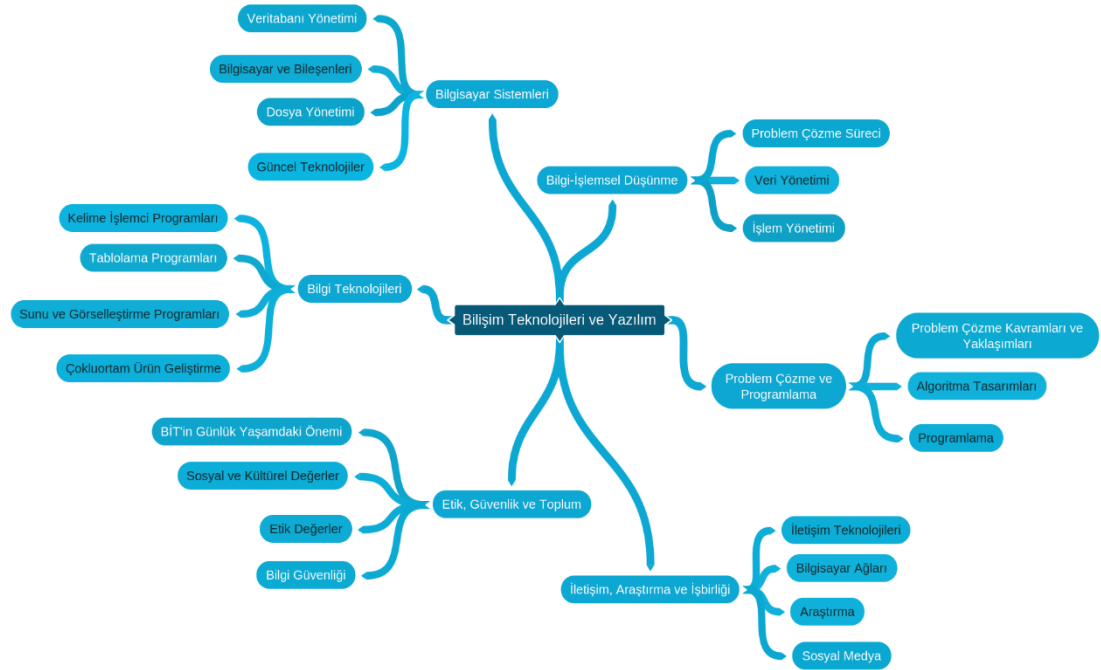
Programlama öğretimi, AB tarafından Kasım 2013'te ülkelerin gelişimine katkısı olması, erken yaşta öğrenilmesi ve çeşitli programlama etkinlikleri yaparak bu beceriyi edinmesi için "Avrupa Kodlama Haftası (Europe Code Week)"nı kabul etmiştir. European Schoolnet tarafından açıklanan 2018 yıllık raporuna göre 2,74 milyon katılımcı ve 43.600'den fazla etkinlik ile birçok ülkeden katılım olduğu sonucuna ulaşılmaktadır (Morvan, 2018). Bunun dışında Bilgisayar Bilimleri Eğitimi Haftası'nda "Kodlama Saati", Code.org ile "Kod Saati", Code Club ve CoderDojo gibi etkinlikler de tüm dünyada programlama öğretiminin yaygınlaştığını göstermektedir.

2.1.4 Ülkemizde Programlama Öğretiminde Güncel Çalışmalar.

Geliştirilen programlama öğretim programları incelendiğinde, her ülkenin kendine ait bir politikası olduğu görülmektedir. Programlama öğretimine bazı ülkeler ders olarak veya başka bir ders ile entegre olarak başlarken bazıları kulüp ve ders dışı aktivitesi olarak başlamıştır.

2017 yılında 20. yılını kutlayan European Schoolnet Milli Eğitim Bakanlık ağı, 1997 yılında 18 üye ülke ile başlayıp, ülkemizde üyesi olduğu 34 üye ülke ile yoluna devam etmektedir. Teknolojinin pedagojik kullanımının evrimini açıklayan raporunda programlama öğretim müfredatları internetin yaygınlaşması ile okullarda yer almaya başlamıştır. 20 ülkenin öğretim programında bilgisayarca düşünme beceri düzeylerinin bütünleştiği iki temel gerekçe yer almaktadır. Bunlardan biri; problem çözme, mantıksal ve algoritmik düşünme gibi 21. yüzyıl becerilerini desteklemek, ikincisi ise farklı sektörlerde bilgisayar bilimi becerilerine sahip bireylerin yetişmesini sağlamak olduğu belirtilmiştir (Balanskat ve Engelhardt, 2015).

Ülkelerdeki bu değişimlerle birlikte Türkiye’de BTY dersindeki güncellemelere 2016’da yılında başlamıştır. Bu programa göre ilkökul, ortaokul ve lise düzeyinde BTY dersi öğretim programına göre yer alan konular Şekil 1’de gösterilmektedir.



Şekil 2. BTY dersi ile bilgisayar bilimi derslerinin kapsamı (Gülbahar ve Kalelioğlu, 2017).

1998 yılında ilkokul 1-5. sınıf düzeylerinde “Seçmeli Bilgisayar Dersi” olarak öğretim programlarında bir süre ile okutulmuştur (Milli Eğitim Tebliğler Dergisi, 1998). Bu dersin kazanım kapsamaları incelendiğinde daha çok bilgisayarı tanıma ve kullanma, basit veritabanları ve programlama temel bilgileri yer almaktadır.

2006 yılında yenilenen programlama öğretim programı ile “Seçmeli Bilgisayar Dersi” hem ilkokul hem ortaokul (1-8. sınıf) düzeylerinde beş öğrenme alanını içine alacak şekildedir. Bu beş alan; Temel İşlemler ve Kavramlar, Bilişim Teknolojileri’nin Kullanımı ve Bilişim Teknolojileri’nde İleri Uygulamalar, Bilişim Teknolojileri’nde Bilimsel Süreç ve Bilişim Teknolojileri Etiği ve Sosyal Değerler olarak belirlenmiştir. Öğretim programındaki Bilişim Teknolojileri’nde İleri Uygulamalar alanı ile programlama öğretiminde programlamaya giriş, nesne tabanlı programlama ve web tasarımı yer almaktadır. İlk beş yıl içerisinde kazanımlar ve üniteler incelendiğinde ise algoritma ve bilgi işlemsel adımların öğretildiği bir öğretim programı olduğu görülmektedir (Gülbahar ve Karal, 2018). Öğretim programında 5. sınıftan itibaren programlama bilgi ve becerilerinin çeşitli ünitelerle desteklenmek istendiği sonucuna ulaşılmaktadır. 5. basamakta yer alan veritabanı konusu ile basit problemlerin çözümü, değişken kavramı ve basit matematiksel işlemler; 6. basamakta algoritma ve akış diyagramları çalışmaları, problem çözümü; 7. basamakta algoritma yazma, programlama dilleri ve web sitesi hazırlama ve 8. basamakta “Programlama Yapıyorum” ünitesi ile nesne tabanlı programlamaya yönelik değişken ve sabitlerin kullanımı, döngüler gibi kavramların öğretilmesi yer almaktadır (MEB, 2006).

BTY dersi ortaokul düzeyinde (5, 6, 7 ve 8. sınıflar), 2012 yılında yayınlanan öğretim programı ile dört yeterlilik üzerine hazırlanarak “standart tabanlı program” anlayışı benimsenmiştir. Bu yeterlilikler; Bilişim Teknolojilerini Kullanarak İletişim Kurma, Bilgi Paylaşma ve Kendini İfade Etme, Araştırma, Bilgiyi Yapılandırma ve İşbirlikli Çalışma ile Problem Çözme, Programlama ve Özgün Ürün Geliştirme olarak belirtilmiştir. Bu yeterlilik düzeylerinden “Problem Çözme, Programlama ve Özgün Ürün Geliştirme” ile programlama öğretimi konusu ön plana çıkmıştır. Düzeylere göre standartlar incelendiğinde problem çözme stratejileri geliştirme, analitik düşünme, akış şeması oluşturma, algoritmik düşünme, değişken ve temel işlemler, programlama dillerini tanıma, derleyici kullanma, program yazma ve çalıştırma, algoritmayı programlama kodlarına dönüştürme ve yazılım projesi geliştirme gibi kazanımlar bulunmaktadır (MEB, 2012).

2016 yılında ortaokul 5. ve 6. sınıf düzeyi için hazırlanan BTY dersi öğretim programı incelendiğinde ise problem çözme ve programlama öğretiminin daha sarmal bir yapıda olduğu görülmektedir (Gülbahar ve Karal, 2018). Bu öğretim programı kapsamında 5. sınıf düzeyinde problem çözme kavramları ve yaklaşımları ele alındığında programlama öncesinde temel oluşturması, problem çözme sürecini yönetmesi ve probleme çözüm üretmesi beklenmektedir. Bununla birlikte değişken, sabit, operatör işlemleri, algoritma, akış şeması oluşturma, test etme ve hata ayıklama gibi konular da yer almaktadır. Programlama basamağında ise blok temelli bir uygulama ile programlama, algoritmik mantık, karar ve tekrar yapılarını kullanarak hatasız yazılımlar oluşturmaları amaçlanmıştır. Programdaki sarmal yapıdan dolayı 6. sınıfta da problem çözme kavramları ve yaklaşımları konusunda benzer kazanımlarla birlikte fonksiyonlar, problemleri basamaklara ayırma ve algoritmayı iyileştirme konuları bulunmaktadır. Programlama kazanımlarında ise çoklu karar yapıları, karmaşık problemler için program oluşturma ve özgün yazılım içeren projeler yer almaktadır (MEB, 2016).

2018 yılında, 7. ve 8. sınıflarda seçmeli ders olarak kabul edilen BTY dersinin öğretim programı güncellenmiştir. Aynı zamanda 1-4. sınıflar düzeyinde de öğretim programı yayınlanmıştır. İlkokul 1-4. sınıflar öğretim programı kapsamında bilgi işlemsel düşünme beceri düzeyleri olan eleştirel ve mantıksal düşünme, problem çözme ve algoritma tasarlama amaçlanmaktadır. “Problem Çözme ve Programlama” ünitesi tüm düzeylerde bulunmaktadır. Problem çözerken farklı mantık yapılarını kullanmaları ve kendi özgün oyunlarını tasarlayarak programlama geliştirmeleri hedeflenmektedir. 7. ve 8. sınıf düzeylerinde ise 6. sınıfta kazanımların devamlılığı ve benzerliği niteliğinde bir yaklaşım olduğu, öğrencilerin metin temelli bir araç ile programlama öğrenmeleri görülmektedir (MEB, 2018).

Ülke genelinde de programlama öğretimi ortaokul düzeyinde 2014 yılından itibaren düzenlenen çeşitli etkinlik, yarışma ve projelerle desteklenmektedir. Bu projelerle öğrencilerin programlama konusunda ilerlemeleri amaçlanmaktadır. Bu kapsamda, ilk projenin 2015 yılında Manisa Valiliği tarafından başlatılan Kodla(MA)nisa projesinin bilişim teknolojilerinde değişim başlattığı görülmektedir. Proje kapsamında 17 ilçe 292 resmi ortaokulda BTY dersinde algoritma, kodlama ve 3 boyutlu tasarım konuları öğretilmektedir. Yerel projeler dışında, Haydi Kızlar Kodlamaya, Kodlama Saati, CoderDojo Türkiye ve Bilge Kunduz-Uluslararası

Enformatik ve Bilgi-İşlemsel Düşünme Etkinliği gibi programlama öğretimine katkıda bulunan tüm dünya ülkeleri ile eşdeğer projeler de yer almaktadır.

2.1.5 Programlama Öğretiminde Cinsiyetin Etkisi. Günümüz yüzyılında, programlama becerisine sadece BT uzmanlarının sahip olmasından öte tüm bireylerin bu beceriye sahip olması beklenmektedir (Alsancak-Sarıkaya, 2018). Bilgisayar biliminin var oluşunda öncü olan kadınların ise sektörde sayılarının azalmasının etkisi merak uyandırmaktadır (Öigaard, 2015). 2015 yılı itibariyle bilgisayar bilimleri okullarından mezunların sadece %18'i kadındır (Vu, 2017). 2016 yılında yayınlanan Women in it Scorecard kaynağında BT alanında cinsiyet dengesizliğinin halen devam etmekte olduğu ve bilişim alanında İsrail'de kadınların en yüksek oranla %26 olduğu açıklanmıştır (The Tech Partnership ve BCS, 2016). Diğer yandan bilgisayar bilimleri alanlarında çoğunlukla erkeklerin tercih edilmesi bireylerin program geliştirme konusunda başarıyı cinsiyetin etkilediği hissettirmektedir. Kadınların programlama alanında daha çok rol model olarak bu ön yargıyı kaldırabilecekleri ise söz konusudur (Byrne ve Lyons, 2001; Zeldin ve Pajares, 2000).

Kadınların programlamaya ilişkin konularda erkeklere nazaran daha az yer edinmesinde ise toplumda yer alan cinslere atfedilen rollerin beceri inançlarını etkileyebileceği söylenmektedir (Aldağ ve Tekdal, 2015). Kız öğrencilere teknoloji ve mühendislik alanlarında atfedilmiş olan sosyal roller, beceri inançlarını etkilese bile programlama başarılarında bağımsız olabilmektedir. Ancak kariyer seçimi noktasında ise ergenlik dönemlerinde bulunan kızlar programlama ve bilgisayar alanlarını öz-yeterlilik algılarının düşük olmasından dolayı seçmemektedir (Aldağ ve Tekdal, 2015).

Kadınların programlama alanındaki genel durumlarına ilişkin alanyazın çalışmasında edinilen bilgiler şu şekildedir; bilgisayar deneyimi, bilgisayarlarla ilgili konu ve işlerle ilgili tutum, bilgisayar oyunlarında cinsiyete göre farklılıklar, rehberlik ve rol modellerde farklılık, sosyal ortam, aile ve öğretmenlerin teşvikleri, öz güven eksikliği olarak sıralanmaktadır (Gürer ve Camp, 2002). Bu dezavantajların programlama öğretiminde kullanılan öğretim stil ve stratejileri ile değişebileceği gözlemlenmiştir (Aldağ ve Tekdal, 2015). Kızların bilgisayar bilimine karşı olan endişesinde programlama öğretiminde kullanılan araçların da etkisi bulunabilmektedir, kız öğrencilerin daha çok görsel tasarım ya da programlamaya yatkın olabilecekleri belirtilmektedir (Morris ve Thrushell, 2014). Aynı zamanda

programlama öğretiminde rol model olarak hemcinslerinin olması öz-yeterlilik algısı ve eğilimlerini bu yönde etkileyebilmektedir (Aldağ ve Tekdal, 2015; Zeldin ve Pajares, 2000). Malezya Üniversitesi'nde yapılan bir çalışmada, rol model olarak kadınların olmasının programlamaya ilgi duymanın yanında çevreden gelen olumlu etkileri de önemli kıldığı kabul edilmiştir (Othman ve Latih, 2006). Programlama dilleri matematikle bağlantılıdır ve kadınlar matematikte başarılıdır. Odaklanması gereken nokta programlama öğretimi yaklaşımlarının cinsiyetlere göre farklılık gösterdiğidir. Özellikle kadınlar yeni öğrendikleri bilgileri öğrenirken gerçeklikle bağlantı kurarlar ve değerlendirirler (Othman ve Latih, 2006; Öigaard, 2015). Bu durum dikkate alınarak programlama öğretim programları hazırlanabilmektedir. Kızların programlama performanslarının erkeklere göre daha iyi niteliklere ulaşabilecekleri gözlemlenmiştir. Ancak programlama becerilerine ve bu alandaki kariyer olanaklarına olan öz-yeterlilik algısının ve motivasyonun yüksek tutulması için gerekli araştırmalara devam edilmesi gerekmektedir (Amaral, Emer, Bim, Setti ve Gonçalves, 2017).

2.2 Blok Temelli Programlama Öğretimi

Günümüzde, programlama becerisi bireylerin sahip olması gereken üst düzey düşünme becerilerini de gerektiren bir beceri olarak görülmektedir (Akıncı ve Tüzün, 2012). 21. yüzyıl öğrencilerinin analitiklik ve eleştirel düşünen, problem çözme becerisi yüksek, iletişim becerileri gelişmiş ve yaşam boyu öğrenmeye açık bireyler olması beklenmektedir. Hızla gelişen teknoloji nedeniyle bu beceriler ile birlikte bilgi okuryazarlığı, medya okuryazarlığı, dijital okuryazarlık, bilgi ve iletişim teknolojileri okuryazarlığı gibi okuryazarlık becerileri de dikkat çekmektedir (Günay ve Şişman, 2019). Sistem tasarlamada, problem çözümünde ve insan davranışını anlamada etkin bireylerin, bilgisayarca düşünme becerilerinin gelişmiş olması gerekmektedir. Temelde tüm becerilere sahip bireylerin yetişmesi istense de özelde bilgisayarca düşünme becerilerine sahip bireylerin yetişmesi amaçlanmaktadır, dolayısıyla erken yaşta programlama öğretimi ile bu amaç sağlanabilir (Yükseltürk ve Altıok, 2018).

Programlama öğrenmenin zorluğu konusunda alternatif çözüm yolları LOGO programlama dilinin geliştirilmesiyle başlamıştır. Programlama dillerinin ve araçlarının geçmiş tarihi incelendiğinde, bu sürecin uzun olduğu ve halen gelişmekte olduğu görülmektedir. Uzman ve öğretmenlerin ortak görüşü ise programlama dili

öğretiminin kolay olmadığı yönündedir (Yükseltürk ve Altıok, 2018). Alanyazın incelendiğinde, programlama öğretimindeki kavramların soyut işlem dönemindeki öğrencilerin soyut düşünme becerilerindeki eksiklikler nedeniyle zorlandığını belirtmiştir (Cevahir ve Özdemir, 2017). Somut işlem döneminde olan öğrencilerin, soyut kavramları öğrenebilmesi için belirlenmiş aralıklarla etkinlikler ve uygulamalar yapılmalıdır (Gagne, 1985). Dolayısıyla programlama öğretiminde kuramsal bilgileri gerçek hayatla özdeşirilen örneklerle hazırlandığında başarının arttığı görülmektedir (Hu, 2004). Öğrenci merkezli ve işbirliğine dayalı yöntem ve tekniklerin kullanılması ile motivasyonun olumlu etkinlendiği, blok temelli programlama gibi görselleştirilmiş araçların öğrenmeyi kolaylaştırdığı ortaya çıkmıştır (Lin ve Zhang, 2003; Malan ve Leitner, 2007).

2.2.2 Blok Temelli Programlama Öğretiminde Araç Seçimi. Programlama öğretimiyle ilgili öne çıkan zorluklar göz önüne alındığında, programlama öğretimine ilk başlayanlar için basit arayüzü olan, metin temelli kod yazmayı gerektirmeyen ve oyun tasarımı odaklı programlama araçlarının kullanılması tavsiye edilmektedir (Kalelioğlu ve Gülbahar, 2014).

Programlama öğretiminde zor olan kuramsal kavramlarını öğretmek için erken yaşta programlamaya başlamalarını sağlamak ve eğlenerek uygulama geliştirecekleri blok temelli programlama araçları kullanılmaya başlanmıştır (Yükseltürk ve Üçgül, 2018).

İnsan beyninin görsel bilgileri, metin kaynaklı bilgilere göre daha iyi anlamlandırıp aralarındaki ilişkiyi daha sağlam kurduğu araştırmalar tarafından desteklenmektedir (Bergin vd., 1996). Bu yüzden programlama öğretiminde de görsel programlama araçlarının kullanılması yaygınlaşmıştır. Görsel programlama araçları farklı başlıklar altında ele alınmaktadır (Cooper vd., 2006):

- **Blok temelli programlama araçları:** Programlamanın grafiksel blokların “sürükle-bırak” yöntemlerle gerçekleştirilmesini sağlarlar (Alice, Scratch, App Inventor vb.).
- **Öyküleştireilmiş algoritma araçları:** Belirli bir karakter aracılığıyla algoritma canlandırılır (Jeroo ve Microsoft Kodu Game Lab vb.).
- **Akış şemalı algoritma araçları:** Programlamanın işlem sırasına ve bağlantısal yapılandırmaya dayalı yöntemlerle çalışırlar (Raptor, FLINT

vb.).

- **Ürünü görselleştirme araçları:** Ürün çıktılarının görselleşmesine odaklanırlar (Lego Mindstorms, RoboMind vb.).
- **Sıralı-aşamalı programlama araçları:** Programlama kavram ve unsunlarını sıralı veya aşamalı olarak düzeye göre genişler (ProfessorJ, RoboLab vb.).

Bu kategoriler içerisinde en çok öne çıkan blok temelli programlama araçlarıdır. Bu araçlar, görsel ve bileşen temelli olarak söz dizimi komutlar yerine, blok yapıların “yap-boz” şeklindeki menüleri ile programlama kavramlarını doğal programlama diline yakın bir biçimde somutlaştırmaktadır (Yükseltürk ve Üçgül, 2018). Söz dizimi yani metin temelli komutların ve kuralların arasında kaybolmadan, kod satırlarını ezberlemeden yeni başlayan kullanıcılara kolaylık sağlamaktadır. Blok temelli programlama araçlarının genel özellikleri: blok kod yapısı, kolay arayüz, hata ayıklama yapısı, çokluortam desteği, tasarım odaklı yapı ve çevrimiçi paylaşım olarak sıralanmaktadır (Resnick vd., 2009).

Blok temelli programlama araçlarının yaş ve seviyeye göre kategorisi aşağıdaki gibidir (Yükseltürk ve Üçgül, 2018).

- **Seviye 1:** 2-7 yaş aralığı, sürükle-bırak veya daha basit, sadece sıralama tekniği öğretir, soyutlama gerektirmez. Fonksiyonlar, değişkenler, döngüler, veri yapıları gibi özellikleri desteklemez. Bee-Bot App, Cato’s Hike, Daisy the Dinosaur, Lightbot Jr., ScratchJr, code.org vb.
- **Seviye 2:** 5-10 yaş aralığı, sürükle-bırak, soyutlama gerektirmez veya sınırlı şekilde uygulanır. Fonksiyonlar, değişkenler, döngüler, veri yapıları gibi özellikleri desteklemez ya da az miktarda destekler. ScratchJr, code.org, Blookly, Lightbot, Kodable, Cargo-Bot, Foos, Tynker vb.
- **Seviye 3:** 8-14 yaş aralığı, sürükle-bırak veya metin temelli, soyutlama vardır. Fonksiyonlar, değişkenler, döngüler, veri yapıları gibi özellikleri neredeyse tamamını destekler. Scratch, HopScotch, Tynker, Kodu, Codemancer vb.
- **Seviye 4:** 12 ve üzeri yaş aralığı, sürükle-bırak veya metin temelli,

soyutlama vardır. Fonksiyonlar, deęişkenler, döngüler, veri yapıları gibi özellikleri neredeyse tamamını destekler. Alice, AppInventor, Looking Glass vb.

Programlama öğretiminde kullanılacak programlama aracının öğretim hedeflerine, öğrencinin ilgisini çekecek ve bilgi seviyesine uygun olarak seçerken çok dikkat edilmesi gerekmektedir.

Son yıllarda ülkemizde her düzeyde yapılmış olan alanyazın çalışmaları incelendiğinde daha çok ortaokul düzeyinde örneklem gruplarının deney ve kontrol grubundan oluştuęu, blok temelli programlama öğretili sonrasında öğrencilerin akademik başarıları, programlamaya ilişkin kaygı, özgüven, özdenetim ve öz-yeterlilik algıları, bilgi-işlemsel düşünme, problem çözme ve algoritma geliştirme becerileri araştırılmıştır. Bu çalışmalara ilişkin özetler aşağıdaki gibidir:

Kalelioęlu ve Gülbahar (2014) ilköğretim öğrencileri ile yaptığı çalışmada Scratch programı ile programlama öğretimini öğrencilerin problem çözme becerilerine olan etkisini incelemiştir. Çalışmada 5 haftalık süreçte basit programlar ve oyunlar tasarlamışlardır. Araştırma sonucunda problem çözme puanları arasında, özgüven, özdenetim ve kaçınma alt boyutları da dahil olmak üzere farklılık olmadığı; fakat öğrencilerin Scratch ile programlamayı sevdiği, programlamaya devam etme ve oyun geliştirme isteklerinin olduğu belirtilmiştir.

Demir (2015) yapmış olduğu çalışmada, programlama öğretimini önündeki zorlukların aşılmasına yönelik olarak, programlama öğretiminde eğitsel programlama dilinin derse entegrasyonunun, öğrencilerin akademik başarısına, performansına ve bilgisayar programlamaya yönelik kaygılarına olan etkisi araştırılmıştır. Deneysel yöntemlerden kontrol grupsuz ön-test son-test deney deseninin kullanıldığı çalışma, üç deney grubu ile yürütülmüştür. Birinci grupta blok temelli programlama dili, dersin uygulama kısmına, ikinci grupta teori kısmına, üçüncü grupta ise hem teori hem de uygulama kısmına entegre edilerek gruplar oluşturulmuştur. Rastgele örneklem yöntemi ile oluşturulan gruplar 61 erkek ve 26 kız olmak üzere 87 öğrenciden oluşmaktadır. Sonuç olarak, blok temelli programlama dilinin dersin hem teori hem de uygulama kısmına entegre edilmesinin bilişsel ve üst düzey bilişsel beceriler açısından öğrencilerin programlamaya yönelik akademik başarılarını, performanslarını ve bilgisayar programlamaya yönelik kaygılarını olumlu yönde etkilediğini

göstermektedir. Akademik başarı ve ders içi performansı artırmak ve bilgisayar programlamaya yönelik kaygıyı azaltmak için, eğitsel programlama dilleri, dersin hem teori hem de uygulamasına entegre edilerek kullanılabilir olduğu belirlenmiştir.

Doğan (2015) çalışmasında bilgisayar oyunu geliştirme sürecinin eleştirel düşünme becerilerine ve algoritma başarılarına etkisini ortaya çıkarmayı amaçlamıştır. Araştırma ön test - son test kontrol gruplu araştırma modelindedir. Araştırmanın çalışma grubu 6. sınıfta öğrenim gören 54 öğrenci oluşturmaktadır. BTY dersindeki algoritma konusu deney grubu ile oyun geliştirme sürecinde Kodu programı ile işlenirken kontrol grubu ile geleneksel yöntemle işlenmiştir. Bilgisayar oyunu geliştirme süreci ile algoritma ve programlama derslerini yapmak geleneksel yöntem ile yapılan derslere göre eleştirel düşünme becerilerini ve algoritma başarılarını anlamlı düzeyde olumlu etkilediği görülmüştür.

Erdem ve Kalelioğlu (2018) çalışmasının amacı 5. sınıf öğrencilerinin yüz yüze eğitim ile ters yüz sınıf modeli olan teknoloji destekli öğrenme ortamlarında olmak üzere iki farklı öğrenme ve öğretme stratejisi ile Scratch programlamayı öğrenmelerinin ve programlama öğretiminin, öğrencilerin bilgi işlemsel düşünme becerilerine etkisini araştırmaktır. Çalışmaya özel bir okuldan 5. Sınıfta öğrenim gören 79 öğrencisi katılmıştır. Hem nicel hem nitel yöntemleri içeren karma araştırma yöntemlerinden faydalanılmıştır. Araştırma sonuçlarına göre yüz yüze eğitim ve ters yüz sınıf modeli ile eğitimin öğrencilerin Scratch programlama öğrenmelerinde ve bilgi işlemsel düşünme becerilerinde istatistiksel olarak anlamlı olmayan bir fark oluşturduğu görülmüştür. Bunun yanında, algoritma geliştirme ve problem çözme yeterlilik puanları ön-test ve son-test puanları arasında farkın anlamlı çıkması, programlama öğretiminin bu yeterliliklerin artmasında önemli bir etken olduğu ortaya çıkmıştır.

Oluk, Korkmaz ve H.A. Oluk (2018) çalışmasında Scratch kullanımının algoritma geliştirme ve bilgi-işlemsel düşünme becerilerini geliştirmede etkisini incelemiştir. Araştırma ön test son test kontrol gruplu yarı deneysel bir çalışma olup 31 deney 31 kontrol grubu olmak üzere 62 tane 5. sınıf öğrencisi ile yürütülmüştür. Çalışma sonucunda deney grubu öğrencilerinin algoritma geliştirme ve bilgi-işlemsel düşünme becerilerinin kontrol grubuna göre anlamlı derecede daha fazla yükseldiği bulunmuştur.

Alp (2019) çalışmasında blok tabanlı programlama öğretiminin, ortaokul 6. sınıf öğrencilerinin problem çözme becerileri ve bilgisayara yönelik tutumları üzerindeki etkisinin belirlenmesi amacıyla, ön test-son test kontrol gruplu yarı deneysel desen kullanılarak gerçekleştirmiştir. Bu amaçla, programlama öğretim aracı olarak Scratch yazılımı kullanılmıştır. Bulgulara göre her iki gruptaki öğrencilerin bilgisayara yönelik tutumları ile problem çözme becerileri, uygulama öncesine göre anlamlı derecede, pozitif olarak değişim göstermiştir.

2.3 Metin Temelli Programlama Öğretimi

Programlama, dijital yerliler olarak adlandırdığımız 21. yüzyıl çocuklarına ve gençlerine tasarım yapan, yenilikçi, girişimci ve üreten bireyler olmaları konusunda onları teşvik edebilecek önemli bir araçtır. Ancak programlama için gerekli öğretim araçları ile planlanırsa daha sağlıklı program geliştirmeleri sağlanabilmektedir (Kandemir, 2018a).

Bilişim sektöründe bilgisayar bilimcileri ve uzmanlarının yaygın olarak kullandıkları diller metin temelli programlama dilleridir. Metin temelli programlama, kaynak kodunun sadece metin komutları ve bunlara ilişkin söz dizimi kuralları ile oluşturduğu programlama dilleridir (Mıhçı, 2014). Programlama geliştirme ortamları (IDE) iki farklı şekilde incelenir: metin temelli kullanıcı arayüz (text-based user interface-TUI) ve grafiksel kullanıcı arayüz (graphical user interface-GUI). Grafiksel kullanıcı arayüzü bazı prosedürlerde görsel programlama dili olarak da adlandırılmaktadır. Ancak bunun blok temelli programlamadaki blok temelli görsel programlama dilleri ile karıştırılmamasına dikkat edinilmelidir.

Blok temelli programlama dilleri, programlama kavramlarının ve problem çözme becerilerinin kazandırılmasında etkin bir şekilde kullanılmaktadır. Geleneksel programlama dili olan metin temelli programlama dilleri ise günlük hayat problemlerinin çözümünde hala aktif bir alandır (Kandemir, 2018a).

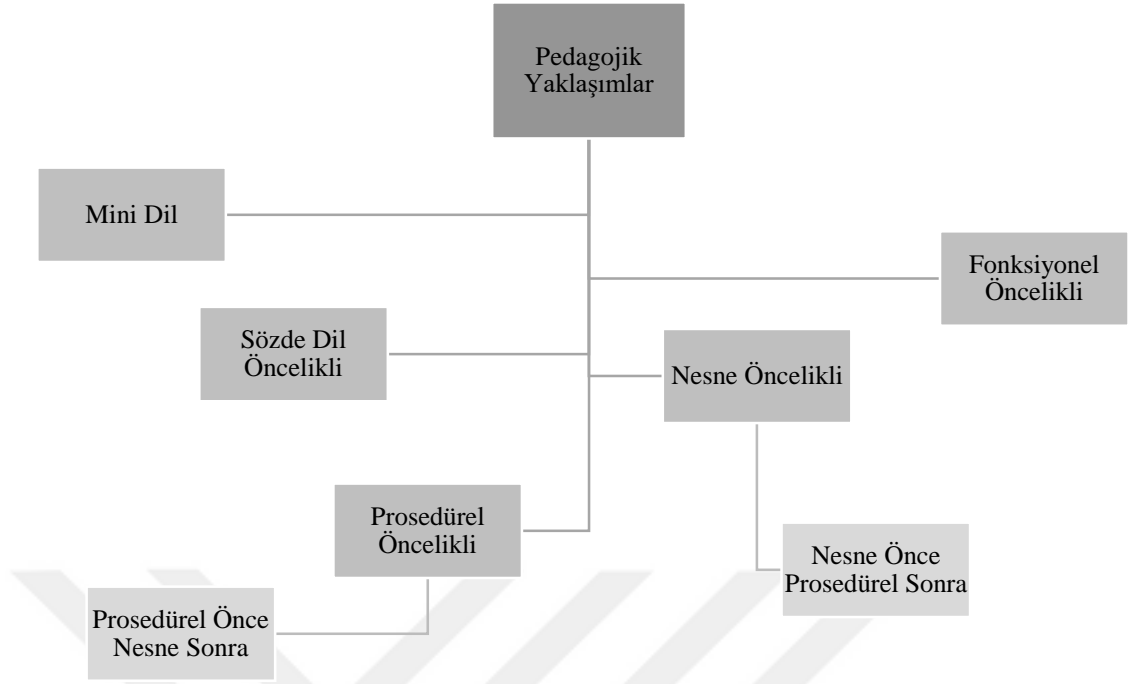
Programlama dillerinin paradigmaları farklı problem çözme yaklaşımlarına göre bilgisayar bilimlerinde 4 temel paradigma olarak incelenmektedir. Bu paradigmaların her biri ayrı bir düşünme biçimini oluşturur ve birtakım programlama dili tarafından desteklenir. Bu paradigmalar aşağıdaki gibidir (Vujosevic-Janicic ve Tosic, 2008).

- **Prosedürel (Zorunlu) Paradigma:** Bir işlemi gerçekleştirmek için bilgisayara verilen komut dizininden oluşur. Daha sonra bu komut dizisi fonksiyon ile çağırılır. Fortran, Cobol, Basic, C vb.
- **Nesne Yönelimli Paradigma:** Olayların benzetimine dayandırılarak geliştirilen programlama dilleridir. Gerçek olaylara benzeştirilirse, bir olayda nesne olmalıdır. Nesnelere arası mesajlaşmanın ve hiyerarşinin programlanmasına nesne tabanlı programlama dilleri adı da verilmektedir. C++, Java ve Smalltalk vb.
- **Fonksiyonel Paradigma:** Matematiksel fonksiyonların teorileri ile oluşmuşlardır. Bir problemin, alt problemlerle soyutlanarak çözümü için gerekli bağımsız fonksiyonlar ile geliştirilmektedir. LISP, ML, Scheme, Miranda ve Haskell vb.
- **Mantıksal Paradigma:** Matematikteki birinci dereceli mantıksal hesaplar ile oluşurlar. Problemin algoritmik çözümüne ulaşmak için mantıksal ilişkilerin belirlenmesi gerekmektedir. Prolog, Gödel ve Curry vb.

Bir programlama dili bir paradigmayı desteklediği gibi birden fazla paradigmayı da destekleyebilir. Örneğin, Python, PHP, C# ve JavaScript bu programlama dillerindedir. Python ve JavaScript programlama dilleri, hem nesne yönelimli hem fonksiyonel bir paradigma yapısına sahip, birinci sınıf fonksiyonlar ile harmanlanmış Script (betik) bir programlama dilleridir.

2.3.1 Programlama Öğretiminde Pedagojik Yaklaşımlar. Programlama dillerinin geliştiricileri alan uzmanı olarak gerçek problemi çözmeyi amaçlayan kendisine ait paradigmasına uyan programlama dilleri geliştirmektedir. O programlama dilini yeni öğrenecek olanların karşılaçıkları sorunlar göz ardı edilir (Kandemir, 2018a).

Bu yüzden, temel programlama becerilerinin kazandırılması için kullanılacak ilk programlama dilinin seçimine göre pedagojik yaklaşımlar geliştirilmiştir (Farooq, Khan, Ahmed, İslam ve Abid, 2014). Beş temel yaklaşımdan üç tanesi programlama paradigmasına dayanmaktadır. Bu yaklaşımlar ve aralarındaki ilişki Şekil 2’de verilmiştir (Farooq vd.,2014).



Şekil 3. İlk programlama dili seçimine yönelik pedagojik yaklaşımlar (Farooq vd.,2014).

Mini Dil Yaklaşımı: Programlamaya yeni başlayanların kolaylıkla öğrenebilmeleri için basit dil tasarlamaktır. Mikro bir dünyada hareket eden bir robot ya da aktif bir aktör vardır, aktörlerin bu dünyadaki hareketleri için ön tanımla programlama dilleri mevcuttur.

Nesne Öncelikli Yaklaşım: Bu yaklaşım programlamadaki nesne ve sınıflarla ilgilidir. Başlıca odaklandığı diğer konular modülleşme, kapsülleme, tekrar kullanılabilirlik, özyleneleme, oluşturma ve işlemedir.

Prosedürel Öncelikli Yaklaşım: Geleneksel programlama öğretim yaklaşımıdır. Değişkenlere, komutların hiyerarşi yapısına, döngü ve karar yapılarının nasıl kurulacağına, fonksiyonların programlamadaki görevlerine odaklanmaktadır.

Nesne Önce Prosedürel Sonra Yaklaşımı: Kaynak kodu yazmadan sanal dünyalar kurmayı sağlayan, nesne yönelimli ve üç boyutlu bir grafik programlama ortamı olan Alice uygulaması bu yaklaşıma uygundur. Yapılan çalışmalarda, motivasyonu artırdığı, döngü, koşul ve olay kavramlarının anlaşılmasında faydalı

olduđu, ancak deęişken konusunda zorluklar yařandığı belirtilmiştir.

Prosedürel Önce Nesne Sonra Yaklaşımı: Bilgisayar bilimleri öğretiminde ilk kullanılan yaklaşımdır. İlk olarak temel programlama kavramları deęişken, döngü, fonksiyon ve karar gibi yapılar verilir, daha sonra nesne yönelimli programlama konuları anlatılmaktadır.

Fonksiyonel Yaklaşım: Matematiksel fonksiyonlar ile programlama mantığının ifade edildiđi yaklaşım türüdür. Bu tür dillerin, basit problemlerin çözümünü ilk günden itibaren kolay hesaplandığı savunulmuştur. Dolayısıyla problem çözme becerilerini artıran temel konularla ilgilidir. Veri yapıları, harita, kümeler, sıra, özyineleme ve fonksiyonlar gibi birçok önemli kavram, resmi olarak bu alanda ortaya çıkmaktadır.

Sözde Dil Öncelikli Yaklaşım: Temel programlama kavramlarının öğretimi için ekstra özelliklere sahip ana programlama dilinin komutları alt kümeleri olarak belirtilmektedir. Bu şekilde en basit şekilde söz dizimi komutları oluşturulur. Dolayısıyla programlama dilinin söz dizimi yerine programlama kavramlarına ve problem çözme becerilerine daha iyi odaklanabilmektedir (Farooq vd.,2014).

Programlama öğretiminde bilgisayarca düşünme becerilerinin kazandırılmasına yönelik çalışmalarda uygun programlama dili aracını ve yaklaşımını seçmek, ders içeriklerini hazırlamak ve bu içerikleri sunmak, hazırlanan etkinliklerde uygun yöntemleri seçmek hala araştırılan konular arasındadır (Kandemir, 2018b).

Programlama öğretimindeki kavramlar soyut terimlerden oluşmaktadır. Piaget'nin bilişsel kuramına göre somut işlemler dönemi 7-11 yaş aralığındadır. Soyut işlemler dönemi ise 12 yaş ve üzeridir, mantıksal ve akıl yürütmenin ortaya çıktığı dönemdir. Gerek bilgisayar biliminde, gerek bilgisayarca düşünmede ilgili alanyazında soyutlama ve problem çözme becerisinin temelidir (Wing, 2006). Bununla birlikte genel düşünce, bilgisayarca düşünme becerilerinin gelişimini desteklemek amacıyla programlama öğretimine beş veya altı yaşlarında başlanabileceğidir (Kandemir, 2018b). Erken yaşta programlama öğretimi oyun veya oyun tabanlı bir ortamdaki etkinliklerle yapılmaktadır. Programlamadaki temel kavramlar oyun etkinlikleri ile bilgisayarsız ortamlarda verilmektedir.

Programlama öğretiminde karşılaşılan zorlukların, programlama dilinin hem dil becerisi hem matematik bilgisi hemde programlamanın bir arada karmaşıklık

yaratması olarak düşünülmektedir. Özellikle erken yaşta programlama öğretiminde, bilişsel döneme saygı duyularak programlamanın bilgisayarca düşünme becerilerinin bir biçimi olduğu ve programlamanın temelinde yer aldığıdır (Hu, 2011). Programlama öğretimine uygun içerik ve programlama dilinin seçiminde Piaget'nin bilişsel gelişim dönemi ile örtüşmesine dikkat edilmesi gerekmektedir (Kandemir, 2018b).

Bilişsel gelişim dönemlerine karşılık olabilecek programlama öğretimleri örnekleri verilmiştir. Buna göre 6 yaş öncesinde programlama öğretiminde metin temelli programlama dili ya da ortamı kullanılmayacağı görülmektedir. Dolayısıyla okuma-yazma bilmeyen işlem öncesi dönemindeki çocuklar için sembol ve yap-boz şeklindeki blok temelli programlama ortamlarının kullanılması daha uygundur. Programlama becerisi kazandırılırken bilgisayarca düşünmenin en önemli alt basamağı olan problem çözmenin temelleri unutulmamalıdır (Kandemir, 2018b). Basit hayat problemlerine yönelik etkinlikler motivasyon aracıdır; problemi çözmek ise programlama kavramlarının yardımı ile mümkündür. Şekil 3'de Piaget'nin bilişsel dönemlerine göre programlama etkinlik önerileri incelenmiştir.

Bilişsel Dönem	Yaş Aralığı	Temel Özellikler	Programlama Öğretimi Etkinlik Önerileri
Duyuşsal Motor	0 - 2	Duyuları ve hareketleri ile dış dünyayı deneyimler. Nesnelere el ile kontrol ederek çevre ile etkileşime girer.	
İşlem Öncesi	2 - 6/7	Algılama düşünme üzerinde egemendir. Fakat nesnelere kelime ve görüntüler ile ifade edebilir. Sembolik işlevleri yerine getirmede daha fazla yetenekli hale gelir. Mantıksal akıl yürütmekten ziyade sezgisel düşünme becerisine sahiptir.	Mikro dünyalar üzerindeki aracı (agent) nesnelere (kukla, kaplumbağa, robot vb.) doğrudan kontrol edilmesi.
Somut İşlemler	7 - 11/12	Mantıksal akıl yürütme sadece gerçek veya görülebilir nesnelere uygulanabilir.	Mikro dünyalar üzerindeki aracı (agent) nesnelere (kukla, kaplumbağa, robot vb.) dolaylı kontrol edilmesi ile birlikte bazı temel programlama kavramları (sıralama, döngü, koşul) kullanabileceği statik ve dinamik senaryolar.
Soyut İşlemler	12 ve üzeri	Birey, potansiyel olaylar veya soyut fikirler hakkında mantıklı düşünebilir.	Fonksiyonlar, parametreler gibi gelişmiş temel programlama kavramlarının yanında etkileşimli mikro dünyalar yardımıyla temel nesneye dayalı programlama kavramlarını öğrenip kullanabileceği kendi senaryolarını üretme.

Şekil 4. Piaget bilişsel gelişim dönemlerine göre programlama öğretimi etkinlik önerileri (Kandemir, 2018b).

2.4 Metin Temelli Programlama Dillerinin Seçimi

Geleneksel öğretim yöntemleri ile programlama öğretimi yapıldığında karşılaşılan zorluklardan biri öğrenenlerin bilişsel gelişim dönemlerine ve yaşlarına en uygun, programlama becerilerini geliştirecek metin temelli programlama aracının seçilmesidir. Bu durumda çocukların soyutlama kapasitelerini doğal yollarla geliştirmeleri önem arz etmektedir. Bu şekilde çocukların bilgisayarca düşünme becerileri alt basamaklarından yaratıcılık ve problem çözme becerileri desteklenmiş olacaktır (Kandemir, 2018b).

İlkokul ve ortaokul düzeylerine göre ihtiyaca yönelik programlama aracı ve dilinin gereksinimleri aşağıda sıralanmıştır (Ristić, Milošević ve Urošević, 2016):

- Kullanılacak programlama dili ve aracı, temel programlama kavramlarını içermelidir.
- Programlama dili güncel programlama kavramlarına sahip olmalı ve programlama becerileri geliştirmelidir.
- Eğitimde kullanılan programlama dilinin yapısı, modern programlamanın kavramlarını içermelidir. Bunun nedeni, bir programlama dilinin temel kavramlarını öğrenmek, diğer programlama dillerinin daha hızlı öğrenilmesini sağlar.
- Programlama dilinin söz dizimi ve derleyicisi kolayca okunabilmeli ve anlaması basit olmalıdır.

Yukarıdaki gereksinimler dışında, hedef kitlenin yaşına bağlı olarak programlama dilinin problem temelli öğrenmeye odaklanılmasının gerektiği savunulmaktadır. Öğrencinin söz diziminin kendisinden daha fazla tekniklere odaklanmaları gerekmektedir (Gupta, 2004). İlk programlama dili seçilirken aşağıdaki bileşenlere önem arz etmektedir. Bu bileşenlere aşağıdaki gibidir (Gupta, 2004):

1. Dil, açık ve sezgisel söz dizimine sahip olmalıdır.
2. Ortak söz dizimsel ve anlamsal yapıları içermelidir.
3. Hataları inceleme sonuçlarının ele alınmasında tutarlı olmalı ve anlamlı hata mesajları vermelidir.
4. Aşırı basit veya fazla ayrıntıya sahip olmamalıdır.

5. Özelleştirilebilmeli ve zamanla değişen ihtiyaçlara göre esnek yapıda olmalıdır.

Bu bileşenlere göre hedef kitlenin yaşı, bilişsel dönemi ve hazırbulunuşluk seviyesine göre öğretilecek programlama dili değişebilmektedir. Kandemir (2018b) göre somut işlemler döneminin ilk yarısında, blok temelli programlama dili ile basit yapıdaki komutlar kullanılarak programlamanın temel kavramlarının görev odaklı verilmesi yeterli olacaktır. Bu sayede öğrencilerin programlamaya karşı motivasyonları artacaktır. Burada blok temelli programlama araçlarından kukla, kamplumbağa veya robot gibi bir aracın komutlar aracılığıyla hareketinin sağlandığı ortamlar seçilebilir. Somut işlemler döneminin ikinci yarısında ise mikro dünya üzerinde hedef odaklı görevleri tamamlama, sınırlı sayıda komut içeren ve problem çözmesi için geliştirilmiş programlama dillerinin seçimiyle komut kümesi ve kavramlar hakkında bilgi verilebilmektedir. CodeMonkey ve CodeCombat gibi çevrimiçi eğitsel hedef odaklı metin temelli programlama ortamında ön tanımlı komutları somut bir şekilde gerçekleştiren programlama dili araçları seçilebilir.

Soyut işlemler döneminde nesneye dayalı programlamanın temel kavramları öğretilir. Başlangıç seviye dili ve .NET dili olan Small Basic programlama dili erken yaşta programlama becerisi ve bilgisayarca düşünme becerilerinin edinilmesinde kullanılabilir. Dinamik senaryo ile hazırlanmış etkinlikler metin temelli programlama dilini ve aracını aktif olarak kullanmasına imkan sağlayabilir. Bu dönemde Python ve JavaScript gibi birçok nesne tabanlı programlamanın da kullanılabildiği diller seçilebilir (Baştuğ, 2019; Kandemir, 2018b).

CSTA (2009) komitesi tarafından hazırlanan raporda, bilgisayar bilimleri ders programı kapsamında programlama kavramı ve bilişsel gelişim dönemlerine göre kazanımlar ve etkinlikler verilmiştir. Bu model müfredatın genel çerçevesi şu şekildedir:

- Bilgisayar donanımı, yazılım uygulamaları ve ağ bağlantılı sistemler hakkında bilgi sahibi olmak,
- Her türlü problemi çözmek için bilgi, algoritma ve programlama dilini yaratıcı bir araç olarak kullanan bir düşünce yolu oluşturmak,
- Bilgisayar, bireyler, kuruluşlar ve kültür arasındaki karmaşık ve değişen etkileşimler arasındaki dengeyi kurmak olarak belirlenmiştir.

K8 müfredatında üç kategori ve bu kategorilerin altında yer alan 12 konu belirlenmiştir. Bu kategoriler: Bilgisayarlar ve Yazılım Uygulamaları, Bilgisayar Bilimi ile Problem Çözme ve Bilgi İşlemenin Sosyal Bağlamı olarak adlandırılmıştır. Problem çözme, algoritma oluşturma ve programlama gibi kazanımlar K8'in tüm düzeylerinde "Bilgisayar Bilimi ile Problem Çözme" temasının altında yer almaktadır. Bilgisayarlar ve Yazılım Uygulamaları temasının altında yer alan "Dünya Çapında Ağlar ve Ağlar ile İletişim Kurma" konusundaki kazanımlarda K3-5 ve K6-8 düzeylerinde standartlara uygun web sayfası şablonu programlama ve tarayıcı ile ilişkileri açıklama kazanımı yer almaktadır (CSTA, 2009). CSTA ve ISTE ile işbirliği içinde çalışan SLO (2019b) kuruluşunun hazırladığı müfredata yönelik çalışmalarda çeşitli destek kaynaklarında web sitesi tasarlama ve oluşturma, JavaScript ve HTML dilleri bulunmaktadır.

2.5 Blok Temelli Programlamadan Metin Temelli Programlamaya Geçiş

Günümüzde, blok temelli programlama araçları erken yaşta programlama öğretimi için başlangıç seviyesinde sıklıkla kullanılmaktadır. Programlama becerilerini kolaylıkla vermek ve motivasyonu yüksek tutmak amacıyla blok temelli programlama araçları tercih edilmektedir (Kandemir, 2018a, 2018b). Buna rağmen, blok temelli programlama ile programlama öğrenen bireylerin metin temelli programlamaya geçtiklerinde zorluklar ile karşılaştıkları görülmüştür (Powers, Ecott ve Hirsfield, 2007).

Powers, Ecott ve Hirsfield (2007) çalışmasında programlama kavramlarının Alice blok temelli programlama aracı ile verildikten sonra nesne yönelimli programlama dilleri olan Java ve C++ ile metin temelli programlama öğretimine geçilmiştir. Öğrenenlerin programlama öğretiminin ilk dersinde programlama kavramlarına karşı özgüvenlerinin yüksek olduğu tespit edilmiştir. Ancak öğrenenlerin metin temelli programlama dillerinde söz dizimine daha az dikkat ettikleri ve derleme sırasında çıkan hatalardan dolayı cesaretlerinin kırıldığı gözlemlenmiştir.

Öğrenenler tarafından belirtilen "gerçek programlama (metin temelli programlama)" olmaması, komutların sınırlı ifade gücünün olması ve gelişmiş programların yapılmaması gibi zorlukların blok temelli programlama araçlarının dezavantajları olduğu tespit edilmiştir (Weintrop ve Wilensky, 2015).

Meerbaum ve arkadaşları (2011) problem çözme açısından bakıldığında blok

temelli programlama aracı ile programlama öğrenen bireylerin program geliştirme alışkanlıkları konusunda değişik davranışlar sergilediği belirtilmiştir. Özellikle öğrenenler programlama görevlerinde algoritmik seviyede ve yazılım tasarımı seviyesinde düşünmeden, problemi çözmede işe yarayacak blok komutları birleştirip işine yarayacakları kullandıktan sonra artan ve programı çalıştırmayan komutları ekranda bıraktığı gözlemlenmiştir. Halbuki programlama yaklaşımında önce bileşen tasarımı ile başlanarak ve daha büyük alt sistem sistem oluşana kadar bileşenler birbiri ile ilişkilendirilir. Bu yaklaşım doğru kullanıldığında programlamanın daha kolay yapılmasını sağlamaktadır. Ancak Scratch ortamı, küçük bileşenler tasarlama konusunda çoğu zaman mantıksal olmayan komut blokları tasarlamaya neden olmaktadır. Üç adımda tasarlanabilen problemi tek komut bloğuyla tasarlamak yerine her adım için ayrı bir blok komutu kullanılarak çözülebilmektedir. Öğrenenlerin ilk programlama deneyimlerinde kolayı mı yoksa doğruyu mu öğretilmesinin hala bir tartışma sorusu olduğunu dile getirmişlerdir. Programlamanın temel kavramları olan sıralama, koşul, döngü ve değişken gibi kavramların blok temelli araçlarda kullanımından kaçınılması ise geleneksel programlama dillerine geçiş için zararlı olabilmektedir (Meerbaum, Armoni ve Ben-Ari, 2011).

Dolayısıyla blok temelli programlama dillerinden metin temelli programlama dillerine geçişte karşılaşılabilecek problemler aşağıdaki gibidir (Kandemir, 2018a):

- Metin temelli programlama dillerindeki söz dizim kurallarına dikkat edilmemesi,
- Metin temelli programlama dillerinde derleme sonucunda hataların oluşması sonucunda tespit etme ve düzeltmede zorluklar yaşanması,
- Blok temelli programlama dili ile metin temelli programlama dili arasında bağlantı kurulamaması,
- Blok temelli programlama ortamlarının gelişmiş komut ve program yazmada kısıtlı kalması,
- BT uzmanlarının kullandığı davranış ve stratejileri blok temelli programlama dilinde uygulayamaması,
- Blok temelli programlama dilinin “gerçek programlama” dili olarak görülememesi.

Sonuç olarak blok temelli programlama dillerinden metin temelli programlama diline geçişte yukarıdaki zorlukların yaşanmaması için metin ve blok temelli programlama dillerinin özdeştirildiği veya bir arada kullanılabileceği araçlar ile etkinlikler hazırlanabilir.

Metin temelli programlamaya geçişte öğretmen ve öğrencilere yönlendirici olması için araştırmalar yapılmaktadır. Blok ve metin temelli programlama dillerinin birlikte kullanılacağı ortamların oluşturulması ya da blok temelli programlama dilini sadece programlama kavramlarını öğretmek için değil metin temelli programlama diline destekleyici şekilde kullanılması gerektiğine dair model geliştirilmiştir (Dorling ve White, 2015). Bu modele göre temel programlama kavramları olan sıralama, koşul, döngü, değişken ve fonksiyon gibi yapılar blok temelli araç ile tamamlandıktan sonra benzer algoritmik yapının eşdeğeri metin temelli programlama aracında geliştirilmelidir. Çalışmanın sonucundaki bulgular ise aşağıdaki gibidir:

- İlkokuldayken metin temelli programlama dili ile programlama öğretimi yapılabilir.
- Tüm ortaokul öğrenenleri metin temelli programlamanın temel kavramları konusunda uzmanlaşabilir.
- Scratch gibi blok temelli programlama dili araçları hem ilkokulda hem ortaokulda yeni başlayan seviyesi için iyi bir seçim olacaktır.

Son yıllarda ülkemizde her düzeyde yapılmış olan blok ve metin temelli programlama dili öğretiminde yapılan alanyazın çalışmaları incelendiğinde ortaokul ve ortaöğretim düzeyinde örneklem gruplarının deney ve kontrol grubundan oluştuğu, blok ve metin temelli programlama öğretimi sonrasında öğrencilerin akademik başarıları, motivasyonları programlamaya karşı tutumları, problem çözme ve mantıksal düşünme becerileri araştırılmıştır. Bu çalışmalara ilişkin özetler aşağıdaki gibidir:

Yiğit (2016), blok temelli görsel programlama ortamı ile öğretimin öğrencilerin programlamayı öğrenmelerine ve programlamaya karşı tutumlarına etkisini incelemiştir. Ondokuz Mayıs Üniversitesi Bilgisayar ve Öğretim Teknolojileri Eğitimi Bölümü'nde öğretim gören 42 ikinci sınıf öğrencisi üzerinde yürütülmüştür. Blok temelli görsel programlama öğretim aracı olan Blockly yazılımı ve metin temelli Python programlama dili üzerinden öğrenim görmüşlerdir. Çalışma sonucunda görsel

programlama öğretimi alan öğrencilerin geleneksel programlama öğretimine göre öğretim alan öğrencilerden programlamayı öğrenmede daha başarılı oldukları fakat programlamaya karşı tutumlarının anlamlı olmadığı tespit edilmiştir.

Bağcı (2017) araştırmasında programlama arayüzleri ve bunların programlama eğitiminin başarısı üzerindeki etkileri üzerine çalışmıştır. Bu nedenle, öğrencilere programlama öğretmek için AESL ve VPL arayüzleri kullanılmıştır. Daha önceden programlama eğitimi almamış öğrenci grupları üçe ayrılmıştır; görsel programlama (Grup 1), metin programlama (Grup 2) ve görsel programlamadan sonra metin programlama çalışan (Grup 3). Bulgulara göre görsel programlama ile kodlama yapan öğrenciler genel olarak daha yüksek başarı oranına sahip oldukları hesaplanmıştır. Buna ek olarak, metin programlamayı öğrenirken, öğrencilerin daha önce görsel programlama bilgisine sahip olmaları durumunda, metin programlama kavramlarını öğrenmede daha başarılı sonuçlar elde edilmiştir. Görsel programlama dilleri ilk kez programlama öğrenmek için kullanılıyorsa metin programlama dillerine daha kolay uyum sağladıkları sonucuna ulaşılmıştır.

Saygıner (2017) çalışmasının amacı blok tabanlı görsel ve metin tabanlı programlama öğretimlerinin öğrencilerin erişimi, mantıksal düşünme ve programlamaya yönelik motivasyonlarına etkilerini belirlemektir. Yapılan çalışmada iki farklı deneysel işlem gerçekleştirilmiş ve sonuçlar ayrı ayrı değerlendirilmiştir. A şubesindeki öğrencilere 10 hafta süresince Scratch blok tabanlı programlama dili ve B şubesinde Small Basic ile metin tabanlı bir programlama öğretimi gerçekleştirilmiştir. Araştırma kapsamında veri toplama aracı olarak programlama erişimi testi, mantıksal düşünme testi ve programlamaya yönelik motivasyon ölçeği kullanılmıştır. Blok tabanlı görsel programlama öğretiminin yapıldığı öğrencilerin erişimi, mantıksal düşünme ve motivasyon puanlarında istatistiksel olarak anlamlı bir artış elde edilmiştir. Metin tabanlı programlama öğretiminin yapıldığı grupta ise erişimi ve mantıksal düşünme puanlarında istatistiksel olarak anlamlı bir artış göstermiştir. Ancak motivasyon puanlarındaki etki düşük düzeyde bulunmuştur.

Baştemur-Kaya (2018) çalışmasında Alice programı ile programlama öğretiminin; öğrencilerin akademik başarısına, problem çözme becerisi algısına, motivasyonuna ve programlamaya hazır bulunuşluk düzeyine etkisini, Alice programı ile ilgili öğrenci değerlendirmelerini belirlemektir. Araştırmada karma yöntem kullanılmıştır. Hem deney hem de karşılaştırma grubunda Gagne'nin dokuz aşamalı

öğretim modeli göz önünde bulundurularak öğretim tasarlanmıştır. Nitel veriler analiz edildiğinde öğrencilerin Alice programının temel kod kavramlarının öğrenilmesini ve programlama mantığını anlamayı kolaylaştırdığı, programlamayı öğrenme isteğini arttırdığını ifade ettikleri görülmüştür. Ancak Alice programında kod yazılamaması nedeniyle kod yazma becerisinin geliştirilememesi ve Türkçe dil desteğinin olmamasını olumsuz yönde eleştirdikleri; Alice programını, ileri seviye programlama için yetersiz gördüklerini dile getirmişler. Nicel veriler sonucunda akademik başarı, problem çözme becerisi algısı ve motivasyon bakımından iki grubun arasında anlamlı bir farklılık bulunmamıştır. İki grupta yer alan öğrenciler Java programlama dilini öğrenmeye odaklanmışlardır.

Baştuğ (2019) çalışmasında ortaokul düzeyinde programlama eğitiminde hem Scratch gibi blok mantığının kullanıldığı, hem de bu kodların karşılığı olan “JQuery” kodlarını içeren bir yazılım geliştirilerek öğrencileri ileri seviye programlama dillerine hazırlamak amacıyla araştırma yapmıştır. 6. sınıf öğrencilerinin programlamaya yönelik tutumlarını değiştirmede etkililiği ve öğrenci görüşleri üzerinden değerlendirilmiştir. Bulgular sonucunda öğrencilerin programlamaya yönelik tutumlarını, özellikle kendi özgün tasarımlarını yaptıktan sonra büyük oranda olumlu yönde değiştirdiğini göstermiştir. Ancak programlamaya yönelik motivasyon ve programlamayı önemli görme durumlarında belirgin bir değişim gözlenmemiştir.

2.6 Bilgisayarca Düşünme Becerileri

Bilgisayarca düşünme, bir problemi çözme aşamasında matematiksel düşünme, mühendislik bakış açısıyla karmaşık yapıları tasarlama ve değerlendirme, akıl, zekâ ve beşeri davranışları gibi kavramları anlamada bilimsel düşünme ile ortak yollar kullanır (Korkmaz, Çakır ve Özden, 2015). 21. yüzyıl ortalarında bilgisayarca düşünmenin temel yetenekler arasında yer alacağı öngörülmektedir. Bilgisayarca düşünmenin her düzeyden öğrenci için gerekli olduğunu vurgulamış ve bilgisayar bilimcileri için de önerilerde bulunmuştur (Wing, 2006).

ISTE (2019) göre bilgisayarca düşünme, teknolojik gelişmelere karşı düşünme gücünü birleştiren problem çözme yaklaşımıdır. SLO kuruluşu, CSTA, ISTE ve Kennisnet ile işbirliği içinde bilgisayarca düşünmenin geliştirilmesi için müfredat çalışmaları yürütmektedir. SLO (2020) göre problemlerin süreç tabanlı yeniden formüle edilmesi aşamasında bilgisayar teknolojisi ile çözüme ulaşılmasıdır.

Bilgisayarca düşünme süreci için gereken tutumlar vardır. Bunlar; karmaşıklık ve belirsizlikle baş edebilme, zor ve açık problemlere karşı azimli olma; ortak bir hedefe ulaşmak için diğerleriyle iletişim kurma ve işbirliği içinde olma olarak belirlenmiştir (SLO, 2020).

Bilgisayarca düşünme, yaratıcılık, mantıklı ve eleştirel düşünmenin yerini almaz, bilgisayarları kullanarak problem çözerken tüm düşünme becerilerini geliştirir (Korkmaz, Çakır, Özden, Oluk ve Sarıoğlu, 2015). Bilgisayar bilimleri eğitimi, bilgisayarca düşünme disiplinleri arasında yer alan problem çözme ve algoritmik düşünmeye yardımcı olmaktadır (Barr ve Stephenson, 2011).

Araştırmalara göre bireyin bilgisayarca düşünmenin bir problemi çözebilmek için gerekli düşünme sürecine hakim olmasına yönelik bir düşünme stratejisi olduğu ve bazı özellikleri kapsadığı söylenmektedir (Barr, Harrison, Conery, 2011). Bu özellikler aşağıdaki gibidir:

- Problemi teknolojik araçların yardımı olabilecek şekilde formüleleştirme
- Verileri mantıklı bir şekilde düzenleme ve analiz etme
- Soyutlayarak verileri sunma
- Algoritmik düşünme ile çözümleri otomatikleştirme
- Amaca ulaşırken analiz etme, belirleme ve en verimli aşamalar yardımıyla olası çözümleri uygulama
- Problem çözme sürecini problem çeşitliliğine dönüştürme ve yaygınlaştırma

Bu bulgulara göre bilgisayarca düşünmenin birçok alt beceriyi kapsadığını söylemek mümkündür. Bununla birlikte ISTE (2019) bilgisayarca düşünme becerisinin yaratıcılık, algoritmik ve eleştirel düşünme, problem çözme, işbirlikli öğrenme ve iletişim becerilerinin dışı yansıması olarak ifade etmektedir. Sonuç olarak bilgisayarca düşünme becerilerin anlaşılabilirliği için bu alt becerilerin doğru anlaşılması gerekmektedir.

Yaratıcılık: Yaratıcı düşünme yeni şeyler düşünmek, üretmek ve yeni yollarla düşünmek demektir. Yaratılan şeyler, fikir, ürün, renk ve kelimelerin yeni birleşimidir. İnsan ihtiyaçlarına karşı üretilen bilimsel buluşlar, sanat ve edebiyat eserleri yaratıcılık becerileri ile ortaya çıkar (Korkmaz vd., 2015; Tok, 2002).

Algoritmik düşünme: Günlük hayatımızda algoritmaları anlama, yorumlama, kullanma ve üretme becerisidir (Brown, 2015). Algoritmik düşünme, kurallar ve bu kuralların uygulanacağı adımları, problem çözerken çözüm yollarını ve problemin durumunu analiz etme olarak tanımlanmaktadır (Futschek, 2006).

İşbirliklilik: Karma büyük bir grup içerisinde alt kümelere ayrılarak ortak bir hedef doğrultusunda bilimsel bir çalışma için birbirlerine öğrenme konusunda yardımcı oldukları uzlaşma, paylaşma ve rekabet duygularını yaşadıkları ortamda sergilenen davranış biçimidir (Çaycı, K. Demir, Başaran ve M. Demir, 2007).

Eleştirel düşünme: Bir şeyi iyi ve kötü yanlarını değerlendirme anlamında kullanılır (Seferoğlu ve Akbıyık, 2006). Bir diğer deyişle; bireyin sahip olduğu bilişsel beceriler ya da stratejileri kullanması olarak anlatılmaktadır (Halpern, 1996).

Problem Çözme: Bireyin bir amaca ulaşırken karşılaştığı engellere genel olarak problem adı verilir (Korkmaz vd., 2015). Bu problemler ile başa çıkmak için geliştirilen strateji, teknik ve kurallar bütünü ise problem çözme becerilerimizi geliştirir. Ancak araştırmalar sonucunda her problemin aynı olmadığı, rutin olmayan problemleri çözüme ulaştırmak için genel stratejilerin dışına çıkılması gerektiği ortaya çıkmıştır. Temel problem çözme becerilerinin gelişmesi için bilişsel becerilere göre problemler analiz edilmelidir (Mayer, 1998).

Yukarıda yer alan bilgisayarca düşünmenin alt becerilerinde iletişim becerisi yer almamaktadır. Korkmaz, Çakır ve Özden (2017) çalışmasında iletişim becerilerinin yukarıda yer alan tüm becerileri kapsadığını belirterek ayrıca yer verilmesine gerek olmadığını savunmuşlardır. Son yıllarda ülkemizde de bilgisayarca düşünme diğer bir deyişle de komputasyonel, hesaplamalı veya bilgi işlemsel düşünme becerileri alanında çeşitli akademik çalışmalar yapılmıştır. Bu çalışmaların sayısı az olsa da gün geçtikçe yapılan akademik araştırmalar artmaktadır.

Ülkemizde yapılmış olan bilgisayarca düşünme üzerine yapılan alanyazın çalışmaları incelendiğinde betimsel tarama ya da deneysel yöntemlerin seçildiği gözlemlenmiştir. Deney çalışmalarında, programlama öğretimi sonrasında öğrencilerin bilgisayarca düşünme becerileri için hazırlanan ölçekler ile gelişim düzeyleri araştırılmıştır. Bu çalışmalara ilişkin özetler aşağıdaki gibidir:

Korkmaz, Çakır, Özden, Oluk ve Sarıoğlu (2015) çalışması betimsel tarama modelindedir. Bu araştırmanın çalışma grubunu 1306 birey oluşturmaktadır.

Araştırma kapsamında öğrencilerin bilgisayarca düşünme becerileri “Bilgisayarca Düşünme Becerileri Ölçeği” kullanılarak toplanmıştır. Elde edilen veriler üzerinde aritmetik ortalama, standart sapma, mod, medyan, frekans, t, Anova, LSD, ve korelasyon analizleri gerçekleştirilmiş, anlamlılık düzeyi 0.05 olarak kabul edilmiş ve şu sonuçlara erişilmiştir: Bireylerin bilgisayarca düşünme beceri düzeyine ilişkin algılarının yarısının yüksek, diğer yarısının ise orta düzeyde olduğu, belirlenmiştir. Teknoloji Fakültesi ve Enstitüde uygulanan programların, öğrencilerin bilgisayarca düşünme beceri düzeylerine diğer birimlere göre anlamlı derecede daha fazla katkı sağlamaktadır. Bulgulara göre matematik, fen ve teknoloji bölümlerinde uygulanan programların öğrencilerin bilgisayarca düşünme beceri düzeylerine diğer bölümlere göre anlamlı derecede daha fazla katkı sağladığı ortaya çıkmıştır.

Şahiner ve Kert (2016) komputasyonel düşünme kavramının 2006-2015 yılları arasında nasıl bir değişimde olduğunu incelemişlerdir. Araştırma sürecinde doküman analizinin adımları takip edilerek, belirlenen kriterler çerçevesinde 22 çalışma dâhil edilmiştir. Bu çalışmalar ScienceDirect, Taylor ve Francis ve SpringerLink veri tabanlarından elde edilmiştir. Çalışma sonucunda elde edilen bulgular makalenin genel bilgileri ve içerik bilgileri kapsamında araştırmışlardır. Sonuçlar son yıllarda komputasyonel düşünme kavramı ile ilgili araştırma alanlarının ve sayılarının arttığını ortaya çıkıştır.

Şahiner (2017) araştırmasında komputasyonel düşünme kavramının 2006-2016 yılları arasında nasıl bir değişimde olduğunu belirtmiştir. Araştırma sürecinde doküman analizinin adımları takip edilerek, belirlenen kriterler çerçevesinde 193 çalışma dâhil edilmiştir. Bu çalışmalar ScienceDirect, Taylor ve Francis, IEEE ve SpringerLink veri tabanlarından elde edilmiştir. Çalışma sonucunda elde edilen bulgularda makalenin genel bilgileri ve içerik bilgileri kapsamında incelenmiştir. Sonuçlar son yıllarda komputasyonel düşünme kavramı ile ilgili araştırma alanlarının ve sayılarının arttığını göstermiştir.

Kukul (2018), farklı yapılandırılmış programlama eğitimi süreçlerinin ortaokul öğrencilerinin bilgi-işlemsel düşünme becerilerine, bilgi-işlemsel düşünme öz yeterliliklerine ve programlama başarılarına etkisinin olup olmadığını araştırmıştır. Programlama başarısının ölçümü için de Scratch programına yönelik olarak oluşturulan 10 sorudan oluşmaktadır. Puanlar karşılaştırıldığında öğrencilerin bilgi-işlemsel düşünme görev testinden aldıkları puanlar arasında anlamlı bir farklılık

olmadığı ortaya çıkmıştır. Öğrencilerin farklı yapılandırılan programlama eğitimi süreçleri sonunda programlama başarıları karşılaştırıldığında da anlamlı bir farklılığın olmadığı belirlenmiştir.

Atiker (2019) araştırmasında 6. sınıf düzeyindeki öğrencilere programlama öğretiminde, bilgi-işlemsel düşünme becerilerini geliştirecek etkinlikler tasarlanıp, bu yöntemin öğrencilerin başarısına ve bilgi-işlemsel düşünme becerilerine etkilerini incelemiştir. Kontrol grubunda, geleneksel yöntem ile programlama öğretimi gerçekleştirilmiştir. Deney grubunda ise, ders etkinliklerine bilgi-işlemsel düşünme becerileri entegre edilmiştir. Derslerde problem çözme, gösterip yaptırma ve soru-cevap yöntemleri kullanılmıştır. Bulgular sonucunda, deney grubu lehine anlamlı bir farklılık olduğu sonucuna varılmıştır. Bilgi-işlemsel düşünme becerilerinin kazanımlara ilişkilendirildiği yöntem, akademik başarının artırılmasında etkili olmuştur.

2.6.1 Bilgisayarca Düşünme Becerisini Geliştirme. Papert (1980) kitabında ilk olarak bilgisayar bilimlerinde programlamanın ve bilgisayarca düşünme becerilerinin arasında ilişki olduğunu belirtmiştir. Programlama, öğrencilerin bilgisayarca düşünme yapılarını birçok disiplinde kolaylaştırabilecektir (Papert, 1980).

21. yüzyılda bilgisayarca düşünmenin birçok tanımı ise programlama ve bilgisayar bilimi ile birlikte yapılmaktadır (Nouri vd.,2019). 2006 yılında tekrar gündeme gelen bilgisayarca düşünmeyi müfredatlara entegre etme ve standartlaştırma çalışmalarında, bilgisayar biliminin temel kavramları kullanırken problem çözme, sistem tasarlama ve insan davranışlarını anlama olarak ele alınmıştır (Wing, 2006). Bilgisayarca düşünme ise problemleri ve çözümlerini formüle ederken kullanılan düşünce sürecini ifade etmektedir (Cuny, Snyder ve Wing, 2010). Her iki durumda da bilgisayarca düşünme, herkes için bilgisayar bilimlerinde ortak yaklaşım, ilke ve kavramlar ele alınmaktadır (Nouri vd., 2019).

Bilgisayarca düşünme, artan popüleritesine rağmen neleri kapsadığı konusunda belirsizlik devam etmektedir (Brennan ve Resnick, 2012). Bilgisayarca düşünmenin tanımlarında genellikle şu unsurlar ele alınmaktadır; soyutlama, algoritmalar, veri, problem ayrışması, paralellik, hata ayıklama, test etme ve kontrol yapısı gibi birbirlerini destekleyen kavramlardır (Rose, Habgood ve Jay, 2007). Bu kavramlar programlama kazanımlarında da yer aldığını söylemek mümkündür. Brennan ve Resnick

(2012) çalışması çerçevesinde bilgisayarca düşünmeyi üç boyuta ayırmaktadır. Bunlar şu şekilde sıralanmaktadır:

- Kavramlar; programlamada, yani diziler, döngüler, olaylar, koşullar, operatörler ve değişkenler;
- Uygulamalar; soyutlama ve modüleleştirme, kodu yeniden kullanma, test ve hata ayıklama, artımlı ve tekrarlayan olma;
- Perspektifler; kendini ifade etme, paylaşma ve başkalarının çalışmalarını temel olarak bağlantı kurma olarak belirlenmiştir.

Nouri ve arkadaşları (2019) programlama öğretiminde, bilgisayarca düşünmenin alt temalarında yaratıcı problem çözmenin daha öne çıktığını vurgulamışlardır. Aynı zamanda programlama öğretiminde, metin temelli dil kullanılması sonucunda bilgisayarca düşünme uygulama boyutunda ilerleme kaydedilebileceğini belirtmişlerdir. Perspektif boyutunda ise blok temelli dillerin çevrimiçi kullanılması sebebiyle daha ilgi çekici bulunmuştur (Nouri vd., 2019).

Bölüm 3

Yöntem

Araştırmanın bu bölümünde, araştırma modeli, evren, örneklem, çalışma grubu, verilerin toplanması, öğretim materyali, uygulama süreci ile birlikte verilerin analizine yer verilmiştir.

3.1 Araştırma Modeli

Bu araştırmada açıklayıcı sıralı karma yöntem deseni kullanılmıştır. Karma yöntemde nicel ve nitel araştırma verilerinin birlikte kullanılmasıdır. Açıklayıcı sıralı karma desende birinci aşamada nicel verilerin toplandığı, bulguların analiz edildiği daha sonra bulguların kullanılarak ikinci aşamada nitel verilerin planlandığı yöntemdir (Creswell, 2017).

Bu araştırmanın nicel boyutunda tek grup ön test-son test araştırma deseniyle yapılmıştır. Bu desende, bir gruba önce ön test ölçümü, sonrasında deneysel işlem uygulanır ve sonunda son test uygulanır (Creswell, 2017). Deneysel desenlerde değişkenler arasındaki neden sonuç ilişkisini test etmek temel amaçtır (Büyüköztürk, Kılıç Çakmak, Akgün, Karadeniz ve Demirel, 2016). Seçkisizlik ve eşleştirme olmayan bu desende uygulama olarak yapılan deneysel işlem tek bir grup üzerinde incelenir, desende tek bir gruba ait ön test ve son test ortalamaları arasında farkın ($O_1 - O_2$) anlamlılığı test edilir (Büyüköztürk vd., 2016). Desenin simgesel gösterimi Tablo 1’de verilmiştir.

Tablo 1

Tek Grup Ön Test-Son Test Desen (Büyüköztürk vd., 2016)

Grup	Ön test	İşlem	Son test
G	$O_1 + O_2$	X	$O_2 + O_3$

O₁: Blok Temelli Programlama Öz-Yeterlilik Algısı Ölçeği

O₂: Bilgisayarca Düşünme Becerisi Düzey Testi

O₃: Metin Temelli Programlama Başarı Testi

X: Deneysel İşlem

Bu araştırmada mevcut okul yapısı ve müfredatı göz önünde bulundurulduğunda sınıf düzeylerinde farklı çalışmaların yapılması mümkün olmadığından tek grup ön test ve son test deseni tercih edilmiştir. Dolayısıyla araştırmanın konusunda yer alan blok temelli öz-yeterlilik algılarının kontrol dahilinde ölçülebilmesi için, araştırmacının 5. sınıfta da ders verdiği öğrencilerden oluşan 6. sınıf düzeyi olarak belirlenmiştir.

Bu araştırma ile blok temelli öz-yeterlilik algısına sahip olan ve her öğrenciye bir bilgisayarın düştüğü bir sınıf ortamında, ortaokul 6. sınıf öğrencilerinin BTY dersinde metin temelli programlama öğretimi ile bilgisayarca düşünme becerilerine olan etkisi incelenmiştir.

Buradan yola çıkılarak metin temelli programlama öğretimi uygulamasından sonra, bağımsız değişken olan blok temelli öz-yeterlilik algısının, bağımlı değişkenler olan bilgisayarca düşünme becerisinin gelişimi ve programlama akademik başarısı üzerindeki etkisinin belirlenmesi amaçlanmıştır. Öğrencilerin blok temelli öz-yeterlilik algıları ve bilgisayarca düşünme becerileri ön test, metin temelli programlama başarıları ve bilgisayarca düşünme becerileri son test olarak uygulanmıştır.

Araştırmanın nitel boyutunda toplanan veriler içerik analizi ile değerlendirilir. İçerik analizi, belirli kurallar doğrultusunda kodlamalar yapılarak metin içindeki sözcüklerin daha küçük içerik kategorileri ile özetlendiği sistematik, yinelenebilir bir sistem olarak tanımlanmaktadır (Büyüköztürk vd., 2016). Araştırma modeli kapsamında BTY dersinde metin temelli programlama öğretimi alan öğrencilerle görüşme yapılmıştır. Görüşmelerde metin temelli programlama öğretimi için hazırlanan ders içi etkinlikler hakkında ve öğrencilerin akademik başarıları, bilgisayarca düşünme becerileri ve blok temelli öz-yeterlilik algılarının farkındalıkları ile ilgili açık uçlu sorularla görüşmeler yapılmıştır.

3.1.1 Değişkenler. Değişken, bir durumdan diğerine farklılık gösteren bir özelliktir (Büyüköztürk vd., 2016). Neden ve sonuç ilişkisinde değişkenlerin net olarak ifade edilmesi gerekmektedir (Creswell, 2017). Araştırma kapsamında yer alan bağımlı ve bağımsız değişkenler aşağıdaki gibidir:

3.1.1.1 Bağımsız Değişken. Bağımsız değişken, araştırmanın bağımlı değişken üzerindeki etkisinin test edilmesi istenilen değişkendir (Büyüköztürk vd., 2016). Sonucu etkileyecek ya da ölçümleri yapılacak olan bu değişkenlerdir (Creswell, 2017). Bu sebepten dolayı, bu araştırmanın bağımsız değişkeni metin temelli programlama öğretimi, cinsiyet ve blok temelli programlama aracını kullanmaya devam etme olarak belirlenmiştir.

3.1.1.2 Bağımlı Değişken. Bağımlı değişken, herhangi bir bağımsız değişken sonucu ortaya çıkan sonuç veya ölçüt değişkeni veyahut üzerine etkisi araştırılan değişkenlerdir (Creswell, 2017; Can, 2019). Bu araştırmanın bağımlı değişkeni ise blok temelli öz-yeterlilik algısı, bilgisayarca düşünme becerileri ve akademik başarıları olarak belirlenmiştir.

3.2 Çalışma Grubu

Bu çalışmada örneklem yöntemi olarak basit seçkisiz örneklem yöntemi kullanılmıştır. Bu örnekleme yönteminde evrendeki katılımcılar, örneğe seçilmek için eşit ve bağımsız şansa sahiptirler. Bu sebepten ötürü tüm bireylerin seçilme olasılığı aynıdır (Büyüköztürk vd., 2016).

Araştırmanın hedef evreni İstanbul ilinde seçilmiş özel bir okulda, BTY derslerinde programlama çalışmaları yapan ortaokul öğrencileridir. Araştırmanın örneklemini özel bir okulda 2018-2019 öğretim yılında 6. sınıfta öğrenim gören öğrenciler oluşturmaktadır.

Araştırmanın örnekleminde haftada 1 ders saati olarak BTY dersi işleyen ve derslerinde programlama çalışmaları yapılan 7 sınıf şubesi, 74 kız, 91 erkek olmak üzere toplamda 165 öğrenci yer almaktadır. Başlangıçta 165 öğrenci sayısı ile uygulamaya başlanmış olsa da 12 haftalık süreç boyunca derse devam sağlayan ve son testlere eksiksiz katılım gösteren öğrenci sayısı 138 öğrenciden meydana gelmiştir. Çalışma grubundaki 3 öğrenci MEB tarafından belirlenmiş BEP (Bireyselleştirilmiş Eğitim Programı), 2 öğrenci RAM (Rehberlik ve Araştırma Merkezi) okulun rehberlik

birimi ile grup çalışması kapsamında değerlendirmeye alınmamıştır (MEB, 2018). 22 öğrenci ise okul gezisi ve turnuvalar, bayram tatili vb. sebebi ile son haftalarda derse katılım sağlayamadıkları için bilgisayarca düşünme becerileri son testine katılamamış ve dolayısıyla araştırmanın dışında tutulmuştur. Bu sebep ile araştırma grup büyüklüğü toplamda 138 kişi ile yürütülmüştür.

Araştırmanın genel yapısı itibariyle çalışma grubunu daha önceden blok temelli ve eğitsel metin temelli programlama dili öğretimi almış olması gerekmektedir. Araştırmacının kendisi, çalışma grubunda bulunan öğrencilere 2017-2018 öğretim yılında 5. sınıf BTY dersi eğitimini veren öğretmen olmasından dolayı bu düzey seçilmiştir. Çalışma grubunda bulunan öğrenciler 5. sınıfta iken blok temelli programlama aracı olan Scratch aracı ile programlama öğretimini 12 hafta boyunca almışlardır.

Çalışma grubu tek gruplu deneysel desen kullanıldığı için çalışma grubu heterojendir. Öğrencilerin akademik düzeyleri bireysel başarı olarak incelendiğinde bazı öğrencilerin başarılı bazıları ise orta düzeydedir. Fakat, sınıf düzeylerinin akademik başarı düzey oranları birbirine yakındır, okul müdür yardımcısından alınmış listeye göre sınıfların gelen başarı ortalamaları aşağıdaki Tablo 2’de verilmiştir.

Tablo 2

Çalışma Grubunda Yer Alan Öğrencilerin Sınıflarına göre Genel Akademik Başarı Puanları

Sınıf	6A	6B	6C	6D	6E	6F	6G
Puan	92,32	89,39	90,02	89,70	90,06	90,14	88,38

Tablo 2 incelendiğinde, çalışma grubunda bulunan öğrencilerin genel başarı ortalamalarının birbirine yakın olduğu görülmektedir.

Okul yapısı incelendiğinde, BTY dersinin yapıldığı sınıflarda 1 adet öğretmen bilgisayar, 1 adet etkileşimli akıllı tahta ve öğrencilerin kişi başı bir bilgisayar kullanabilecekleri 24 adet bilgisayar bulunmaktadır. Ayrıca, her öğrencinin bilgisayarında kendine ait yerel kullanıcı hesabı, elektronik posta hesabı ve her sınıfın öğretmen ile bağlantılı sanal sınıf yönetim sistemleri vardır.

3.2.1 Öğrencilerin Demografik Özellikleri. BTY derslerine katılan 7 sınıf şubesi, 64 kız, 74 erkek olmak üzere toplamda 138 öğrenci ile çalışma yürütülmüştür. Çalışma grubunun cinsiyete göre dağılımı ve sınıf şubelerine göre dağılımı Tablo 3 ve Tablo 4’te gösterilmiştir.

Tablo 3

Çalışma Grubunda Yer Alan Öğrencilerin Cinsiyete Göre Dağılımı

Cinsiyet	Sayı	Yüzde
	(n)	(%)
Kız	64	46,4
Erkek	74	53,6
Toplam	138	100

Tablo 3’e göre, çalışma grubundaki öğrencilerin %46,4 kız, %53,6 erkek öğrenci olduğu gözlemlenmektedir. Sınıf düzeylerinin toplamı göz önüne alındığında erkek öğrencilerin sayıca fazla olduğu görülmektedir.

Tablo 4

Çalışma Grubunda Yer Alan Öğrencilerin Sınıflarına Göre Dağılımı

Sınıf	Cinsiyet			
	Kız		Erkek	
	Sayı	Yüzde	Sayı	Yüzde
	(n)	(%)	(n)	(%)
6A	7	11,9	10	13,5
6B	11	12,7	11	14,9
6C	9	14,1	13	17,6
6D	10	15,6	12	16,2
6E	12	18,8	10	13,5
6F	8	12,5	12	16,2
6G	7	10,9	6	8,1
Toplam	64	100	74	100

Tablo 4 incelendiğinde çalışma grubunda bulunan öğrencilerin cinsiyete göre sınıflara dağılımı görülmektedir. 7 farklı şubeden oluşan örnekleme en çok kız oranı %18,8 ile 6E sınıfıdır, en az ise %10,9 ile 6G sınıfıdır. En çok erkek oranı ise %17,6 ile 6E sınıfıdır, en az ise %8,1 ile 6G sınıfıdır. Genel dağılıma bakılırsa 6G sınıfından çalışmaya katılan öğrenci sayısının daha az olduğu görülmektedir. Sınıflara göre cinsiyet incelendiğinde, şubelerdeki kız ve erkek öğrencilerin sayılarındaki dağılımın birbirine oldukça yakın olduğu gözlemlenmektedir.

3.3 Ders Materyalleri

Konu ile ilgili olarak araştırmacı tarafından daha önceden seçilmiş olan “Programlama Öğreniyorum” kitabı yardımcı materyal olarak kullanılmıştır. MEB Talim ve Terbiye Kurulu tarafından 24/01/2018 tarihinde onaylanmış BTY dersi (5-6. sınıflar) öğretim programına uygun yöntem ve teknikler kullanılarak ders planları hazırlanmıştır. Ders akışını öğrencilerin takip edebilmesi için çalışma kağıtları sanal sınıf üzerinden paylaşılmıştır. Araştırma kapsamında hazırlanan ders planları ve etkinliklerle ilgili ayrıntılı bilgiler aşağıda verilmiştir.

3.3.1 Ders Planları. Araştırma kapsamında, çalışma takvimine uyacak şekilde 8 ayrı ders planı düzenlenmiştir (Ek A). Ders planları hazırlandıktan sonra bir akademisyen, iki ayrı alan uzmanı ve bir bölüm başkanı tarafından incelenmiştir. Ders planları okula ait toplamda 3 kampüsteki 6. sınıf düzeyinde uygulanmıştır. Ders planlarını hazırlayan araştırmacı aynı zamanda kampüsler arası alınan karar kapsamında 6. sınıf düzeyinin ders müfredatını ve ders planlarını hazırlayan düzey koordinatörüdür. Ders planları hazırlanmadan önce MEB (2018) müfredatında yer alan kazanımlara uygun olarak metin temelli programlama kazanım listesi ve Bloom Taksonomisine göre belirtke tablosu oluşturulmuştur.

Ders planları hazırlanırken Gagne'nin bilişsel öğrenme kuramına göre öğretim modeli olan “9 Aşamalı Öğretim Modeli” kullanılmıştır. Gagne'in öğretme yaklaşımına göre birden fazla kuram bir arada kullanılabilir (Fidan, 2012). Dolayısıyla araştırmacı ders içeriklerini planlarken birden fazla kuram, strateji ve yöntemden faydalanmıştır. Ausubel'in geliştirdiği “Anlamlı Öğrenme” modeline göre sunuş yoluyla öğretme stratejisi ile ders içeriğindeki kavramlar basitten karmaşığa öğrencilere sunularak verilmiştir. Bütün ders planları giriş, gelişme ve sonuç bölümlerine ayrılarak hazırlanmıştır.

Metin temelli programlama dili olarak çoklu paradigmaya sahip, nesne tabanlı ve fonksiyonel programlama dili olan JavaScript ve HTML seçilmiştir. Programlama dilinin seçilmesinde etkenlerden biri, zümre toplantısında kampüs öğretmenleri tarafından 6. sınıf düzeyinde, metin temelli programlama diline geçiş için web programlamanın uygun olacağı kararı alınmasıdır. Dolayısıyla 6. sınıf düzey öğretmenleri tarafından JavaScript ve HTML dili seçilmiştir. Bir diğer etken ise, web programlama dillerinin hem MEB hem uluslararası kuruluşlar tarafından (CSTA, SLO gibi) müfredatlarında ortaokul 6. sınıfta web programcılığına yer verilmiş olması da göz önünde bulundurulmuştur.

Her öğrenci başına bir bilgisayar düşen sınıf ortamında, “Gösterip Yaptırma” ve “Uygulama” öğretim teknikleri kullanılmıştır. Araştırmacı, sınıf içerisindeki etkinlikleri planlarken “dikkat çekme”, “hedeften haberdar etme” ve “güdüleme” adımlarında sıklıkla “Soru-Cevap” tekniği sıklıkla kullanmıştır.

Programlama öğretimi kapsamında, öğrencilerin bilgisayarca düşünme becerilerinin gelişmesi için alt boyutlara ait becerilere ders planlarında dikkat edilmiştir. Bilgisayarca düşünme becerileri göz önünde tutularak öğrencilerin bilişsel gelişim dönemlerine ve hazırbulunuşluklarına uygun ders etkinlikleri hazırlanmıştır.

Öğrencilerin yaratıcılık becerisinin gelişmesi için, özgün web sayfası tasarımları geliştirmeleri için CSS komut yapıları verilmiştir. Kendi tercihlerini yaparak oluşturacakları web sayfalarında bu komutların kullanılmasına özen gösterilmiştir. Ders etkinliklerinde tercihen öğrencilerin kendi seçimlerini yapmalarına izin verilmiştir.

Algoritmik düşünme becerileri için ders etkinliğinde geliştirilecek web sayfasının program yapısı için gerekli algoritma öğrenciler ile birlikte geliştirilmiştir. Soru-cevap tekniği ile geliştirilecek olan algoritmanın yapısı öğrenciler tarafından belirlenmiştir, öğretmen daha çok rehberlik yapmıştır.

İşbirliklilik becerisi için öğrencilerin program geliştirme sırasında akranlar arası yardımlaşmalarına öğretmen tarafından izin verilmiştir. Bu sayede sınıf ortamında birlikte hata ayıklama ve algoritma geliştirme sürecinde işbirliği yapılabilmıştır.

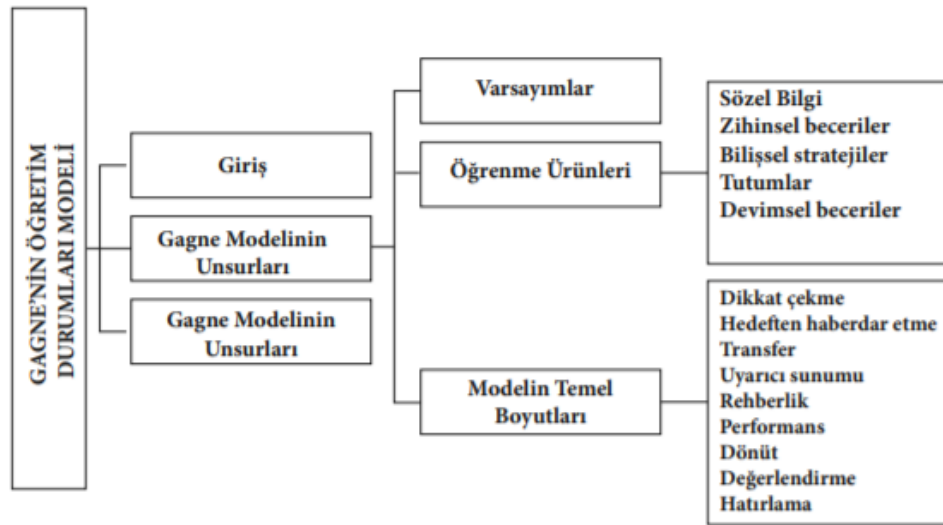
Öğrenciler eleştirel düşünmesi için öğretmen tarafından sık sık sorularla teşvik edilmiştir. Bu süreç içerisinde farklı çözüm önerisi sunan öğrencilerin bu deneyimlerini paylaşmalarına izin verilmiştir. Öğretmen, öğrencilerden ders sonunda

geliştirdikleri program yapısı ile fikirlerini paylaşmalarını istemiştir.

Ders etkinlikleri öğrencilere bir problem şeklinde sunulmuştur. Bu sayede öğrenciler geliştirilecek olan program yapısının inşası sırasında öncelikle probleme çözüm aramışlardır. Daha sonra algoritmik yapılar adım adım oluşturularak web sayfalarını tasarlamışlardır. Bu sayede ilk olarak problem çözme becerisi teşvik edilmiştir.

3.3.1.1 Uygulama Öncesi Hazırlık: Ders Planlarının Tasarımı. Öğrenme ve öğretme sürecinin nasıl planlanacağı ile ilgili hazırlıkları yapıldığı bu aşamada, çalışmanın bir öğretim kuramına dayalı olarak yürütülmesi hedeflenmiştir.

Bilgiyi işleme süreci ve davranışçı yaklaşımın ilkelerinin birleştiği, amacı öğrencilerin problem çözme becerilerinin gelişmesi olan, öğrenmenin aşamalı ve birikimli planlanmış bir süreç olduğunu savunan, öğrencinin aktif katılımıyla gerçekleşen bilişsel öğretim kuramı Gagne'nin 9 aşamalı öğretim modeli, bu çalışmada kullanılmıştır. Şekil 5'de Gagne'nin 9 aşamalı modeline ait unsurlar gösterilmektedir.



Şekil 5. Gagne'nin öğretim durumları modeli.

Öğretme sürecinde, yeni bilgilerin daha önceden öğrenilmiş bilgi ve beceriler üzerine, basitten karmaşığa zihinsel becerilerin hiyerarşi içinde oldukları, öğrencinin kendi yaptıkları ile aktif katılım sağladıkları, öğrenci ve öğretmenin her aşamasını ulaşılacak hedefe göre planladığı bir öğretim modelidir.

Modelin temel boyutlarına göre giriş aşamasında dikkat çekme, hedeften haberdar etme, ön bilgilerin hatırlatılması, güdüleme, gelişme aşamasında uyarıcının sunulması ve davranışı ortaya çıkarma, sonuç aşamasında ise dönüt-düzeltilme verme, değerlendirme ve hatırlama adımlarını içermektedir (Ekici, 2016).

Bu kurama göre öğrencinin içsel süreci, çevredeki dışsal uyarıcılar tarafından etkilenir, belli hedefleri öğrenmeye destelemek için dışsal uyarıcıların planlanması, sürdürülmesi ve değerlendirilmesi gerekmektedir (Senemoğlu, 1997).

Dikkat çekme aşamasında dışsal bir uyarıcı ile öğrenciyi tetikte tutmak amaçlanmaktadır. Dersin başında kullanılan bu uyarıcılar merak duygusunu harekete geçirmektedir (Senemoğlu, 1997). Çalışma kapsamında hazırlanan ders planlarında en çok kullanılan uyarıcılar ise soru-cevap tekniği ile konu hakkında öğrencilerin fikir yürütmesi, etkinliğe benzer bir web sayfasının gösterilmesi, konu hakkında hazırlanmış slaytlar, resimler ya da hikayelerden yararlanılmıştır.

Öğrenci bir bilgiyi öğrenmeden önce amacını bilmek istemektedir. Öğrenciyi hedeften haberdar etme aşamasında araştırma kapsamındaki hedeflere göre konu hakkında bilgilendirilmiştir. Çünkü, merak ve beklenti öğrencinin yoğun olarak öğrenme çabasını güdülemesine yardım eder, performansını artırır ve öz-yeterlilik algılarının gelişimine katkı sağlar (Senemoğlu, 1997).

Güdüleme ya da ön öğrenmelerin hatırlanmasını sağlama aşamasında uyarıcıları vermeden önce, yeni öğrenmeyle ilgili olan önceki öğrenmelerin kısa süreli belleğe çağrılmaktadır (Senemoğlu, 1997). Bu şekilde ön koşul öğrenmeler hazırlanarak eski ve yeni bilgiler arasında ilişkiler kurulur. Çalışmanın bu aşamasında ise araştırmacı programlama kavramlarının güdülemesinde, öğrencilerin eski öğrenmeleri olan blok temelli programlama ve tablolama öğretiminden faydalanmıştır.

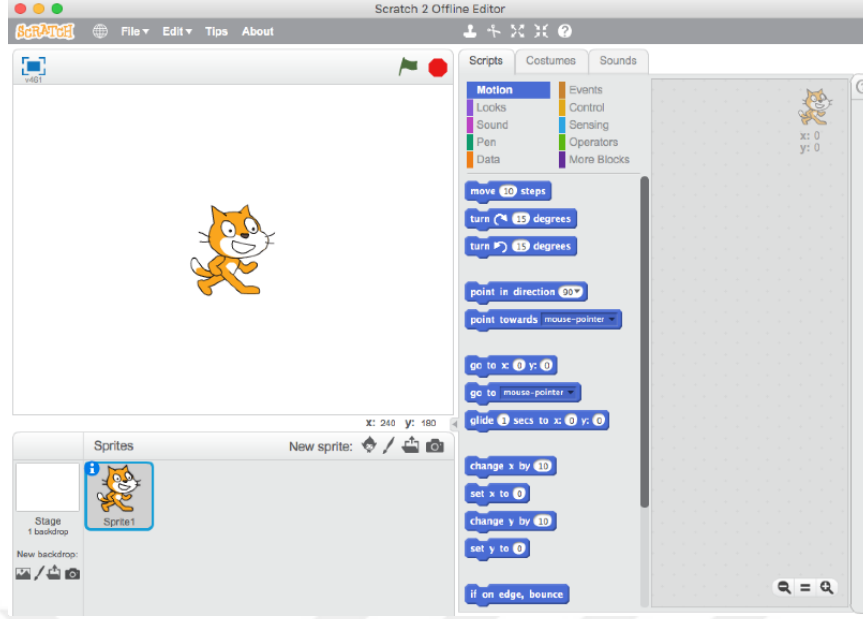
Gagne'ye göre uyarıcıların sunulması aşaması öğrenilecek ürüne göre farklılık göstermektedir (Ekici, 2016). Öğrenme ürünleri beş bölümden oluşmaktadır: sözel bilgi, zihinsel beceriler, bilişsel stratejiler, tutumlar ve devinimsel (motor) beceriler. Çalışmanın kapsamında metin temelli programlama becerisi kazandırılmak istenen öğrenci grubunun öğrenme ürünü Gagne'ye göre bilişsel stratejilerde yer almaktadır. Eğer bilişsel strateji öğretilecekse sözel olarak anlatım ya da öğretmen stratejiyi kendisi uygulayarak adım adım gösterebilir (Senemoğlu, 1997). Dolayısıyla araştırmacı, uyarıcıların sunulması aşamasında öğrencilere kavramları sözel olarak anlatırken

programlamada kullanılacak komutları öğrencilere uygulayarak göstermiş ve daha sonra onların da uygulamalarını istemiştir. Sonuç olarak bu aşamada gösterip-yatırma tekniği de kullanmıştır.

Anlamalı öğrenmenin gerçekleşmesi için rehberlik aşamasında kodlama taktiği sıkça kullanılmıştır. Rehberlik aşamasındaki amaç öğrencinin öğrendiklerini depolamasına ve geri getirmesine yardımcı olmaktır (Senemoğlu, 1997). Programlama kavramlar, basitten karmaşığa hedeflere göre alt kavramlara bölünmüştür. Araştırmacı konu hakkında grafik ve şekillerden de faydalanmıştır.

BTY dersi kapsamında eğer okul şartları uygunsa öğrencinin bilgiyi uygulaması öğrenmenin gerçekten oluşup oluşmadığını gösterir ve davranışın öğrenci tarafından gösterilmesi ile daha da anlamlı kılar. Çalışmanın uygulandığı okul şartlarında her öğrenci başına bir bilgisayar düştüğü için performansı ortaya çıkarma aşamasında araştırmacı tarafından öğretilen bilgilerin her öğrenci tarafından uygulamasını istemiştir. Dönüt sağlama ve performansı değerlendirme aşamalarında araştırmacı doğru davranışların pekiştirilmesi, yanlışların ise anında düzeltilmesi için ders akışı sırasında sık sık öğrencilerin arasında dolaşarak bire bir dönütler vermiştir. Her dersin sonunda öğrencilerden kullandıkları komutları arayüz programında kaydederek çalıştırmalar istenmiştir, programı çalışmayan öğrencilere anında dönütler verilmiştir. Performansların düzenli kontrol edilmesi için öğrencilerin kendi bilgisayar hesaplarında oluşturdukları klasörlere kaydetmesi sağlanmıştır.

3.3.1.2 Çalışma Kapsamında Önemi Araştırılan Blok Temelli Programlama Ortamının Özelliklerinin İncelenmesi. Çalışma grubu 5. sınıfta blok temelli programlama öğretiminde kullanılan Scratch 2.0 arayüz programı MIT Medya Laboratuvarı tarafından geliştirilmiştir. Şekil 6'da programın arayüzü bulunmaktadır.



Şekil 6. Scratch 2.0 programı arayüz görünümü.

Scratch arayüz programı, müzik, video, görsel gibi çeşitli medya araçlarını kütüphanesinde bulunduran animasyon, dijital hikaye ya da bilgisayar oyunlarının blok yapılarıdaki programlama yapısı kullanılan bir grafik programlama dilidir.

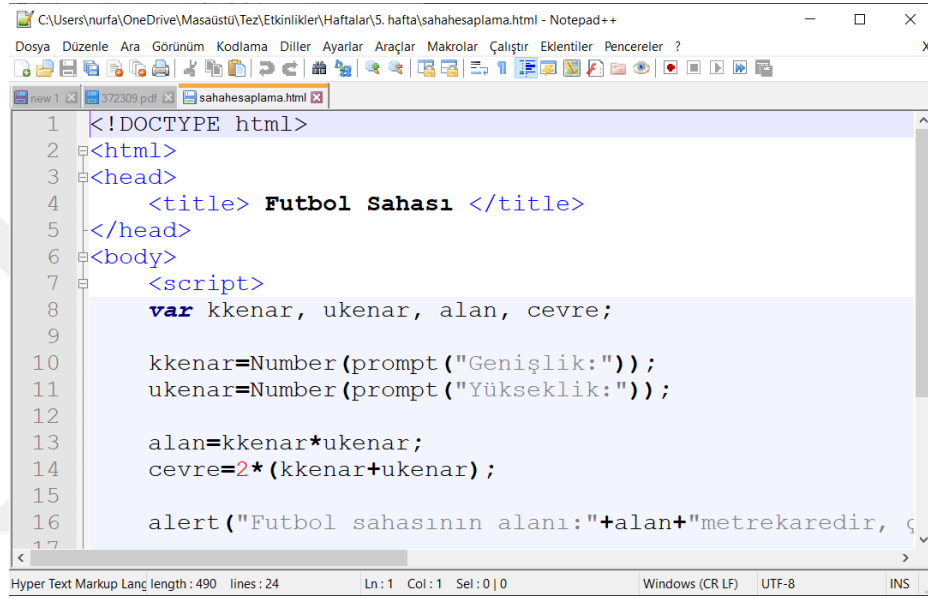
Grafik programlama dili olmasından kaynaklı alışlagelmiş sayfalarca komut satırları yerine çocukların ilgisini çekmektedir. Sürükle-bırak yöntemi ile blok yapı komutlarının kullanıldığı bir ortamdan oluşmaktadır. Okulun yapısı incelendiğinde 2017-2018 eğitim öğretim yılında öğrencilerin yaş seviyesine uygun olarak Scratch blok temelli programlama aracı seçilmiştir.

3.3.1.3 Çalışma Kapsamında Kullanılan Metin Temelli Programlama Ortamının Özelliklerinin İncelenmesi. Bu bölümde, çalışma grubunda Windows işletim sistemi içerisinde gömülü olarak bulunan Notepad yazılımı yerine kullanılmak üzere C++ ile saf Win32 API ile SRTL ile geliştirilmiş GPL ile dağıtılan açık kaynak komutlu bir komut düzenleyici ya da derleyici programı olan “Notepad++” tercih edilmiştir. Notepad++ arayüz programının, birçok yazılıma ait dili görüntüleyen ve uygun olarak komutları renklendiren, komut dizilimlerininin kısaltılması için açılır-kapanır komut özellikleri vardır.

Notepad++ programını tasarlayanlar tarafından arayüz kullanımında kullanıcı tanımlı komut renklendirme, komut satırlarını büyütüp küçültme, sürükle bırak

desteđi, yazılım dili için komut tamamlama, birden fazla sayfa ekranında çalışma, çalışma ekranını bölümlere ayırma ve pencerelerde dinamik görüntü sağlama gibi imkanlar kılmaktadır.

Blok temelli programlama öğretimi görmüş olan öğrencilerin, metin temelli programlama öğretimi için birçok yazılım dilini desteklemesi, komutları tamamlama özelliđi olması, komut satırlarını renklendirilmesi ve arayüzünün kullanışlı olması nedeniyle Notepad++ programı seçilmiştir. Şekil 5'de programın arayüzü bulunmaktadır.



```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title> Futbol Sahası </title>
5 </head>
6 <body>
7   <script>
8     var kkenar, ukenar, alan, cevre;
9
10    kkenar=Number(prompt("Genişlik:"));
11    ukenar=Number(prompt("Yükseklik:"));
12
13    alan=kkenar*ukenar;
14    cevre=2*(kkenar+ukenar);
15
16    alert("Futbol sahasının alanı:"+alan+"metrekaredir, ç
17
```

Şekil 7. Notepad++ programı arayüz görünümü.

3.4 Uygulama Süreci

Araştırma sürecinde hazırlanan metin temelli programlama ders planları her hafta 1 ders saati olmak üzere 8 hafta boyunca araştırmacı tarafından 7 sınıf şubesinde bulunan öğrenciler ile birlikte uygulanmıştır. Uygulama öncesinde 2 hafta ön hazırlık sürecinde ön testler, deney aşamasından sonraki 2 hafta ise başarı testi ve son test uygulaması için ayrılmıştır. Toplamda 12 hafta süren bir çalışma olmuştur.

3.4.1 Uygulama. Uygulama, araştırmacı ile birlikte 2018-2019 eğitim öğretim yılının ikinci döneminde 6 A-B-C-D-E-F-G sınıfları ile bilgisayar laboratuvarında BTY dersinde haftada 1 ders saati içinde uygulanmıştır. Uygulama 18/03/2019 ve 20/06/2019 tarihleri arasında 12 haftalık bir süreci kapsamaktadır.

Birinci hafta öğrencilerin 5. sınıfta öğrenmiş oldukları blok temelli

programlama aracı öz-yeterlilik algılarının tespit edilmesi için “Blok Temelli Öz-Yeterlilik Algısı Ölçeği” ve ikinci hafta ise bilgisayarca düşünme becerileri ön seviyelerinin belirlenmesi için “Bilgisayarca Düşünme Becerileri Testi” uygulanmıştır. Öğrencilerin bilgisayar kullanma becerileri oldukça yüksek olduğundan ve öğretmen ile birlikte kullandıkları sanal sınıf yönetim sistemi aracılığıyla ölççekler paylaşılmış olup, öğrenciler kendileri ölççekleri tamamlamışlar ve öğretmene teslim etmişlerdir.

Ön testler tamamlandıktan sonra öğrencilere, 8 haftalık süreçte ne tür etkinlikler ile programlama öğretimi yapılacağından ve ders akışının nasıl olacağından, metin temelli programlama aracından bahsedilmiştir. 12 haftalık uygulama süreci Tablo 5’te yer almaktadır.

Tablo 5

Araştırmanın Uygulama Süreci

Haftalar	Yapılan Etkinlikler
1. Hafta (18/03/2019)	Blok Temelli Öz-Yeterlilik Algısı ölçeğinin uygulanması
2. Hafta (25/03/2019)	Bilgisayarca Düşünme Becerileri ön test uygulanması
3. Hafta (01/04/2019)	Metin temelli programlama aracı ve temel komutlar
4. Hafta (15/04/2019)	Metin ve tablo işlemleri, form hazırlama komutları
5. Hafta (22/04/2019)	Bağlantı oluşturma, çerçeve ve stil şablonu komutları
6. Hafta (29/04/2019)	Resim, görüntü ve çoklu ortam komutları
7. Hafta (06/05/2019)	Sabit ve değişkenler, karar yapıları oluşturma
8. Hafta (13/05/2019)	Mantıksal işlem operatörleri, çoklu karar yapıları
9. Hafta (20/05/2019)	Karar ve döngü yapıları oluşturma
10. Hafta (27/05/2019)	Hazır fonksiyon kullanma ve fonksiyon oluşturma
11. Hafta (10/06/2019)	Metin Temelli Başarı testinin uygulanması
12. Hafta (17/06/2019)	Bilgisayarca Düşünme Becerileri son test uygulanması

Üçüncü hafta itibariyle metin temelli programlama etkinliklerine ders planlarının temel akışı, BTY dersi öğretme-öğrenme yöntem ve teknikleri temel alınarak ileri letilmiştir. Ders planlarındaki konular, Bloom’un “Tam Öğrenme Modeli”nde yer alan, konular küçük birimlere bölünerek aşamalı olarak verilmiştir

(Bloom, 1956). Onuncu hafta sonunda tüm etkinlikler tamamlanmış ve son iki haftada “Metin Temelli Programlama Başarı Testi” ve “Bilgisayarca Düşünme Becerileri Testi” son test olarak uygulanmıştır.

3.5 Verilerin Toplanması

Bu araştırma durum araştırması methodu kullanılarak yapılmış bir çalışma olup verilerin toplanması sırasında açılımlayıcı sıralı karma yöntem kullanılmıştır. Durum araştırması, bilimsel sorulara yanıt aramada kullanılan ayırt edici bir yaklaşımdır. Durum çalışmalarını, bir ya da daha fazla olayın, ortamın, programın, sosyal grubun ya da birbirine bağlı sistemlerin derinlemesine araştırıldığı yöntemler olarak bilinmektedir (Büyüköztürk vd., 2016). Durum araştırması kapsamında veriler hem nitel hem nicel yöntemle toplanabilmektedir. Nitel yöntemler; görüşme, sesli düşünme, açık uçlu sınavlar iken anket, ölçek, çoktan seçmeli sınavlar ise nicel yöntemleri oluşturmaktadır. Araştırmaya derinlik katabilmek, geçerli ve güvenilir sonuçlar elde edebilmek adına nitel ve nicel yöntemler bir arada kullanılabilir. Araştırma kapsamında okul yönetimi, bölüm başkanı ve öğrencilerden gerekli izinler alınmıştır.

3.5.1 Veri Toplama Araçları

Bu çalışmada nitel veri toplama aracı olarak öğrenci görüşleri, nicel veri toplama aracı olarak da bilgisayarca düşünme becerileri ölçeği, blok temelli programlamaya ilişkin öz-yeterlilik algıları ölçeği ve metin temelli akademik başarı testi kullanılmıştır.

3.5.1.1 Bilgisayarca Düşünme Beceri Düzeyleri Testi (Ortaokul Düzeyi için).

Alanyazın incelendiğinde bilgisayarca düşünme beceri düzeyleri için farklı ölçüm araçlarına rastlanmaktadır. Bu ölçüm araçlarından bazıları ortaokul düzeyi için uygun olmamakla birlikte bazıları bilgisayarca düşünme becerileri alt düzeylerini incelemede kısıtlı kaldığı görülmektedir (Özden, 2015).

Hem ortaokul düzeyine uygun olması hem de bilgisayarca düşünme beceri düzeylerine dünya standartlarında yaklaşımı sebebiyle, bu çalışmada Korkmaz, Çakır ve Özden (2015) tarafından geliştirilmiş Bilgisayarca Düşünme Beceri Düzeyleri Testi kullanılmıştır (Ek B). Ölçek 5’li likert tipi, beş faktör (yaratıcılık, algoritmik düşünme, işbirliklilik, eleştirel düşünme, problem çözme) altında toplanabilen 22 maddeden oluşmaktadır. Ölçekten alınabilecek maksimum puan 100, minimum puan ise 20’dir. Faktörlerden yaratıcılık, algoritmik düşünme, işbirliklilik ve eleştirel düşünme

becerileri birbirlerinden bağımsız puanları maksimum 20, minimum 5 olarak hesaplanmıştır. Problem çözme becerisinin maksimum puanı ise 30, minimum puanı 6 olarak belirlenmiştir.

Ölçeğin iç-tutarlılık katsayıları, Cronbach Alpha güvenilirlik formülü ile hesaplanmıştır. Ölçeğin güvenilirlik katsayısı 0,809, faktörlere ilişkin güvenilirlik katsayıları ise 0,640 ile 0,867 arası değerler aldığı belirlenmiştir. Maddeler; hiçbir zaman (1), nadiren (2), bazen (3), genellikle (4), her zaman (5) şeklinde derecelendirilmiştir. Ölçekteki beşli likert yapısına verdikleri cevaplar, alt ölçeklerdeki sayı nedeniyle değişkenlik gösterebilmektedir. Ölçekten alınan puanlar şu şekilde özetlenmektedir: 20-51: Düşük Düzey; 52-67: Orta Düzey; 68-100: Yüksek Düzey.

Araştırma kapsamında bilgisayarca düşünme beceri düzeyleri testi, öğrencilerin metin temelli programlama öğretimi sonrasında bilgisayarca düşünme becerileri, düzeyleri ve seviyeleri üzerindeki etkisi olup olmayacağı amacıyla kullanılmıştır.

3.5.1.2 Blok Temelli Programlamaya İlişkin Öz-yeterlilik Algısı Ölçeği.

Öğrencilerin blok temelli öz-yeterlilik algılarının belirlenmesi için iki faktörün, 5 maddeden oluşan birinci faktörü “basit blok temelli programlama görevleri”, 7 maddeden oluşan ikinci faktörü ise “karmaşık blok temelli programlama görevleri” olan Altun ve Kasalak (2018) tarafından geliştirilmiş bir ölçek seçilmiştir (Ek C).

Ölçek 5’li likert tipi, basit blok temelli programlamaya ilişkin öz-yeterlilik algısı için alınabilecek maksimum puan 25, minimum puan ise 5, karmaşık blok temelli programlamaya yönelik öz-yeterlilik algısı için alınabilecek maksimum puan 35, minimum puan ise 5’tir. Ölçekte bulunan her maddenin madde toplam korelasyonu 0.3’ün üstünde, toplamda 12 maddenin madde toplam korelasyonları 0,491-0,702 arasında hesaplanmıştır. Ölçeğin güvenilirlik katsayısı Cronbach's Alpha ile 0,893 olarak bulunmuştur.

Bu ölçek ile araştırma kapsamında metin temelli programlama öğretimi öncesinde blok temelli öz-yeterlilik algılarının bilgisayarca düşünme becerileri ve akademik başarıları arasındaki ilişkiyi incelemek amaçlanmıştır.

3.5.1.3 Metin Temelli Programlamaya Yönelik Akademik Başarı Testi.

Araştırmada metin temelli programlama akademik başarısına yönelik verilerin toplanması için programlama geliştirme becerilerini ölçen başarı testi araştırmacı tarafından hazırlanmıştır (Ek D). Akademik başarı testi hazırlanırken ilgili BTY dersi (5-6. sınıflar) öğretim programındaki kazanımlar ve soruların bilişsel düzeylerini gösteren belirtke tablosuna dikkat edilerek hazırlanmıştır (Ek E).

Bloom ve arkadaşları (1956) tarafından oluşturulmuş bilişsel alan boyutu ile belirtke tablosu hazırlanmıştır. 40 yılı aşan farklı teoriler önerilse de Bloom'un orjinal taksonomisi hala güncelliğini korumaktadır. Gelişen yapılandırmacı yaklaşım, ölçme değerlendirme araçları ve öğrenci merkezli eğitimin öne çıkması ile Bloom'un taksonomisinin revize edilmesinde öneriler atılmıştır (Yurdabakan, 2012). Bloom'un bilişsel alan olarak nitelendirdiği alan bilişsel süreç ve bilgi olarak iki boyuta bölünmüştür (Krathwohl, 2000). Bilişüstü becerilerde en iyi verimin ders sonunda biriktirilen hedef ve kazanımlara yönelik ürünlerin olması gerektiği vurgulanmıştır. Bu şekilde orjinal taksonominin eksik yönleri kapatılmıştır. Revize edilmiş basamağın en üstünde bulunan "yaratma" kritik bir nokta olduğu ve özen gösterilerek toplumlar için faydalı modeller geliştirilmesi gerektiği açıklanmıştır (Tutkun, 2012). Başarı testi soru havuzu hazırlanırken öğretim programının kazanımlarını karşılayacak şekilde olmasına özen gösterilmiştir.

Araştırmada kullanılan akademik başarı testinin kapsam geçerliliğini sağlamak için bir akademisyen, üç ayrı alan uzmanı ve bir dil uzmanı mevcut soruları incelemiştir. Alınan geri bildirimler sonucunda akademik başarı testine son hali verilerek 25 sorudan oluşan bir pilot akademik başarı testi hazırlanmıştır.

Pilot akademik başarı testi metin temelli programlama öğretimi almış, araştırmada kullanılan programlama dili olan HTML, JavaScript 'e ve kullanılan arayüz programı olan Notepad++'a hakim iki farklı okulda çalışan BTY dersi öğretmenleri tarafından toplamda 129 6. sınıf öğrencisine uygulanmıştır.

Öğrenciler testten aldıkları puanlara göre sıralanmış, madde analizi yapmak için %27'lik alt ve üst grup seçilerek gerekli işlemler yapılmıştır. Maddenin ayırıcılık indeksi 0,40 ve daha yüksek değerlerde olan maddeler için çok iyi, 0,30 ile 0,39 değerleri arasında olan maddeler için oldukça iyi, 0,20 ile 0,29 değerleri arasında olan maddeler için düzeltilmesi ve geliştirilmesi gerektiği, 0,19 ve daha düşük değerlerde olan maddeler için çok zayıf ve testten çıkarılması gerektiği şeklinde değerlendirildi ve bu doğrultuda

25 sorudan 2, 3, 8 ve 9. soru olmak üzere 4 soru testten çıkartılmıştır. Kalan sorulardan 6 soruda düzenlemeler yapılarak testin son hali oluşturulmuştur. Metin temelli programlama akademik başarı testi pilot uygulamasında bulunan sorulara ait madde ayırt edicilik ve güçlük indeksleri Tablo 6’da verilmiştir.

Tablo 6

Metin Temelli Programlama Başarı Testi Pilot ve Son Haline Ait Madde İstatistikleri

Pilot Çalışma			Testin Son Hali				
Madde No	Güçlük İndeksi	Ayrılcılık İndeksi	Madde No	Güçlük İndeksi	Ayrılcılık İndeksi	Varyans	Std. Sapma
1	0,80	-0,16	1	0,66	0,76	0,22	0,34
2	0,94	0,11	-	-	-	-	-
3	0,95	0,09	-	-	-	-	-
4	0,89	0,16	2	0,83	0,61	0,14	0,17
5	0,83	0,32	3	0,72	0,74	0,21	0,28
6	0,75	0,21	4	0,62	0,42	0,24	0,38
7	0,41	-0,02	5	0,84	0,53	0,13	0,16
8	0,91	0,16	-	-	-	-	-
9	0,94	0,12	-	-	-	-	-
10	0,55	0,60	6	0,65	0,68	0,23	0,35
11	0,55	0,60	7	0,72	0,63	0,20	0,28
12	0,63	0,27	8	0,72	0,39	0,20	0,28
13	0,88	0,23	9	0,48	0,65	0,25	0,52
14	0,82	0,34	10	0,70	0,66	0,21	0,30
15	0,63	0,65	11	0,65	0,74	0,23	0,35
16	0,63	0,65	12	0,65	0,71	0,23	0,35
17	0,58	0,37	13	0,55	0,74	0,25	0,45
18	0,86	0,14	14	0,73	0,61	0,20	0,27
19	0,86	0,14	15	0,89	0,45	0,11	0,11
20	0,79	0,28	16	0,86	0,45	0,12	0,14
21	0,85	0,21	17	0,82	0,61	0,15	0,18
22	0,80	0,30	18	0,41	-0,05	0,24	0,59
23	0,76	0,07	19	0,76	0,76	0,18	0,24
24	0,64	0,62	20	0,67	0,81	0,22	0,33
25	0,47	0,51	21	0,67	0,81	0,22	0,33

3.5.2 Veri Toplama İşlemleri. Araştırmada kullanılan nicel veriler, 2018-2019 eğitim öğretim yılı ikinci döneminde toplanmıştır. Çalışma grubuna ilk hafta 2017-2018 eğitim öğretim yılında 5. sınıfta BTY dersinde öğrendikleri blok temelli programlama aracı anımsatıldıktan sonra “Blok Temelli Programlama Öz-Yeterlilik Algıları Ölçeği”, ikinci hafta ise herhangi bir açıklama yapılmadan “Bilgisayarca Düşünme Beceri Düzeyleri Ölçeği” uygulanmıştır.

Öğrencilerin bilgisayar kullanım becerileri düzeyleri yüksek olmasından dolayı ölçekler öğrencilerle ile sanal sınıf aracılığı ile paylaşılarak araştırmacı tarafından ölçeklerin “Google Form” uygulaması ile hazırlandıktan sonra her öğrenciye bir bilgisayar düşen sınıf ortamında doldurmaları sağlanmıştır.

Ön testler tamamlandıktan sonra 8 haftalık uygulama süreci başlatılmış ve 3. hafta itibariyle ders akışlarının nasıl olacağından bahsedilmiştir. 8 hafta boyunca planlanan 8 farklı ders planı ve etkinlikler tamamlandıktan sonra son testlerden “Metin Temelli Programlama Akademik Başarı Testi” uygulanarak öğrencilerin akademik düzeyleri belirlenmiştir. Son haftada “Bilgisayarca Düşünme Beceri Düzeyleri Ölçeği” tekrar uygulanarak son test verileri elde edilmiştir.

3.5.2.1 Öğrenci Görüşmeleri. Çalışma grubunda bulunan öğrencilere, yapılan uygulama ile ilgili görüşlerini ayrıntılı inceleyebilmek amacıyla görüşme formu kullanılmıştır. Görüşme formları hazırlanırken uzman görüşleri doğrultusunda sorulacak sorular dört ana tema olarak belirlenmiştir. Bu temalar; “Ders içi uygulama ve etkinliklere dair genel”, “Bilgisayarca düşünme beceri düzeylerine yönelik”, “Akademik başarıya yönelik” ve “Blok temelli öz-yeterlilik algılarına yönelik” görüşme altında toplamıştır. Araştırmacı tarafından öğrencilerin görüşlerini belirleyecek her tema altında açık uçlu sorular yazılmıştır. Görüşme formu hazırlanırken öğrencilerin seviyesine uygun açık uçlu sorular hazırlanmıştır (Ek F). Rastgele seçilen 6 öğrenci ile açık uçlu sorular ile görüşmeler gerçekleştirilmiştir.

3.5.3 Verilerin Analizi. Araştırmada karma yöntem kullanıldığından dolayı hem nitel hem nicel veriler toplanmıştır. Bu nedenle veriler nitel ve nicel veriler olarak incelenmiştir. Bu kapsamda nitel ve nicel veri analizi altında iki başlık altında veri analizi yapılmıştır. Belirlenen araştırma sorularına ilişkin hangi veri toplama aracının kullanıldığı ve ne şekilde verilerin analiz edildiği Tablo 7’de verilmiştir.

Tablo 7

Araştırma Soruları ve Analiz Yöntemleri

Araştırma Sorusu	Veri Toplama Aracı	Veri Analizi
Ortaokul öğrencilerinin blok temelli programlama öz-yeterlilik algıları, bilgisayarca düşünme becerileri ve akademik başarıları ne düzeydedir?	BDBOT BDBST BTPOT MTPBT	Betimsel Analiz
Metin temelli programlama öğretiminde öğrencilerin, akademik başarıları cinsiyete göre anlamlı farklılık göstermekte midir?	MTPBT	İlişkisiz Örneklemler t-Testi
Metin temelli programlama öğretiminde, öğrencilerin bilgisayarca düşünme becerilerine göre ön test ve son test puanları arasında anlamlı farklılık var mıdır?	BDBST BDBOT	İlişkili Örneklemler t-Testi
Bilgisayarca düşünme becerileri alt boyutlarına göre ön test ve son test puanları arasında anlamlı farklılık var mıdır?		
Ortaokul öğrencilerinin, blok temelli programlama öz-yeterlilik algıları, bilgisayarca düşünme becerileri ve akademik başarıları, blok temelli programlama aracını kullanmaya devam etme durumuna göre anlamlı farklılık göstermekte midir?	BDBST BTPOT MTPBT	Çok Yönlü Varyans Analizi (MANOVA)
Ortaokul öğrencilerinin, blok temelli programlama öz-yeterlilik algıları, bilgisayarca düşünme becerileri ve akademik başarıları, ön düşünme düzeyine göre anlamlı farklılık göstermekte midir?	BDBST BTPOT MTPBT	Çok Yönlü Varyans Analizi (MANOVA)
Ortaokul öğrencilerinin, tasarlanan ders etkinliklerine, bilgisayarca düşünme becerileri ve blok temelli programlama öz-yeterlilik algılarına yönelik betimsel görüşleri nelerdir?	Öğrenci Görüşme Formu	Kategori- Tema

3.5.3.1 Nicel Veri Analizi. Çalışmada elde edilen nicel veriler SPSS The Statistical Package for the Social Sciences) programı kullanarak analiz edilmiştir. Bilgisayarca Düşünme Beceri Düzeyleri Ölçeği, Blok Temelli Programlamaya Yönelik Öz-Yeterlilik Algısı Ölçeği ve Metin Temelli Programlama Başarı Testi ile elde edilen puanların normal dağılımları katımlıların grup sayısının fazla ve heterojen bir yapıda olması sebebiyle Kolmogorov-Smirnov testine ait p değeri, merkezi eğilim ve dağılım ölçüleri, çarpıklık ve basıklık katsayıları, histogram ve normal kantil grafiklerine dair sonuçlar incelenmiştir. Bilgisayarca düşünme becerileri için yapılan son test uygulamasına rehberlik çalışmaları, gezi, yarışma vb. nedenlerle katılamayan 24 öğrencinin sonuçları değerlendirmeye alınmamıştır. BEP öğrencisi olan 3 öğrenci ise araştırma kapsamı dışında tutulmuştur.

3.5.3.2 Nitel Veri Analizi. Araştırmada nitel verileri, gönüllülük esasına dayanarak seçilen altı öğrenci ile açık uçlu sorular ile görüşmeler yapılmıştır. Bununla birlikte görüşmelere ek olarak araştırmacı tarafından nitel verilerin analizini desteklemek adına uygulama esnasında fotoğraf ve video kayıtları da alınmıştır. Görüşmeler yaklaşık olarak 30 dakika sürmüştür ve öğrenciler görüşlerini bir kağıda yazmışlardır. Görüşmeler sonucu elde edilen veriler içerik analizi tekniği ile analiz edilmiştir. İçerik analizi belirli kurallar doğrultusunda kodlamalar yapılarak metin içindeki sözcüklerin daha küçük içerik kategorileri ile özetlendiği sistematik, yinelenebilir bir sistem olarak tanımlanmaktadır (Büyüköztürk vd, 2016).

3.5.4 Geçerlik ve Güvenirlik. Bir araştırmada nitel ya da nicel araştırma yöntemlerinden biri kullanılacağı gibi iki yöntem birlikte de kullanılabilir. Sonuçların büyük gruplar için genellenebilirliği için sonuçların faktörler ile açıklanması gerekmektedir. Bağımlı değişkenlerdeki değişmelerin, bağımsız değişkenler ile açıklanabilirliğine “iç geçerlilik”; büyük gruplara genellenebilirliğine ise “dış geçerlilik” olarak belirtilmiştir (Büyüköztürk vd., 2016).

Bu çalışmanın geçerli ve güvenilir olması için veri toplama araçlarının geçerlik ve güvenilirliğine dair çalışmalar yapılmıştır. Çalışma sürecinde gerekli geçerlik ve güvenilirlik önlemleri alınmıştır. Geçerlik ve güvenilirliği sağlamak için aşağıda belirtilen aşamalar dikkate alınmıştır:

- Veri toplama ve analiz süreci ayrıntılı bir şekilde açıklanmıştır.
- Çalışma grubunun demografik özellikleri ve seçim şekli ayrıntılı bir

şekilde açıklanmıştır.

- Çalışmanın uygulama süreci ve kullanılan yöntemler gerekçeleri belirtilerek ayrıntılı bir şekilde açıklanmıştır.
- Çalışmanın sınırlılıklar ve varsayımları belirtilmiştir.
- Çalışmada kullanılan nicel veri toplama araçlarının geçerlik ve güvenirlik önlemleri alınmıştır.
- Uygulama alanındaki öğretmen, araştırmacının kendisidir. Uygulamalarda ve ölçek geliştirme sırasında alan uzmanları ve akademisyen desteği alınmıştır.
- Uygulama alanındaki öğretmen 5 yıl mesleki tecrübesi bulunmaktadır. Aynı zamanda çalışma grubundaki öğrencilerle uzun zamandır eğitim-öğretim faaliyetleri gerçekleştirmektedir.
- Uygulamada bütün sınıflarla aynı öğretmen ders anlatmıştır.
- Çalışma verileri 12 haftalık bir süreçte toplanmıştır.
- Çalışmanın aşamalarında gerekli uzman görüşleri alınmıştır.

3.6 Sınırlamalar

Bu araştırmanın sınırlamaları aşağıdaki gibidir:

1. 2018-2019 eğitim-öğretim ikinci dönem arası ile sınırlıdır.
2. Bu uygulama, İstanbul ili Sarıyer ilçesinde özel bir okulda 6. sınıfta öğrenim gören öğrenci ile sınırlıdır.
3. Bu araştırma BTY dersi ile sınırlıdır.
4. Araştırmanın uygulaması 12 hafta ve 12 ders saati ile sınırlıdır.
5. Araştırmada kullanılan metin temelli programlama öğretimi aracı “Notepad++”, programlama dili HTML ve JavaScript ile sınırlıdır. Akademik başarı belirleme testi olan “Metin Temelli Programlama Başarı Testi” programlama başarısız, orta, iyi ve çok iyi düzeyde hakim olabilme ve bu programlama dilini hedeflenen kazanımlara göre kullanabilme ile sınırlıdır.

Bölüm 4

Bulgular

Bu bölümde araştırmada toplanan verilerin analizi ile elde edilen bulgular araştırmada incelenen alt problemler doğrultusunda sunulmuştur. Veri analizi için IBM SPSS Statistics 22.0 istatistik programı kullanılmıştır. Analizlerde ele alınan anlamlılık düzeyi ($p \leq 0,05$) olarak kabul edilmiştir. Anlamlılık düzeyin olan p değeri ($p > 0,05$) olduğu durumlarda incelenen çalışmada anlamlı farklılığın olmadığı, ($p \leq 0,05$) olduğu durumlarda ise incelenen çalışmada istatistik anlamlı farklılığın var olduğu sonucuna varılır (Can, 2019).

Araştırma sonucunda elde edilen verilerin normal dağılım gösterdiği veri analizlerinde, ölçeklerin etki ve fark anlamlılıkları için parametrik test kapsamındaki İlişkili Örneklem t-Testi, İlişkisiz Örneklem t-Testi ve çok yönlü varyans analizi (MANOVA) kullanılmıştır.

Araştırma sorularına ilişkin analizlerde kullanılan ön test ve son testlerin puan dağılımlarına ilişkin normallik testlerine, ölçeklerin puanlarının aralarında anlamlı fark olmadığına yönelik bulgulara da yer verilmiştir.

4.1 Verilerin Normal Dağılımına Uygunluğun Belirlenmesi ve Betimsel İstatistikleri

Araştırmada kapsamında, çalışma grubunun seçkisiz ve heterojen tek gruptan oluşması nedeniyle normalliğe uygunluğu bağımsız değişken olan cinsiyete göre incelenmiştir. Cinsiyete göre grup ikiye bölünmüştür.

Gruplarda bulunan örneklem sayısı 50'nin üzerinde olmasından dolayı Kolmogorov-Smirnov testine ait p değeri incelenmiştir. Ölçeklerin merkezi eğilim ve dağılım ölçüleri, çarpıklık ve basıklık katsayıları, histogram ve normal kantil grafiklerine dair sonuçlar yorumlanmıştır. Araştırmalarda katılımcı sayısı 50-300 arasındaysa, çarpıklık ve basıklık katsayıları incelenirken kendi standart hatalarına bölünmeleri sonucunda çıkan rakamın (yani z skorların) -3,29 ve + 3,29 değerlerine

göre yorumlanır (Kim, 2013).

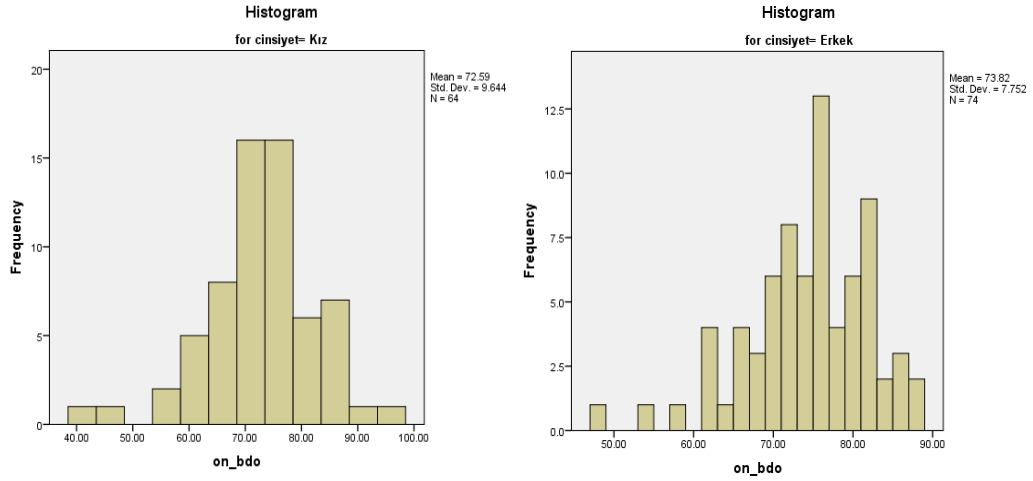
4.1.1 Çalışma Gruplarının Bilgisayarca Düşünme Beceri Düzeyleri Ölçeğinin Normal Dağılıma Uygunluğunu Gösteren Veriler. İncelenen araştırmada, çalışmaya katılan sağlayan grubun metin temelli programlama öğretimi yapılmadan önce ve sonra bilgisayarca düşünme becerilerinin gelişimini incelemek için Bilgisayarca Düşünme Beceri Düzeyleri Ölçeği ön test ve son test olarak uygulanmıştır. Bağımlı değişkenler olan bu ölçeklerin normal dağılım gösterip göstermediği aşağıdaki verilerde anlatılmıştır. Verilere ait bulgular Tablo 8 ve Tablo 9’da verilmiştir.

Tablo 8

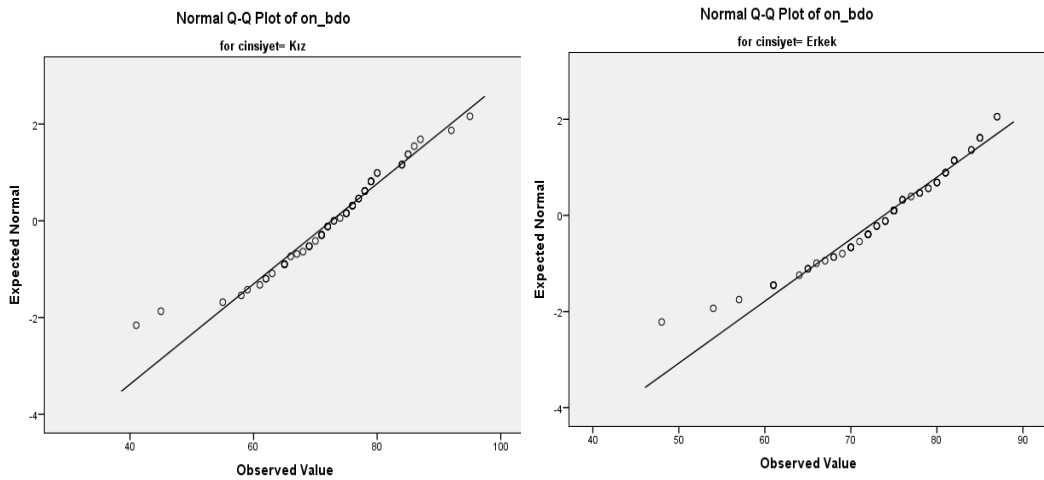
Bilgisayarca Düşünme Becerileri Ölçeğinin Ön test Sonuçlarının Cinsiyete Göre Kolmogorov-Smirnov (K-S) Ve Shapiro-Wilk Normal Dağılıma Uygunluk Testi Değerleri, Merkezi Eğilim Ve Dağılım Ölçüleri

Merkezi Eğilim ve Dağılım Parametreleri	Kız	Erkek
Normallik Testi	n=64	n=74
Ortalama	72,59	73,82
Ortalamanın Standart Hatası	1,21	,901
Medyan (Ortanca)	73	75
Mod	75	75
Standart Sapma	9,64	7,76
Varyans	93,1	60,1
Minimum	41	48
Maksimum	95	87
Çarpıklık	-,661	-,800
Çarpıklık Standart Hatası	,299	,279
Basıklık	1,714	,962
Basıklık Standart Hatası	,590	,552
Shapiro-Wilk Değeri	,055	,014
Kolmogorov-Smirnov Değeri	,200	,028

Normal dağılımı belirlemek amacıyla yapılan analiz tablosu olan Tablo 8'e göre, çalışma grubunun bilgisayarca düşünme becerileri ölçeğinin ön test puanları cinsiyete göre incelendiğinde Shapiro-Wilk ve Kolmogorov-Smirnov değerinin kızlarda 0,05'ten büyük olduğu, çarpıklık ve basıklık katsayısının -3,29 ile +3,29 arasında olduğu, ortancanın ortalamaya oldukça yakın olduğu görülmektedir. Ayrıca histogram ve kutu grafikleri de incelendiğinde bulgular **normal dağılım** göstermektedir. Erkeklerde ise Shapiro-Wilk ve Kolmogorov-Smirnov değerinin 0,05'ten küçük olduğu fakat çarpıklık ve basıklık katsayısının -3,29 ile +3,29 arasında olduğu, ortancanın da ortalamaya yakın olduğu görülmektedir. Bu verilerin dışında, histogram ve kutu grafikleri de incelendiğinde bulgular **normal dağılım** gösterdiği görülmektedir. Aşağıdaki Şekil 7.1 ve Şekil 7.2'de kız ve erkeklere göre histogram ve kutu grafiklerine ait bulgulara yer verilmiştir.



Şekil 8.1. BDBOT'nin cinsiyete göre histogram grafiği.



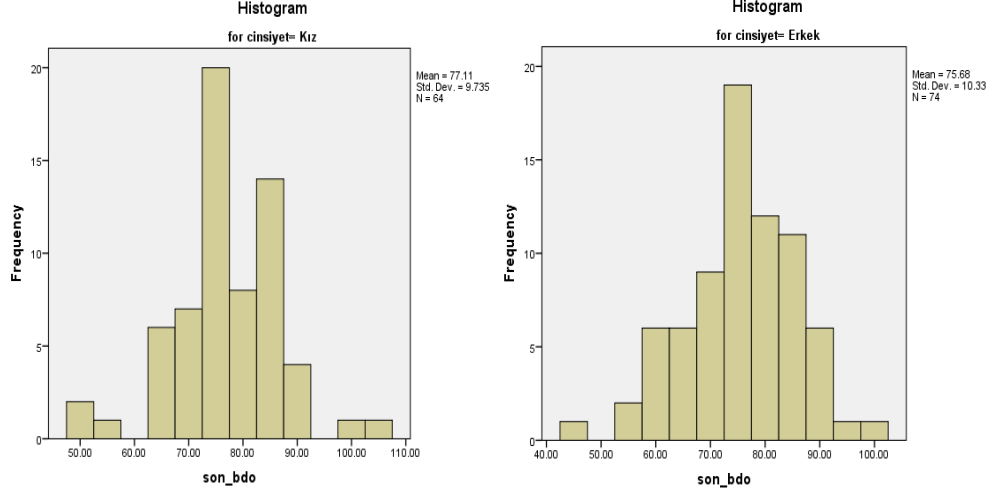
Şekil 8.2. BDBOT'nin cinsiyete göre normal kantil grafiği.

Tablo 9

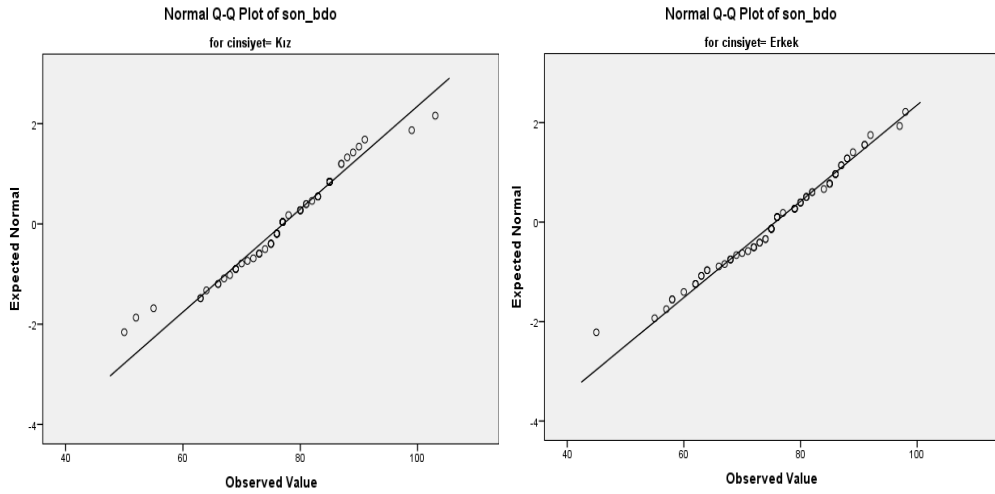
Bilgisayarca Düşünme Becerileri Ölçeğinin Son test Sonuçlarının Cinsiyete Göre Kolmogorov-Smirnov (K-S) Ve Shapiro-Wilk Normal Dağılıma Uygunluk Testi Değerleri, Merkezi Eğilim Ve Dağılım Ölçüleri

Merkezi Eğilim ve Dağılım Parametreleri	Kız	Erkek
Normallik Testi	n=64	n=74
Ortalama	77,11	75,68
Ortalamanın Standart Hatası	1,21	1,21
Medyan (Ortanca)	77	75
Mod	75	75
Standart Sapma	9,73	10,33
Varyans	94,76	106,7
Minimum	50	45
Maksimum	103	98
Çarpıklık	-,331	-,342
Çarpıklık Standart Hatası	,299	,279
Basıklık	1,107	,141
Basıklık Standart Hatası	,590	,552
Shapiro-Wilk Değeri	,122	,497
Kolmogorov-Smirnov Değeri	,097	,091

Normal dağılımı belirlemek amacıyla yapılan analiz tablosu olan Tablo 9'a göre, çalışma grubunun bilgisayarca düşünme becerileri ölçeğinin son test puanları cinsiyete göre incelendiğinde Shapiro-Wilk ve Kolmogorov-Smirnov değerinin kızlarda 0,05'ten büyük olduğu, çarpıklık ve basıklık katsayısının -3,29 ile +3,29 arasında olduğu, ortancanın ortalamaya oldukça yakın olduğu görülmektedir. Ayrıca histogram ve kutu grafikleri de incelendiğinde bulgular **normal dağılım** göstermektedir. Erkeklerde ise Shapiro-Wilk ve Kolmogorov-Smirnov değerinin 0,05'ten büyük olduğu, çarpıklık ve basıklık katsayısının -3,29 ile +3,29 arasında olduğu, ortancanın da oldukça ortalamaya yakın olduğu görülmektedir. Bu verilerin dışında, histogram ve kutu grafikleri de incelendiğinde bulgular **normal dağılım** gösterdiği görülmektedir. Aşağıdaki Şekil 7.3 ve Şekil 7.4'de kız ve erkeklere göre histogram ve kutu grafiklerine ait bulgulara yer verilmiştir.



Şekil 8.3. BDBST'nin cinsiyete göre histogram grafiği.



Şekil 8.4. BDBST'nin cinsiyete göre normal kantil grafiği.

4.1.2 Çalışma Gruplarının Metin Temelli Programlama Başarı Testinin Normal Dağılıma Uygunluğunu Gösteren Veriler. Çalışma gruplarının metin temelli programlama öğretiminden sonra programlama akademik başarılarını incelemek için Metin Temelli Programlama Başarı Testi son test olarak uygulanmıştır. Bağımlı değişken olan bu testin normal dağılım gösterip göstermediği aşağıdaki verilerde anlatılmıştır. Verilere ait bulgular Tablo 10'da yer almaktadır.

Tablo 10

Metin Temelli Programlama Başarı Testi Puanlarının Cinsiyete Göre Kolmogorov-Smirnov (K-S) Ve Shapiro-Wilk Normal Dağılıma Uygunluk Testi Değerleri, Merkezi Eğilim Ve Dağılım Ölçüleri

Merkezi Eğilim ve Dağılım Parametreleri	Kız	Erkek
Normallik Testi	n=64	n=74
Ortalama	80,25	78,04
Ortalamanın Standart Hatası	1,27	1,42
Medyan (Ortanca)	81	81
Mod	86	86
Standart Sapma	10,17	12,28
Varyans	103,49	150,80
Minimum	57	48
Maksimum	100	100
Çarpıklık	-,237	-,349
Çarpıklık Standart Hatası	,299	,276
Basıklık	-,460	-,393
Basıklık Standart Hatası	,590	,552
Shapiro-Wilk Değeri	,058	,056
Kolmogorov-Smirnov Değeri	,000	,010

Normal dağılımı belirlemek amacıyla yapılan analiz tablosu olan Tablo 10'a göre, çalışma grubunun metin temelli programlama başarı testi puanları cinsiyete göre incelendiğinde Shapiro-Wilk değerinin kızlarda 0,05'ten büyük olduğu fakat Kolmogorov-Smirnov değerinin 0,05'ten küçük olduğu görülmektedir. Bunun yanında, çarpıklık ve basıklık katsayısının -3,29 ile +3,29 arasında olduğu, ortancanın da ortalamaya yakın olduğu görülmektedir. Ayrıca histogram ve kutu grafikleri de incelendiğinde bulguların **normal dağılım** gösterdiğini söyleyebiliriz. Erkeklerde ise Shapiro-Wilk değerinin 0,05'ten büyük olduğu fakat Kolmogorov-Smirnov değerinin 0,05'ten küçük olduğu, çarpıklık ve basıklık katsayısının -3,29 ile +3,29 arasında olduğu, ortancanın da ortalamaya yakın olduğu görülmektedir. Bu verilerin dışında, histogram ve kutu grafikleri de incelendiğinde bulguların **normal dağılım** gösterdiği görülmektedir. Kız ve erkeklere göre histogram ve kutu grafiklerine ait bulgulara ekler bölümünde yer verilmiştir (Ek 9).

4.1.2 Çalışma Gruplarının Blok Temelli Programlamaya İlişkin Öz-Yeterlilik Algıları Ölçeğinin Normal Dağılıma Uygunluğunu Gösteren Veriler.

Çalışma gruplarının metin temelli programlama öğretiminden önce blok temelli programlamaya yönelik öz-yeterlilik algılarını incelemek için Blok Temelli Programlamaya İlişkin Öz-Yeterlilik Algıları Ölçeği ön test olarak uygulanmıştır. Bağımlı değişken olan bu ölçeğin normal dağılım gösterip göstermediği aşağıdaki verilerde anlatılmıştır. Verilere ait bulgular Tablo 11’de verilmiştir.

Tablo 11

Blok Temelli Öz-Yeterlilik Algısı Testinin Cinsiyete Göre Kolmogorov-Smirnov (K-S) Ve Shapiro-Wilk Normal Dağılıma Uygunluk Testi Değerleri, Merkezi Eğilim Ve Dağılım Ölçüleri

Merkezi Eğilim ve Dağılım Parametreleri	Kız	Erkek
Normallik Testi	n=64	n=74
Ortalama	40,58	43,15
Ortalamanın Standart Hatası	1,118	1,414
Medyan (Ortanca)	40,5	44
Mod	40	40
Standart Sapma	8,94	12,166
Varyans	79,93	148,02
Minimum	15	12
Maksimum	59	60
Çarpıklık	-,366	-,682
Çarpıklık Standart Hatası	,299	,279
Basıklık	-,217	,081
Basıklık Standart Hatası	,590	,552
Shapiro-Wilk Değeri	,288	,003
Kolmogorov-Smirnov Değeri	,200	,018

Normal dağılımı belirlemek amacıyla yapılan analiz tablosu olan Tablo 11’e göre çalışma grubunun blok temelli öz-yeterlilik algıları test puanları cinsiyete göre incelendiğinde Shapiro-Wilk ve Kolmogorov-Smirnov değerinin kızlarda 0,05’ten büyük olduğu, çarpıklık ve basıklık katsayısının 3,29 ile +3,29 arasında olduğu, ortancanın ortalamaya oldukça yakın olduğu görülmektedir. Ayrıca histogram ve kutu

grafikleri de incelendiğinde bulgular **normal dağılım** göstermektedir. Erkeklerde ise Shapiro-Wilk ve Kolmogorov-Smirnov değerinin 0,05'ten küçük olduğu fakat çarpıklık ve basıklık katsayısının -3,29 ile +3,29 arasında olduğu, ortancanın da ortalamaya yakın olduğu görülmektedir. Bu verilerin dışında, histogram ve kutu grafikleri de incelendiğinde bulgular **normal dağılım** gösterdiğini söyleyebiliriz. Kız ve erkeklere göre histogram ve kutu grafiklerine ait bulgulara ekler bölümünde yer verilmiştir (Ek 9). Uygulanan blok temelli programlama öz-yeterlilik algısı ölçeğinin iki alt boyutu bulunmaktadır. Bu alt boyutlar basit ve karmaşık blok temelli programlama olarak belirlenmiştir ve alt boyutların normal dağılımları incelenmiştir. Verilere ait bulgulara Tablo 12 ve Tablo 13'te yer verilmiştir.

Tablo 12

Basit Blok Temelli Öz-Yeterlilik Algısı Testinin Cinsiyete Göre Kolmogorov-Smirnov (K-S) Ve Shapiro-Wilk Normal Dağılıma Uygunluk Testi Değerleri, Merkezi Eğilim Ve Dağılım Ölçüleri

Merkezi Eğilim ve Dağılım Parametreleri	Kız	Erkek
Normallik Testi	n=64	n=74
Ortalama	18,34	19,28
Ortalamanın Standart Hatası	,525	,646
Medyan (Ortanca)	18,5	21
Standart Sapma	4,198	5,553
Varyans	17,626	30,836
Minimum	7	5
Maksimum	25	25
Çarpıklık	-,443	-,885
Çarpıklık Standart Hatası	,299	,279
Basıklık	-,182	,079
Basıklık Standart Hatası	,590	,552
Shapiro-Wilk Değeri	,043	,000
Kolmogorov-Smirnov Değeri	,002	,000

Normal dağılımı belirlemek amacıyla yapılan analiz tablosu olan Tablo 12'ye göre çalışma grubunun basit blok temelli öz-yeterlilik algıları test puanları cinsiyete

göre incelendiğinde Shapiro-Wilk ve Kolmogorov-Smirnov değerinin kızlarda 0,05'ten küçük olduğu, çarpıklık ve basıklık katsayısının 3,29 ile +3,29 arasında olduğu, ortancanın ortalamaya oldukça yakın olduğu görülmektedir. Erkeklerde Shapiro-Wilk ve Kolmogorov-Smirnov değerinin 0,05'ten küçük olduğu fakat çarpıklık ve basıklık katsayısının -3,29 ile +3,29 arasında olduğu, ortancanın da ortalamaya yakın olduğu görülmektedir. Bulguların çarpıklık ve basıklık katsayılarına göre **normal dağılım** gösterdiği görülmektedir.

Tablo 13

Karmaşık Blok Temelli Öz-Yeterlilik Algısı Testinin Cinsiyete Göre Kolmogorov-Smirnov (K-S) Ve Shapiro-Wilk Normal Dağılıma Uygunluk Testi Değerleri, Merkezi Eğilim Ve Dağılım Ölçüleri

Merkezi Eğilim ve Dağılım Parametreleri	Kız	Erkek
Normallik Testi	n=64	n=74
Ortalama	22,05	24,67
Ortalamanın Standart Hatası	,693	,838
Medyan (Ortanca)	21,5	25
Standart Sapma	5,542	7,211
Varyans	30,712	52,003
Minimum	7	7
Maksimum	35	35
Çarpıklık	,041	-,447
Çarpıklık Standart Hatası	,299	,279
Basıklık	-,182	-,310
Basıklık Standart Hatası	,654	,552
Shapiro-Wilk Değeri	,263	,015
Kolmogorov-Smirnov Değeri	,200	,200

Normal dağılımı belirlemek amacıyla yapılan analiz tablosu olan Tablo 13'e göre çalışma grubunun karmaşık blok temelli öz-yeterlilik algıları test puanları cinsiyete göre incelendiğinde Shapiro-Wilk ve Kolmogorov-Smirnov değerinin kızlarda 0,05'ten büyük olduğu, çarpıklık ve basıklık katsayısının 3,29 ile +3,29 arasında olduğu, ortancanın ortalamaya yakın olduğu görülmektedir. Erkeklerde ise

Shapiro-Wilk değeri 0,05'ten büyük fakat Kolmogorov-Smirnov değerinin 0,05'ten küçük olduğu gözlemlenmiştir. Bu bulguların dışında, çarpıklık ve basıklık katsayısının -3,29 ile +3,29 arasında olduğu, ortancanın da ortalamaya oldukça yakın olduğu görülmektedir. Bulguların çarpıklık ve basıklık katsayılarına göre **normal dağılım** gösterdiğini söyleyebiliriz.

4.2 Araştırma Problemlerine İlişkin Bulgular

4.2.1 Birinci Alt Probleme İlişkin Bulgular. Araştırmanın birinci alt problemi olan “Ortaokul öğrencilerinin blok temelli programlama öz-yeterlilik algıları, bilgisayarca düşünme becerileri ve akademik başarıları ne düzeydedir?” sorusuna yanıt aramak amacıyla öğrencilere yapılan Bilgisayarca Düşünme Beceri Düzeyleri Ölçeği ön test (BDBOT), son test (BDBST), Blok Temelli Programlamaya İlişkin Öz-Yeterlilik Algıları Ölçeği (BTPOT) ve Metin Temelli Programlama Başarı Testi (MTPBT) uygulanmıştır. Çalışma grubuna uygulanan ölçeklerin ve testin betimsel verilerin incelemeleri aşağıdaki tablolarda gösterilmektedir.

4.2.1.1 Bilgisayar Düşünme Beceri Düzeyleri Ölçeği Betimsel İstatistikler. Bilgisayarca düşünme becerilerinin ve beş alt boyutunun ön test ve son test puanlarına ilişkin betimsel istatistikler aşağıda verilmiştir.

BDBOT ve BDBST ortalama puanlarının betimsel istatistik tablosu incelendiğinde, metin temelli programlama öğretimi sonrasında bilgisayarca düşünme becerilerinin toplam oranında bir artış gözlemlenmektedir ($\bar{X}^{bdbst}=76,34 > \bar{X}^{bdbot}=73,25$). Beceri düzeylerinde ise “Problem Çözme” becerisinde ön teste oranla son testte yüksek oranda bir artış olmuştur ($\bar{X}=22,21 > 18,56$). “Yaratıcılık”, “Algoritmik Düşünme”, “İşbirliklilik” ve “Eleştirel Düşünme” düzeylerinde ise değişim gözlenmemektedir. Verilere ait bulgular Tablo 14’te verilmiştir.

Bilgisayarca düşünme becerisinin alt boyutlarından yaratıcılık, algoritmik düşünme, işbirliklilik ve eleştirel düşünme becerilerinin maksimum puanları 20’dir. Dolayısıyla öğrencilerin yaratıcılık, algoritmik düşünme, işbirliklilik ve eleştirel düşünme puanları maksimum puanlara oldukça yakındır. Problem çözme becerisi maksimum puanı ise 30’dur. Öğrencilerin ön test problem çözme beceri puanları buna kıyasla oldukça düşüktür. Ancak problem çözme becerisinin son test puanının oldukça yükseldiği hesaplanmıştır.

Tablo 14

BDBOT Ve BDBST Düzeylerine Göre Ortalama Puanlarına İlişkin Betimsel İstatistikler

BDBST BDBOT	\bar{X}	Medyan	Mod	Std. Sapma	Min	Max
BDBOT	73,25	74,5	75	8,66999	41	95
BDBST	76,34	76	75	10,04769	45	100
Yaratıcılık ^{ön test}	16,29	17	15	2,77215	6	20
Yaratıcılık ^{son test}	16,03	17	15	2,94708	9	20
Algoritmik Düş. ^{ön test}	14,46	15	14	3,63511	4	20
Algoritmik Düş. ^{son test}	14,15	15	14	3,82062	4	20
İşbirliklilik ^{ön test}	17,22	18	20	3,12050	5	20
İşbirliklilik ^{son test}	17,20	18,5	20	3,21983	6	20
Eleştirel Düş. ^{ön test}	14,01	14	14	3,73186	4	20
Eleştirel Düş. ^{son test}	13,61	14	13	4,15298	4	20
Problem Çöz. ^{ön test}	18,56	18	16	2,23708	13	24
Problem Çöz. ^{son test}	22,21	22	21	2,22282	17	28

Araştırma kapsamında kullanılan bilgisayarca düşünme becerileri ölçeğinde beşli likert yapısına verdikleri cevaplara göre oluşan düzeyleri ve öğrencilerin ölçekten aldıkları puanlar şu şekilde yorumlanmıştır: 20-51: Düşük Düzey; 52-67: Orta Düzey; 68-100: Yüksek Düzey.

Tablo 15

Cinsiyete Göre BDBOT ve BDBST Puanlarının Analizi

Puan Aralığı	Kategorisi	BDBOT			BDBST		
		Kız	Erkek	Toplam	Kız	Erkek	Toplam
20-51	Düşük	2	1	3	1	1	2
52-67	Orta	14	12	26	8	14	22
68-100	Yüksek	48	61	109	55	59	114
Toplam		64	74	138	1	74	138

Tablo 15 incelendiğinde, çalışma grubunun bilgisayarca düşünme becerilerinin cinsiyete göre ön test ve son test betimsel sonuçlarının düzeyleri analiz edilmiştir. Yüksek düzeydeki öğrenci sayısına bakıldığında ön testte 109, son testte 114 olduğu hesaplanmıştır. Orta ve düşük seviyedeki öğrenci sayısının ise azaldığı gözlemlenmektedir.

Cinsiyete göre ise, kızlarda yüksek düzeyde ön testte 48 öğrenci, son testte 55 öğrenci ile artma bulunmaktadır. Erkek öğrencilerde ise ön testte 61 öğrenci, son testte 59 öğrenci ile azalma görülmektedir. Bununla birlikte, kız öğrencilerde düşük ve orta seviyede azalma olduğu söylenebilir. Fakat erkek öğrencilerde düşük seviyede değişim yokken, orta seviyede artış olduğu görülmektedir. Bu bağımsız değişkenin, bağımlı değişkenlere etkisi “Beşinci Alt Probleme İlişkin Bulgular” bölümünde verilecektir.

4.2.1.2 Metin Temelli Programlama Başarı Testi Betimsel İstatistikler. Metin temelli programlama başarı testi puanlarına ilişkin betimsel istatistikler aşağıdaki Tablo 16 ve Tablo 17’de verilmiştir.

Tablo 16

Metin Temelli Programlama Başarı Testinin Cinsiyete Göre Ortalama Puanlarına İlişkin Betimsel İstatistikler

MTPBT	Kız n=64	Erkek n=74	Toplam n=138
\bar{X} (100 üzerinden)	80,25	78,04	79,07
Medyan	81	81	81
Mod	86	86	86
Std. Sapma	10,18	12,28	11,36
Min	57	48	48
Max	100	100	100

Tablo 16’ya göre, metin temelli programlama başarı testi puanlarına ilişkin betimsel istatistikler yer almaktadır. Çalışma grubunun toplam ortalamaları ele alındığında 100 puan üzerinden ortalama puan 79,07, medyan (ortanca) puan 81 ve mod (tepe değer) puan 86 olarak görülmektedir. Tabloya göre MTPBT’nin standart sapma katsayısı 11,36, en düşük öğrenci puanı 48 ve en yüksek öğrenci puanı 100

olarak hesaplanmıştır. Bu bulguya göre standart sapmanın oldukça büyük olduğu, en düşük ve en yüksek puanların arasındaki farkın fazla olduğu gözlemlenmektedir. Bulgulara göre öğrencilerin ortalamaları incelendiğinde 100 puan üzerinden ortalama puan 79,07, medyan (ortanca) puan 81 ve mod (tepe değer) puan 86 olarak belirlenmiştir.

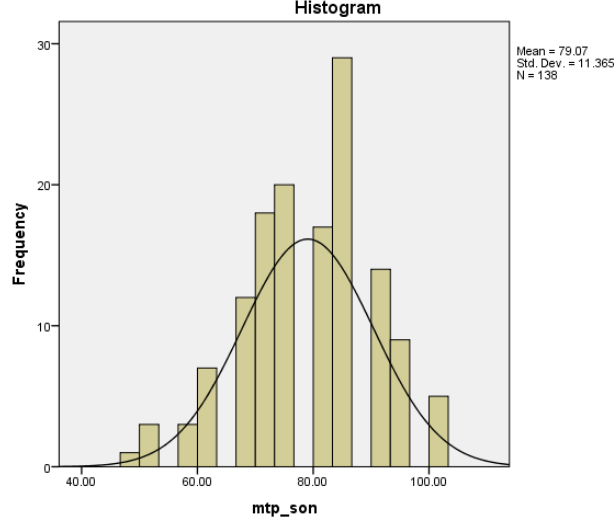
Tablo 17

Metin Temelli Programlama Başarı Testi Puanlarının Analizi

	Puan Aralığı	Kategorisi	Öğrenci Sayısı (n)	Yüzde (%)
MTPBT	54-45	BAŞARISIZ	4	2,9
	69-55	ORTA	22	15,9
	84-70	İYİ	55	39,9
	100-85	ÇOK İYİ	57	41,3
Toplam			138	100

Metin temelli programlama öğretimi sonrasında yapılan başarı testinin sonuçları öğrencilerin aldıkları puanlara göre başarı seviyeleri yukarıdaki tabloda gösterilmiştir. Buradan yola çıkılarak toplam puan üzerinden 54 ve altı alan öğrencilerin seviyesinin “başarısız”, 55-69 puan arasında puan alan öğrencilerin seviyesinin “orta”, 70-84 puan arasında puan alan öğrencilerin seviyesinin “iyi” ve 85 puan üstü alan öğrencilerin seviyesinin “çok iyi” olduğu söylenebilmektedir.

Tablo 17’ye göre toplam öğrenci sayısına göre başarısız (%2,9), orta seviyede (%15,9), iyi seviyede (%39,9) ve başarılı (%41,3) oranında öğrenci vardır. Buradan yola çıkarak istatistiksel olarak öğrencilerin yarısından fazlasının başarılı ve iyi düzeyinde oldukları görülmektedir.



Şekil 8.9. MTPBT'nin Betimsel Histogram Grafiği.

Şekil 7.9'de metin temelli programlama başarı testinin histogram grafiği görülmektedir. Buna göre dağılım üzerinde ortalama ve medyan farklı noktalarsa ise dağılımı betimlemek için çarpıklık katsayısı incelenir. Çarpıklık katsayısı ortalamanın medyan arasındaki farkının 3 katı alınır ve standart sapma katsayısına bölünerek elde edilir. Çarpıklık katsayısı negative ise sola çarpıklık, pozitif ise sağa çarpıklık söz konusudur (Yılmaz, 2018). Dolayısıyla çalışma grubundaki öğrencilerin akademik başarı dağılım katsayı değeri -0,51 olarak hesaplanmıştır. Buna göre dağılım sola çarpıktır.

Mod, medyan ve ortalama arasında değerlerin ilişkisi de dağılımın yönünü belirler. Eğer mod, medyan ve ortalama birbirine eşitse, simetrik dağılım; mod, medyan ve ortalamadan büyükse sola çarpık; ortalama ise mod ve medyandan büyükse sağa çarpıklık vardır (Civelek, 2014). Dağılımın yönüne göre testin başarısı hakkında yorum yapılabilir. Dağılımda sağa çarpıklık var ise öğrencilerin başarısı düşüktür, öğrencilerin yarısından fazlası ortalamanın altında kalmıştır ve öğretim yetersizdir. Dağılımda sola çarpıklık var ise öğrencilerin başarısı yüksektir, öğrencilerin yarısından fazlası ortalamanın üstündedir ve öğretim hedefi gerçekleşmiştir (Yılmaz, 2018). Mod, medyan ve ortalamanın ilişkisi incelendiğinde dağılımın sola çarpık olduğu görülmektedir ($Mod=86 > Medyan=81 > \bar{X}=79,07$). Buna göre öğrencilerin başarısı yüksektir ve öğretim hedefi gerçekleşmiştir.

4.2.1.3 Blok Temelli Programlamaya İlişkin Öz-Yeterlilik Alguları Ölçeği Betimsel İstatistikler. Blok temelli programlama öz-yeterlilik alguları puanlarına ilişkin betimsel istatistikler aşağıdaki Tablo 18 ve Tablo 19’da verilmiştir.

Blok Temelli Programlamaya İlişkin Öz-Yeterlilik Algısı Ölçeği’nde bulunan demografik sorularının veri dağılımı, Tablo 18 incelendiğinde görülmektedir. Demografik sorularda dört adet soru bulunmaktadır. Bu sorular aşağıdaki gibidir:

- Evinizde bilgisayar var mı?
- Evinizde internet var mı?
- Evde Scratch programı ile uygulama yapmaya devam ediyor musunuz?
- Scratch uygulaması ile programlama yapmayı seviyor musunuz?

Tablo 18

Öğrencilerin Blok Temelli Öz-Yeterlilik Algısı Anket Sorularının Dağılımı

Sorular	Yanıt	Sayı (n)	Yüzde (%)
1. Evinizde bilgisayar var mı?	Evet	129	93,5
	Var, ama bozuk	4	2,9
	Hayır	5	3,6
	Toplam	138	100
2. Evinizde internet var mı?	Evet	138	100
	Hayır	-	-
	Toplam	138	100
3. Evde Scratch programı ile uygulama yapmaya devam ediyor musunuz?	Evet	25	18,1
	Hayır	113	81,9
	Toplam	138	100
4. Scratch uygulaması ile programlama yapmayı seviyor musunuz?	Evet	60	43,5
	Kısmen	42	30,4
	Hayır	36	26,1
	Toplam	138	100

“Evinizde bilgisayar var mı?” sorusuna “Evet” yanıtı veren öğrenciler toplamda %93,5, “Var, ama bozuk” yanıtı veren öğrenciler toplamda %2,9 ve “Hayır” yanıtı veren öğrenciler toplamda %3,6 olarak görülmektedir. 1. soru yanıt analizlerine göre bilgisayar kullanımını evde de devam ettiren öğrenci sayısı oldukça fazladır, öğrencilerin bilgisayar kullanım becerileri hazırbulunuşluk olarak düşünüldüğünde bu öğrencilerin öğretim programına uygun oldukları ve blok temelli programlama yapmak için kullanabilecek bir bilgisayarı olduğunun oranının da yüksek olduğu gözlemlenmektedir.

“Evinizde internet var mı?” sorusuna “Evet” yanıtı veren öğrenciler toplamda %100’dür. Bu bağlamda, 2. soruya göre öğrencilerin tamamı internet evde de kullanabilmektedir. Blok temelli programlama araçlarında web tabanlı olanları kullanmak için gerekli alt yapı bulunmaktadır.

“Evde Scratch programı ile uygulama yapmaya devam ediyor musunuz?” sorusuna “Evet” yanıtı veren öğrenciler toplamda %18,1, “Hayır” yanıtı veren öğrenciler toplamda %81,6 olarak görülmektedir. Bu durumda öğrencilerin evde yeterli olanakları olmasına rağmen blok temelli programlama aracı olan Scratch’i kullananların düşük oranda olduğu ortaya çıkmıştır.

“Scratch uygulaması ile programlama yapmayı seviyor musunuz?” sorusuna “Evet” yanıtı veren öğrenciler toplamda %43,5, “Kısmen” yanıtı veren öğrenciler toplamda %30,4 ve “Hayır” yanıtı veren öğrenciler toplamda %26,1 olarak görülmektedir. 4. sorudaki “Evet” ve “Kısmen” yanıtı veren öğrencilerin sayılarına göre Scratch programını kullanmayı genel olarak yüksek oranda sevmektedirler.

Bulgulara göre 3. ve 4. sorunun yanıtlarına göre öğrenciler Scratch ile programlama yapmayı sevmelerine rağmen evde uygulamayı tercih etmemektedirler. Elde edilen verilerde öğrenciler blok temelli programlamaya devam eden ve etmeyen olarak gruplara ayrılmaktadır. Bu bağımsız değişkenin öğrencilerin bağımlı değişkenlere etkisi “Dördüncü Alt Probleme İlişkin Bulgular” bölümünde verilecektir.

Tablo 19

BT POT, BBTPOT, KBTPOT Cinsiyete Göre Ortalama Puanlarına İlişkin Betimsel İstatistikler

BT POT, BBTPOT, KBTPOT		Kız	Erkek
		n=64	n=74
Blok Temelli Programlama Öz-Yeterlilik Algısı	\bar{X} (60 üzerinden)	40,58	43,15
	Medyan	40,5	44
	Mod	40	40
	Std. Sapma	1,118	1,414
	Min	15	12
	Max	59	60
Basit Blok Temelli Programlama Öz-Yeterlilik Algısı	\bar{X} (25 üzerinden)	18,34	19,28
	Medyan	18,5	21
	Std. Sapma	,525	,646
	Min	7	5
	Max	25	25
	Karmaşık Blok Temelli Programlama Öz-Yeterlilik Algısı	\bar{X} (35 üzerinden)	22,05
Medyan		21,5	25
Std. Sapma		,693	,838
Min		7	7
Max		35	35

Tablo 19'a göre, blok temelli programlamaya ilişkin öz-yeterlilik algılarının ve alt boyutlarının puanlarına yönelik betimsel istatistikler yer almaktadır. Çalışma grubunun blok temelli programlama öz-yeterlilik algılarının toplam ortalamaları ele alındığında 60 puan üzerinden kızların ortalama puanı 40,58, medyan puanı 40,5 ve mod (tepe değer) puan 40 olduğu; erkeklerin ortalama puanı 43,15, medyan puanı 44 ve mod (tepe değer) puan 40 olduğu hesaplanmıştır. Blok temelli programlamaya ilişkin öz-yeterlilik algılarının birbirine oldukça yakın dağılımda olduğu görülmektedir. Bununla birlikte blok temelli programlamaya ilişkin öz-yeterlilik algıların incelendiğinde ortalama puanlarında fark olduğu ve erkeklerin algılarının daha yüksek olduğu belirlenmiştir ($\bar{X}=43,15>40,58$).

Ölçeğin alt boyutlarından basit blok temelli programlamaya ilişkin öz-yeterlilik algıları ortalamaları ele alındığında 25 puan üzerinden kızların ortalama puanı 18,34

ve medyan puanı 18,5 olduğu; erkeklerin ortalama puanı 19,28 ve medyan puanı 21 olduğu görülmektedir. Basit blok temelli programlamaya ilişkin öz-yeterlilik algılarının incelendiğinde ortalama puanlarının oldukça yakın olduğu fakat erkeklerin algılarının daha yüksek olduğu belirlenmiştir ($\bar{X}=19,28>18,34$).

Karmaşık blok temelli programlamaya ilişkin öz-yeterlilik algıları ortalamaları ele alındığında 35 puan üzerinden kızların ortalama puanı 22,05 ve medyan puanı 21,5 olduğu; erkeklerin ortalama puanı 24,67 ve medyan puanı 25 olduğu görülmektedir. Karmaşık blok temelli programlamaya ilişkin öz-yeterlilik algılarının incelendiğinde ortalama puanlarında fark olduğu ve erkeklerin algılarının daha yüksek olduğu belirlenmiştir ($\bar{X}=24,67>22,05$).

4.2.2 İkinci Alt Probleme İlişkin Bulgular. Araştırmanın ikinci alt problemi olan “Metin temelli programlama öğretiminde öğrencilerin akademik başarıları cinsiyete göre anlamlı farklılık göstermekte midir?” sorusuna yanıt aramak amacıyla Metin Temelli Programlama Başarı Testi (MTPBT) uygulanmıştır. Bu kapsamda öğrencilerin cinsiyete göre akademik başarılarının farkının belirlenmesi için İlişkisiz Örneklemeler t-Testi yapılmıştır. İlişkisiz Örneklemeler t-Testi’ne ilişkin bulgular Tablo 20’de yer almaktadır.

Tablo 20

MTPBT Puanlarının Cinsiyete Göre t-Testi Sonuçları

MTPBT	n	\bar{X}	S	sd	t	p
Kız	64	80,25	10,173	136	1,14	,256
Erkek	74	78,06	12,28			

Tablo 20’ye göre, öğrencilerin akademik başarılarının cinsiyete göre sonuçlarının ortalamaları istatistiksel olarak incelendiğinde anlamlı farklılık göstermediği bulunmuştur ($p=0,256>0,05$). MTPBT kız öğrencilerin puan ortalaması ($\bar{X}=80,25$), erkek öğrencilerin puan ortalaması ($\bar{X}=78,06$) istatistiksel olarak iyi düzeydedir. Öğrencilerin metin temelli programlama öğretimi sonrasında kızların erkeklere oranla puanlarında bir artış olduğu gözlenmektedir [$t(136)= 1,14$].

4.2.3 Üçüncü Alt Probleme İlişkin Bulgular. Araştırmanın üçüncü alt problemi olan “Metin temelli programlama öğretiminde, öğrencilerin bilgisayarca düşünme becerilerine göre ön test ve son test puanları arasında anlamlı farklılık var mıdır?” sorusuna yanıt aramak amacıyla öğrencilere yapılan Bilgisayarca Düşünme Beceri Düzeyleri Ölçeği ön test (BDBOT), son test (BDBST) uygulanmıştır. Çalışma grubuna uygulanan ölçeklerin ve testin betimsel verilerin incelemeleri aşağıdaki tablolarda gösterilmektedir. BDBOT ve BDBST aralarındaki farkı belirlemek için kullanılacak analiz yöntemi olarak İlişkili Örneklemeler t-Testi kullanılmıştır. İlişkili Örneklemeler t-Testi’ne ilişkin sonuçlar Tablo 21’de verilmiştir.

Tablo 21

BDBST Ve BDBOT Puanlarının t-Testi Sonuçları

BDBOT	n	\bar{X}	S	sd	t	p
Son test	138	76,34	10,047	137	3,41	,001
Ön test	138	73,25	8,669			

Tablo 21’e göre, bilgisayarca düşünme becerileri ön test ve son test sonuçlarının ortalamaları istatistiksel olarak incelendiğinde anlamlı bir farklılık bulunmuştur ($p^{bdbot}=0,001<0,05$). BDBST puan ortalaması ($\bar{X}=76,34$), BDBOT puan ortalamasına ($\bar{X}=73,25$) göre istatistiksel olarak yükselmiştir. Öğrencilerin metin temelli programlama öğretimi sonrasında bilgisayarca düşünme becerileri puanlarında bir artış olduğu gözlenmektedir [$t(137)=3,41$]. Test sonucunda hesaplanan etki büyüklüğü ise ($\eta^2=0,29$) olarak hesaplanmıştır, bu farkın orta düzeyde bir etki olarak değerlendirilir. Çünkü etki büyüklüğü 0,5 ve 0,2 arasında ise orta etki olarak yorumlandırılır (Green ve Salkind, 2005).

4.2.3.1 Üçüncü Alt Problemin Alt Boyutlarına İlişkin Bulgular. Araştırmanın üçüncü alt problemin boyutu olan “Metin temelli programlama öğretiminde, öğrencilerin bilgisayarca düşünme becerileri alt boyutlarına göre ön test ve son test puanları arasında anlamlı farklılık var mıdır?” sorusuna yanıt aramak amacıyla öğrencilere BDBOT ve BDBST ölçekleri uygulanmıştır.

Bilgisayarca Düşünme Beceri Düzeyleri Ölçeği’nde bulunan beş ayrı faktör yaratıcılık, algoritmik düşünme, işbirliklilik, eleştirel düşünme ve problem çözme

ortalama puanları ön test ve son test aralarındaki farkı belirlemek için kullanılacak analiz yöntemi olarak İlişkili Örneklem t-Testi kullanılmıştır. İlişkili Örneklem t-Testi'ne ilişkin veriler Tablo 22'de yer almaktadır.

Tablo 22

BDBST Ve BDBOT Puanlarının Beceri Düzeylerine Göre t-Testi Sonuçları

BDBST	n	\bar{X}	S	sd	t	p
BDBOT						
Yaratıcılık	138	16,29	2,77215	137	,822	,412
		16,03	2,94708			
Algoritmik Düş.	138	14,46	3,63511	137	,819	,414
		14,15	3,82062			
İşbirliklilik	138	17,22	3,12050	137	,076	,939
		17,20	3,21983			
Eleştirel Düş.	138	14,01	3,73186	137	,995	,322
		13,61	4,15298			
Problem Çöz.	138	18,56	2,23708	137	13,919	,000
		22,21	2,22282			

Tablo 22'ye göre, bilgisayarca düşünme beceri düzeyleri ön test ve son test sonuçlarının ortalamaları istatistiksel olarak incelenmiştir. Bilgisayarca düşünmenin beceri düzeylerinden “Yaratıcılık”, “Algoritmik Düşünme”, “İşbirliklilik” ve “Eleştirel Düşünme” alanlarında anlamlı bir farklılık bulunmamıştır ($p^{yaratıcılık}=0,412>0,05$, $p^{algoritmik}=0,414>0,05$, $p^{işbirliklilik}=0,939>0,05$, $p^{eleştirel}=0,322>0,05$).

Metin temelli programlama öğretimi sonrasında “Problem Çözme” beceri düzeyinde anlamlı bir farklılık bulunmuştur ($p^{problem}=0,00<0,05$). BDBST^{problem} puan ortalaması ($\bar{X}=22,21$), BDBOT^{problem} puan ortalamasına ($\bar{X}=18,56$) göre istatistiksel olarak yükselmiştir. Öğrencilerin problem çözme becerilerinde anlamlı bir artış olduğu gözlenmektedir [$t(137)^{problem}=13,91$]. Test sonucunda hesaplanan etki büyüklüğü ise ($\eta^2=1,18$) olarak hesaplanmıştır, bu fark büyük düzeyde bir etki olarak değerlendirilir. Çünkü, etki büyüklüğü 0,8'den yüksek ise büyük etki olarak yorumlandırılır (Green ve Salkind, 2005).

4.2.4 Dördüncü Alt Probleme İlişkin Bulgular. Araştırmanın dördüncü alt problemi olan “Ortaokul öğrencilerinin, blok temelli programlama öz-yeterlilik algıları, bilgisayarca düşünme becerileri ve akademik başarıları, blok temelli programlama aracını kullanmaya devam etme durumuna göre anlamlı farklılık göstermekte midir?” sorusuna yanıt aranmıştır.

Öğrencilerin BT POT ölçeğindeki birinci bölümde yer alan demografik sorulara verdikleri yanıtlar doğrultusunda öğrenciler blok temelli programlama aracını kullanmaya devam etme durumlarına göre iki kategori altında toplanmışlardır. Bilgisayarca düşünme becerileri, blok temelli programlama öz-yeterlilik algıları ve akademik başarıları bağımlı değişkenleri üzerinde öğrencilerin blok temelli programlama aracını kullanmaya devam etmeleri bağımsız değişkenlerinin, anlamlı etkisi olup olmadığını belirlemek amacıyla çok yönlü varyans (MANOVA) analizi yapılmıştır. Birden fazla bağımlı değişken bakımından anlamlı farklılık gösterip göstermediğini test etmek amacıyla MANOVA testi kullanılır (Büyüköztürk, 2019).

Bağımlı değişkene ilişkin puanlar tek değişkenli ya da çok değişkenli normal dağılım gösterebilir. Bağımlı değişkenlerin her biri, bağımsız değişkenin her düzeyinde normal dağılır ya da bağımlı değişkenler bağımsız değişkenlerin düzeylerinde çok değişkenli normal dağılır. Normal dağılım durumunu, z istatistiği kullanılarak incelenir. Bağımlı değişkenlerin normal dağılım ve z istatistikleri Tablo 8, Tablo 9 ve Tablo 11’de verilmiştir.

Bağımlı değişkenler arasında doğrusal bir ilişki mevcuttur. Bu varsayımlar ikili kombinasyonlarda saçılma grafikleri bakılarak incelenir. Bağımlı değişkene ilişkin puanların varyans ve kovaryans matrisleri homojen olmalıdır. Bağımlı değişkenin tüm ikili kombinasyonlarında kovaryansların eşit olduğu varsayılır (Büyüköztürk, 2019). Varyans-kovaryans matrislerinin homojenliği testi sonucunda elde edilen bulgulara göre kovaryans matrislerinin homojenliği belirlenmiştir. Tablo 23’te varyans-kovaryans matrislerinin homojenliği testi ile elde edilen değerler verilmiştir.

Tablo 23

Varyans-Kovaryans Matrisinin Homojenliği Testi

	Box’ M	Sd1	Sd2	F	p
Değer	2,663	6	10785,398	,423	,864

Levene F testiyle bağımlı deęişkenlerin her biri için hata varyanslarının homojenlięi test edilmiştir. Elde edilen bulguların p deęeri 0,05'ten büyük olduęu durumlarda homojen olarak belirtilmiştir. Levene F testi ile hata varyanslarının homojenlięi bağımlı deęişkenlerde deęişkenlik gösterdięi görülmektedir. BTPOT, MTPBT ve BDBST'nin p deęerleri 0,05'ten büyük olduęu için hata varyanslarını homojendir. Levene F testiyle hata varyanslarının homojenlięi testi ile elde edilen deęerler Tablo 24'te verilmiştir. Buna göre bağımlı deęişkenlerin hata varyanslarının homojenlięi bulunmuştur.

Tablo 24

Levene F Testi Sonuçları

	F	Sd1	Sd2	p
BTPOT	,401	1	136	,527
MTPBT	,094	1	136	,759
BDBST	,309	1	136	,579

Çoklu doğrusal bağlantı ve tekillik, bağımlı deęişkenlerin arasındaki korelasyonla ilgilidir. Buna göre, bağımlı deęişkenler arasındaki korelasyon katsayısı 0,9'un altında olmalıdır (Can, 2019). Bağımlı deęişkenlerin arasındaki ilişkiyi belirlemek için Pearson Korelasyon testi yapılmıştır. Korelasyon katsayısı, iki veya daha çok deęişken arasındaki ilişkiyi yorumlamak amacıyla kullanılır. Pearson Korelasyon Katsayısı, iki deęişkenin normal dağılım gösterdięi sürekli ve ikili olması durumunda kullanılır (Büyüköztürk, 2019). Bağımlı deęişkenler anlamlı düzeydedir ve korelasyon katsayıları 0,9 deęerinin altındadır. Buna ilişkin deęerler Tablo 25'te verilmiştir.

Tablo 25

BT POT, MTPBT ve BDBST Sonuçlarının Korelasyon Analizi

Değişken		n	r	p
BT POT	MTPBT	138	,215*	,011*
	BDBST	138	,513	,000
MTPBT	BT POT	138	,215*	,011*
	BDBST	138	,371	,000

* Aynı veriler mevcuttur.

Çok yönlü varyans (MANOVA) analizini uygulamak için tek değişkenli ve çok değişkenli normal dağılım sergilemesi gerekmektedir. Normallik testlerine göre BT POT, MTPBT ve BDBST bağımlı değişkenleri tek değişkenli normal dağılım göstermektedir. Tek değişkenli normal dağılımların kontrolünden sonra çok değişkenli normal dağılım testi yapılırken, Mahalanobis uzaklıklarının hesaplanması yapılmaktadır (Can, 2019).

Araştırma kapsamında hesaplanan Mahalanobis değişkeni ve aldığı değerler incelenmiştir. Mahalanobis kritik değeri ile çok değişkenli normalliği bozan uç değerlerin olup olmadığı belirlenmektedir. Kritik değer, Büyüköztürk (2005), Field (2005), Barnett ve Levis (1978) aktardığı örneklem ve yordayıcı değişken sayılarına göre örneklem sayısı 100 ve değişken sayısı 3 olduğu durumlarda, 15 ve üzeri uç değerler olarak adlandırılır (akt. Can, 2019). Çalışmada hesaplanan Mahalanobis kritik değerlerinde 15 ve üzeri değer bulunmamaktadır. Bu durumda çok değişkenli normalliğin sağlandığı karar alınmıştır.

MANOVA testinin sonuçları Tablo 26'da verilmiştir. Tablo 26 incelendiğinde, Scratch kullanmaya devam etme durumlarının göre yönelik blok temelli programlama öz-yeterlilik algıları, bilgisayarca düşünme becerileri ve akademik başarıları üzerinde anlamlı etkisinin olup olmadığı görülmektedir. Bu durumda Scratch'e devam etme durumlarının blok temelli programlama öz-yeterlilik algıları, bilgisayarca düşünme becerileri ve akademik başarıları üzerinde anlamlı etkisi olduğu görülmektedir (Wilks' $\Lambda=0,887$, $F_{(3-134)}=5,713$, $p=0,001<0,05$). Ayrıca bağımsız değişkenin bağımlı değişken üzerindeki etki büyüklüğü (eta kare) değerleri de görülmektedir. Etki büyüklüğü hesaplandıktan sonra yorumlarken Cohen's f kullanılır. Cohen's f değeri 0,10 küçük, 0,25 orta ve 0,40 yüksek kabul edilir (Cohen, 1988). Tablo 26'de etki büyüklükleri incelendiğinde Scratch'e devam etme durumu değişkeninin blok temelli

programlamaya ilişkin öz-yeterlilik algıları, bilgisayarca düşünme becerileri ve akademik başarıları üzerinde orta dereceli etki düzeyine sahip olduğu görülmektedir ($\eta^2 = 0,113$, Cohen's $f = 0,36$).

Tablo 26

BTPOT, MTPBT ve BDBST Puanlarının, Blok Temelli Programlama Aracını Kullanmaya Devam Etme (Scratch'e Devam Etme) Ve Aracı Sevme (Scratch Sevme) Değişkenliğine Göre MANOVA Testi Sonuçları

Değişken	Wilks' Λ	F	Hipotez Sd	Hata Sd	p	η^2
Scratch'e Devam Etme Durumu	,887	5,713	3	134	,001	,113

Tablo 27'de blok temelli programlamaya ilişkin öz-yeterlilik algıları, bilgisayarca düşünme becerileri ve akademik başarıları bağımlı değişkenlerinin, Scratch'e devam etme durumlarının bağımsız değişkenlerine göre birbiri üzerindeki etkileri gösterilmektedir. MANOVA testine göre, Scratch kullanmaya devam etme durumlarının bilgisayarca düşünme becerileri ve blok temelli programlama öz-yeterlilik algıları üzerinde anlamlı bir etkisi bulunmuştur ($p < 0,05$).

Tablo 27

BTPOT, MTPBT ve BDBST Bağımlı Değişkenlerinin, Scratch'e Devam Etme Bağımsız Değişkenine Göre Birbirleri Üzerindeki Etkileri

Bağımsız Değişken	Bağımlı Değişkenler	Kareler Toplamı	Kareler Ortalaması	F	p	η^2
Düzeltilmiş Model	BTPOT	1709,809	1786,396	19,301	,000	,222
	MTPBT	407,816	224,089	1,754	,177	,025
	BDBST	513,081	381,531	3,941	,022	,055
Ortak Etkileşim	BTPOT	164861,809	39793,431	429,943	,000	,761
	MTPBT	530475,178	181641,345	1421,686	,000	,913
	BDBST	497375,805	155837,043	1609,895	,000	,923

Tablo 27'nin Devamı

Bağımsız Değişken	Bağımlı Değişkenler	Kareler Toplamı	Kareler Ortalaması	F	p	η²
Scratch'e	BTPOT	1709,809	493,567	5,333	,022	,038
Devam Etme Durumu	MTPBT	407,816	445,432	3,486	,064	,025
	BDBST	513,081	227,961	2,355	,127	,017
Hata	BTPOT	14357,930	92,555			
	MTPBT	17288,598	127,765			
	BDBST	13317,912	96,799			
Toplam	BTPOT	258996				
	MTPBT	880377				
	BDBST	818079				
Düzeltilmiş Toplam	BTPOT	16067,739				
	MTPBT	17696,413				
	BDBST	13830,993				

Bu araştırma sorusu kapsamında Scratch'i kullanmaya devam etme durumlarının blok temelli programlamaya ilişkin öz-yeterlilik algıları üzerinde ortaya çıkan anlamlı etkisinin hangi alt boyutlardan kaynaklandığını araştırmak için ikinci bir MANOVA testi yapılmıştır.

Levene testi sonucuna göre Scratch'e devam eden ve etmeyen öğrencilerin durumuna göre basit blok temelli programlama öz-yeterlilik algı puanlarında ($F_{(1-136)}=0,017$, $p=0,896>0,05$) ve karmaşık blok temelli programlama öz-yeterlilik algılarında ($F_{(1-136)}=2,97$, $p=0,088>0,05$) homojen dağılım görülmemektedir. Tablo 28'de görüldüğü üzere Scratch'e devam etme durumunun blok temelli programlama öz-yeterlilik algılarının birleşik alt boyutları üzerinde anlamlı bir farklılık oluşturduğu bulunmuştur (Wilks' $\Lambda=0,899$, $F_{(2-135)}=7,635$, $p=0,001<0,05$).

Tablo 28

BT POT Alt Boyutuna Ait Düzey Puanlarının, Blok Temelli Programlama Aracını Kullanmaya Devam Etme (Scratch'e Devam Etme) Değişkenliğine Göre MANOVA Testi Sonuçları

Değişken	Wilks' Λ	F	Hipotez Sd	Hata Sd	p	η^2
Scratch'e Devam Etme Durumu	,899	7,635	2	135	,001	,101

Tablo 29'a göre Scratch'e devam etme durumunun blok temelli programlama öz-yeterlilik algıları alt boyutları basit ve karmaşık blok temelli programlama öz-yeterlilik algısı (BBTPOT ve KBTPOT) bağımlı değişkenleri üzerindeki etkileri gösterilmektedir. Scratch'e devam etme durumlarının, basit ve karmaşık blok temelli programlama öz-yeterlilik algısı alt boyutlarında değişkenler tek ele alındığında anlamlı etkiye sahip olduğu görülmektedir ($p < 0,05$). Basit blok temelli programlama öz-yeterlilik algısı değeri 0,00 ve karmaşık blok temelli programlama öz-yeterlilik algısı değeri 0,001 p değerinden küçük olarak bulunmuştur.

Scratch programlama aracını kullanmaya devam etmeye yönelik durumlarının basit blok temelli programlama öz-yeterlilik algıları üzerindeki etki büyüklüğü incelendiğinde ($\eta^2=0,9$, Cohen's $f=3$) yüksek düzeyde etkiye ve karmaşık blok temelli programlama öz-yeterlilik algıları üzerindeki etki büyüklüğü incelendiğinde ($\eta^2=0,82$, Cohen's $f=2,13$) yüksek düzeyde etkiye sahiptir.

Tablo 29

KBTPOT Ve BBTPOT Bağımlı Değişkenlerinin, Scratch'e Devam Etme Bağımsız Değişkenine Göre Birbirleri Üzerindeki Etkileri

Bağımsız Değişken	Bağımlı Değişkenler	Kareler Toplamı	Kareler Ortalaması	F	p	η²
Düzeltilmiş Model	KBTPOT	534,338	534,338	13,373	,000	,090
	BBTPOT	280,704	280,704	12,271	,001	,083
Ortak Etkileşim	KBTPOT	51528,135	51528,135	1289,649	,000	,905
	BBTPOT	32847,023	32847,023	1435,889	,000	,913
Scratch'e Devam Etme Durumu	KBTPOT	534,338	534,338	13,373	,000	,090
	BBTPOT	280,704	280,704	12,271	,001	,083
Hata	KBTPOT	5433,902	39,955			
	BBTPOT	3111,100	22,876			
Toplam	KBTPOT	81897,000				
	BBTPOT	52415,000				
Düzeltilmiş Toplam	KBTPOT	5968,239				
	BBTPOT	3391,804				

Tablo 30

KBTPOT ve BBTPOT Bağımlı Değişkenlerinin, Blok Temelli Programlama Aracını Kullanmaya Devam Etme (Scratch'e Devam Etme) Değişkenine Göre Grup Dağılım Sonuçları

Bağımlı Değişken	Scratch'e Devam Etme Durumları	Kareler Farkı	p
KBTPOT	Evet	5,109	,000
BBTPOT	Hayır	3,703	,001

Tablo 30'a göre Scratch'e devam edenler ve etmeyenler arasında bağımlı değişkenlere göre anlamlı değişim gözlemlenmektedir. Buna göre Scratch'e devam eden grubun basit ve karmaşık blok temelli programlama öz-yeterlilik algıları Scratch'e devam etmeyenlere göre daha yüksektir.

Bu araştırma kapsamında Scratch'i kullanmaya devam etme durumlarının bilgisayarca düşünme becerileri üzerinde ortaya çıkan anlamlı etkisinin hangi alt boyutlardan kaynaklandığını araştırmak için üçüncü bir MANOVA testi yapılmıştır. MANOVA sayıltılarından biri olan varyans-kovaryans matrislerinin eşitliği söz konusudur. Yapılan Box Matrislerin Eşitliği Testi sonucunda p değeri 0,05 değerinden küçük olduğu için eşitlik sağlanamamıştır ($p=0,001$). Eğer bu eşitlik sağlanmadıysa araştırmada Pillai's Trace kullanılmalıdır (Büyüköztürk, 2019). Levene testi sonucuna göre Scratch'e devam eden ve etmeyen öğrencilerin durumuna göre bilgisayarca düşünme becerileri altı boyutlarında yaratıcılık puanlarında ($F_{(1-136)}=0,315$, $p=0,575>0,05$), algoritmik düşünme puanlarında ($F_{(1-136)}=0,748$, $p=0,389>0,05$), eleştirel düşünme puanlarında ($F_{(1-136)}=2,174$, $p=0,143>0,05$) ve problem çözme puanlarında ($F_{(1-136)}=1,092$, $p=0,298>0,05$) homojen dağılım görülmektedir. Ancak işbirliklilik puanlarında ($F_{(1-136)}=5,56$, $p=0,02<0,05$) homojen dağılım görülmektedir.

Tablo 31

BDBST Alt Boyutuna Ait Düzey Puanlarının, Blok Temelli Programlama Aracını Kullanmaya Devam Etme (Scratch'e Devam Etme) Değişkenliğine Göre MANOVA Testi Sonuçları

Değişken	Pillai's Trace	F	Hipotez Sd	Hata Sd	p	η^2
Scratch'e Devam Etme Durumu	,070	2,001	5	132	,083	,070

Tablo 31'de görüldüğü üzere Scratch'e devam etme durumunun bilgisayarca düşünme becerileri birleşik alt boyutları üzerinde anlamlı bir farklılık oluşturmadığı bulunmuştur (Pillai's Trace=0,07, $F_{(5-132)}=2,001$, $p=0,083>0,05$).

Tablo 32

BDBST Alt Boyutları Bağımlı Değişkenlerinin, Scratch'e Devam Etme Bağımsız Değişkenine Göre Birbirleri Üzerindeki Etkileri

Bağımsız Değişken	Bağımlı Değişkenler	Kareler Toplamı	Kareler Ortalaması	F	p	η^2
Düzeltilmiş Model	Yaratıcılık	1,924	1,924	,220	,640	,002
	Algoritmik Düş.	36,129	36,129	2,502	,116	,018
	İşbirliklilik	52,964	52,964	5,268	,023	,037
	Eleştirel Düş.	17,233	17,233	,999	,319	,007
	Problem Çöz.	2,931	2,931	,591	,443	,004
Ortak Etkileşim	Yaratıcılık	21295,692	21295,692	2437,972	,000	,947
	Algoritmik Düş.	17396,477	17396,477	1204,843	,000	,899
	İşbirliklilik	25699,138	25699,138	2556,090	,000	,949
	Eleştirel Düş.	15823,639	15823,639	917,455	,000	,871
	Problem Çöz.	40832,670	40832,670	8239,545	,000	,984
Scratch'e Devam Etme Düzeyi	Yaratıcılık	1,924	1,924	,220	,640	,002
	Algoritmik Düş.	36,129	36,129	2,502	,116	,018
	İşbirliklilik	52,964	52,964	5,268	,023	,037
	Eleştirel Düş.	17,233	17,233	,999	,319	,007
	Problem Çöz.	2,931	2,931	,591	,443	,004
Hata	Yaratıcılık	1187,960	8,735			
	Algoritmik Düş.	1963,675	14,439			
	İşbirliklilik	1367,355	10,054			
	Eleştirel Düş.	2345,636	17,247			
	Problem Çöz.	673,975	4,956			
Toplam	Yaratıcılık	36646,000				
	Algoritmik Düş.	29639,000				
	İşbirliklilik	42260,000				
	Eleştirel Düş.	27920,000				
	Problem Çöz.	68751,000				

Tablo 32'nin Devamı

Bağımsız Değişken	Bağımlı Değişkenler	Kareler Toplamı	Kareler Ortalaması	F	p	η^2
Düzeltilmiş Toplam	Yaratıcılık	1189,884				
	Algoritmik Düş.	1999,804				
	İşbirliklilik	1420,319				
	Eleştirel Düş.	2362,870				
	Problem Çöz.	676,906				

Tablo 32'ye göre Scratch'e devam etme durumunun bilgisayarca düşünme becerileri alt boyutları yaratıcılık, algoritmik düşünme, işbirliklilik, eleştirel düşünme ve problem çözme bağımlı değişkenleri üzerindeki etkileri gösterilmektedir. Scratch'e devam etme durumlarının bilgisayarca düşünme becerileri alt boyutlarında değişkenler tek ele alındığında işbirliklilik becerisinin anlamlı etkiye sahip olduğu görülmektedir ($p=0,023<0,05$).

Tablo 33

BDBST Alt Boyutları Bağımlı Değişkenlerinin, Blok Temelli Programlama Aracını Kullanmaya Devam Etme (Scratch'e Devam Etme) Değişkenine Göre Grup Dağılım Sonuçları

Scratch'e Devam Etme Durumları		Bağımlı Değişken	Kareler Farkı	p
Evet	Hayır	Yaratıcılık	0,307	0,64
		Algoritmik Düşünme	1,328	0,116
		İşbirliklilik	1,608	0,023
		Eleştirel Düşünme	0,918	0,319
		Problem Çözme	0,378	0,443

Tablo 33'e göre Scratch'e devam edenler ve etmeyenler arasında bağımlı değişkenlere göre anlamlı değişimler gözlemlenmektedir. Buna göre işbirliklilik bağımlı değişkeninde Scratch'e devam eden grubun becerisi devam etmeyenlere göre

daha yüksektir.

4.2.5 Beşinci Alt Probleme İlişkin Bulgular. Araştırmanın beşinci alt problemi olan “Ortaokul öğrencilerinin, blok temelli programlama öz-yeterlilik algıları, bilgisayarca düşünme becerileri ve akademik başarıları, ön düşünme durumuna göre anlamlı farklılık göstermekte midir?” sorusuna yanıt aranmıştır.

Öğrencilerin BDBOT ölçeğindeki sorulara verdikleri yanıtlar doğrultusunda öğrenciler düşünme beceri durumları kapsamında düşük, orta ve yüksek olarak üç kategori altında toplanmışlardır. Bu kategoriler, ön test uygulamasından elde edilen veriler olduğundan dolayı araştırma kapsamında ön düşünme durumları olarak belirtilecektir.

Bilgisayarca düşünme becerileri, blok temelli programlama öz-yeterlilik algıları ve akademik başarıları bağımlı değişkenleri üzerinde öğrencilerin ön düşünme durumu bağımsız değişkeninin, anlamlı etkisi olup olmadığını belirlemek amacıyla çok yönlü varyans (MANOVA) analizi yapılmıştır. MANOVA testi için bağımlı ve bağımsız değişkenlerin belirli koşulları sağlaması gerekmektedir. Bu sebepten dolayı MANOVA testini uygulamak için bir örneklem grubunun en az 15 katılımcıdan oluşması gerekmektedir (Can, 2019). Sonuç olarak araştırmaya katılan düşünme durumu düşük olan 5 öğrenci bu testte değerlendirme dışında tutulmuştur.

Bağımlı değişkenlerin z istatistikleri incelenerek bağımsız değişkenler üzerinde tek ve çok değişkenli normal dağılımları Tablo 8, Tablo 9 ve Tablo 11’de görülmektedir. Çoklu doğrusal bağlantı ve tekillik için bağımlı değişkenlere ait bilgiler Tablo 26’da verilmiştir. Bu durumda bağımlı değişkenler anlamlı düzeydedir ve korelasyon katsayıları 0,9 değerinin altındadır. Varyans-kovaryans matrislerinin homojenliği testi sonucunda elde edilen bulgulara göre kovaryans matrislerinin homojenliği belirlenmiştir. Tablo 34’de varyans-kovaryans matrislerinin homojenliği testi ile elde edilen değerler aşağıdaki gibidir.

Tablo 34

Varyans-Kovaryans Matrisinin Homojenliği Testi

	Box’ M	Sd1	Sd2	F	p
Değer	6,098	6	9891,002	,966	,446

Levene F testiyle bağımlı değişkenlerin her biri için hata varyanslarının homojenliği test edilmiştir. Elde edilen bulguların p değeri 0,05'ten büyük olduğu durumlarda homojen olarak belirtilmiştir. Levene F testi ile hata varyanslarının homojenliği bağımlı değişkenlerde değişkenlik gösterdiği görülmektedir. BTPOT, MTPBT ve BDBST'nin p değerleri 0,05'ten büyük olduğu için hata varyanslarını homojendir. Tablo 35'de Levene F testiyle hata varyanslarının homojenliği testi ile elde edilen değerler aşağıdaki gibidir. Buna göre bağımlı değişkenlerin hata varyanslarının homojenliği bulunmuştur.

Tablo 35

Levene F Testi Sonuçları

	F	Sd1	Sd2	p
BTPOT	,969	1	131	,327
MTPBT	,009	1	131	,923
BDBST	2,454	1	131	,120

Mahalanobis kritik değeri ile çok değişkenli normalliği bozan uç değerlerin olup olmadığı kontrol edilmiştir. Buna göre hesaplanan Mahalanobis kritik değerlerinde 15 ve üzeri değer bulunmamaktadır. Bu durumda çok değişkenli normalliğin sağlandığı karar alınmıştır.

Tablo 36

BTPOT, MTPBT ve BDBST Puanlarının, Ön Düşünme Durumu Değişkenliğine Göre MANOVA Testi Sonuçları

Değişken	Wilks' Λ	F	Hipotez Sd	Hata Sd	p	η^2
Ön Düşünme Durumu	,908	4,374	3	129	,006	,092

MANOVA testinin sonuçları Tablo 36'da verilmiştir. Tablo 36 incelendiğinde ön düşünme durumlarına göre blok temelli programlama öz-yeterlilik algıları, bilgisayarca düşünme becerileri ve akademik başarıları üzerinde anlamlı etkisinin olup

olmadığı görülmektedir.

Buna göre ön düşünme durumu bağımsız değişkeninin, blok temelli programlamaya ilişkin öz-yeterlilik algıları, bilgisayarca düşünme becerileri ve akademik başarıları birleşik bağımlı değişkenler üzerinde anlamlı etkisi olduğu görülmektedir (Wilks' $\Lambda=0,908$, $F_{(3-129)}=4,374$, $p=0,006<0,05$). Ayrıca bağımsız değişkenin bağımlı değişken üzerindeki etki büyüklüğü (eta kare) değerleri de görülmektedir. Tablo 36'ya göre etki büyüklükleri ön düşünme durumu değişkeninin blok temelli programlamaya ilişkin öz-yeterlilik algıları, bilgisayarca düşünme becerileri ve akademik başarıları üzerinde orta dereceli birleşik etki düzeyine sahip olduğu görülmektedir ($\eta^2 = 0,092$, Cohen's $f = 0,318$).

Tablo 37

BTPOT, MTPBT ve BDBST Bağımlı Değişkenlerinin, Ön Düşünme Durumu Bağımsız Değişkenlerine Göre Ortalama Ve Standart Sapma Değerleri

Bağımlı Değişkenler	Bağımsız Değişken		
	Ön Düşünme Durumu Düzeyleri	Ortalama	Std. Sapma
BTPOT (60 üzerinden)	Orta	39,042	11,28092
	Yüksek	43,165	9,99585
MTPOT (100 üzerinden)	Orta	80,667	11,27778
	Yüksek	78,706	11,48695
BDBST (100 üzerinden)	Orta	71,958	11,44924
	Yüksek	77,927	8,69835

Tablo 37'de ön düşünme durumlarına ait BTPOT, MTPBT ve BDBST ortalama puanları ve standart sapmalarının sonuçları verilmiştir. Ön düşünme durumu yüksek olan öğrencilerin blok temelli programlama öz-yeterlilik algıları ve bilgisayarca düşünme becerileri puanları, orta olan öğrencilerin puanlarından yüksektir. Ancak ön düşünme durumu yüksek olan öğrencilerin akademik başarı puanları, orta olan öğrencilerin puanlarından düşüktür.

Tablo 38

BT POT, MTPBT ve BDBST Bağımlı Değişkenlerinin, Ön Düşünme Durumu Bağımsız Değişkenlerine Göre Birbirleri Üzerindeki Etkileri

Bağımsız Değişken	Bağımlı Değişkenler	Kareler Toplamı	Kareler Ortalaması	F	p	η^2
Düzeltilmiş Model	BT POT	334,435	334,435	3,194	,076	,024
	MTPBT	75,580	75,580	,576	,449	,004
	BDBST	700,621	700,621	8,205	,005	,059
Ortak Etkileşim	BT POT	132923,458	132923,458	1269,353	,000	,906
	MTPBT	499592,693	499592,693	3810,368	,000	,967
	BDBST	441877,704	441877,704	5174,688	,000	,975
Ön Düşünme Durumu	BT POT	334,435	334,435	3,194	,076	,024
	MTPBT	75,580	75,580	,576	,449	,004
	BDBST	700,621	700,621	8,205	,005	,059
Hata	BT POT	13717,986	104,717			
	MTPBT	17175,939	131,114			
	BDBST	11186,371	85,392			
Toplam	BT POT	253392,000				
	MTPBT	848569,000				
	BDBST	797367,000				
Düzeltilmiş Toplam	BT POT	14052,421				
	MTPBT	17251,519				
	BDBST	11886,992				

Tablo 38’de blok temelli programlamaya ilişkin öz-yeterlilik algıları, bilgisayarca düşünme becerileri ve akademik başarıları bağımlı değişkenlerinin, ön düşünme durumu bağımsız değişkenine göre birbiri üzerindeki etkileri gösterilmektedir. MANOVA testine göre, ön düşünme durumunun, blok temelli programlama öz-yeterlilik algısı ve akademik başarıları üzerinde anlamlı bir etkisi bulunmazken ($p>0,05$), bilgisayarca düşünme becerileri üzerinde anlamlı bir etkisi bulunmuştur ($p<0,05$).

Bu araştırma sorusu kapsamında ön düşünme durumunun, bilgisayarca düşünme becerileri üzerinde ortaya çıkan anlamlı etkisinin hangi alt boyutlardan kaynaklandığını araştırmak için ikinci MANOVA testi yapılmıştır. Ön düşünme durumunun bilgisayarca düşünme becerileri alt boyutlarına ilişkin MANOVA testi sonuçları aşağıdaki gibidir. MANOVA sayıltılarından biri olan varyans-kovaryans matrislerinin eşitliği söz konusudur. Yapılan Box Matrislerin Eşitliği Testi sonucunda p değeri 0,05 değerinden küçük olduğu için eşitlik sağlanamamıştır ($p=0,039$). Eğer bu eşitlik sağlanmadıysa araştırmada Pillai's Trace kullanılır.

Tablo 39'da görüldüğü üzere ön düşünme düzeyi bağımsız değişkeninin; bilgisayarca düşünme becerisi bağımlı değişkeninin birleşik alt boyutları olan yaratıcılık, algoritmik düşünme, işbirliklilik, eleştirel düşünme ve problem çözme becerileri üzerinde anlamlı bir farklılık oluşturduğu bulunmuştur (Pillai's Trace=0,163, $F_{(5-127)}=4,978$, $p=0,00<0,05$).

Tablo 39

BDBST Alt Boyutuna Ait Düzey Puanlarının, Ön Düşünme Durumu Değişkenliğine Göre MANOVA Testi Sonuçları

Değişken	Pillai's Trace	F	Hipotez Sd	Hata Sd	p	η^2
Ön Düşünme Durumu	,163	4,948	5	127	,000	,163

Tablo 40'ta ön düşünme durumu bağımsız değişkeninin, yaratıcılık, algoritmik düşünme, işbirliklilik, eleştirel düşünme ve problem çözme olmak üzere bilgisayarca düşünme becerisi alt boyutları olan bağımlı değişkenleri üzerindeki etkileri gösterilmektedir. Tablo 39'a göre ön düşünme durumlarının yaratıcılık, algoritmik düşünme, eleştirel düşünme ve problem çözme değişkenleri tek ele alındığında hepsinde anlamlı etkiye sahip olduğu ($p<0,05$), ancak işbirliklilik değişkeninde anlamlı bir etkiye sahip olmadığı görülmektedir ($p>0,05$).

Tablo 40

BDBST Alt Boyutu Bağımlı Değişkenlerinin, Ön Düşünme Durumu Bağımsız Değişkenlerine Göre Birbirleri Üzerindeki Etkileri

Bağımsız Değişken	Bağımlı Değişkenler	Kareler Toplamı	Kareler Ortalaması	F	p	η^2
Düzeltilmiş Model	Yaratıcılık	61,241	61,241	7,758	,006	,056
	Algoritmik Düş.	92,238	92,238	6,907	,010	,050
	İşbirliklilik	23,427	23,427	2,661	,105	,020
	Eleştirel Düş.	93,414	93,414	6,011	,016	,044
	Problem Çöz.	80,400	80,400	18,088	,000	,121
Ortak Etkileşim	Yaratıcılık	19021,602	19021,602	2409,737	,000	,948
	Algoritmik Düş.	14426,253	14426,253	1080,194	,000	,892
	İşbirliklilik	22971,818	22971,818	2609,349	,000	,952
	Eleştirel Düş.	13394,677	13394,677	861,883	,000	,868
	Problem Çöz.	41129,603	41129,603	9253,300	,000	,986
Ön Düşünme Durumu	Yaratıcılık	61,241	61,241	7,758	,006	,056
	Algoritmik Düş.	92,238	92,238	6,907	,010	,050
	İşbirliklilik	23,427	23,427	2,661	,105	,020
	Eleştirel Düş.	93,414	93,414	6,011	,016	,044
	Problem Çöz.	80,400	80,400	18,088	,000	,121
Hata	Yaratıcılık	1034,067	7,894			
	Algoritmik Düş.	1749,536	13,355			
	İşbirliklilik	1153,279	8,804			
	Eleştirel Düş.	2035,894	15,541			
	Problem Çöz.	582,276	4,445			
Toplam	Yaratıcılık	35625,000				
	Algoritmik Düş.	28785,000				
	İşbirliklilik	41611,000				
	Eleştirel Düş.	27254,000				
	Problem Çöz.	66317,000				

Tablo 40'ın Devamı

Bağımsız Değişken	Bağımlı Değişkenler	Kareler Toplamı	Kareler Ortalaması	F	p	η^2
Düzeltilmiş Toplam	Yaratıcılık	1095,308				
	Algoritmik Düş.	1841,774				
	İşbirliklilik	1176,707				
	Eleştirel Düş.	2129,308				
	Problem Çöz.	662,677				

Tablo 40'a göre ön düşünme durumuna yönelik yaratıcılık üzerindeki etki büyüklüğü incelendiğinde ($\eta^2=0,56$, Cohen's $f=0,24$) orta düzeyde etkiye, algoritmik düşünme üzerindeki etki büyüklüğü incelendiğinde ($\eta^2=0,05$, Cohen's $f=0,22$) orta düzeyde etkiye, eleştirel düşünme üzerindeki etki büyüklüğü incelendiğinde ($\eta^2=0,044$, Cohen's $f=0,21$) orta düzeyde etkiye ve problem çözme üzerindeki etki büyüklüğü incelendiğinde ($\eta^2=0,67$, Cohen's $f=0,37$) yüksek düzeyde etkiye sahiptir.

Tablo 41

BDBST Alt Boyutları Bağımlı Değişkenlerinin, Ön Düşünme Durumu Değişkenine Göre Grup Dağılım Sonuçları

Bağımlı Değişken	Ön Düşünme Durumları		Kareler Farkı	p
Yaratıcılık			1,765	0,006
Algoritmik Düşünme			2,166	0,01
İşbirliklilik	Yüksek	Orta	1,091	0,105
Eleştirel Düşünme			2,179	0,016
Problem Çözme			2,022	0,00

Tablo 41'e göre ön düşünme durumu yüksek ve orta olanlar arasında bağımlı değişkenlere göre anlamlı değişimler gözlemlenmektedir. Buna göre yaratıcılık, algoritmik ve eleştirel düşünme, problem çözme bağımlı değişkenlerinde ön düşünmesi yüksek olan grubun becerisi orta olanlara göre daha yüksektir.

4.2.6 Altıncı Alt Probleme İlişkin Bulgular. Araştırmanın altıncı alt problemi olan “Ortaokul öğrencilerinin, tasarlanan ders etkinliklerine, bilgisayarca düşünme becerilerine ve blok temelli programlama öz-yeterlilik algılarına yönelik betimsel görüşleri nelerdir?” şeklinde oluşturulmuştur. Bu araştırma probleminin bulguları öğrenciler ile yapılan açık uçlu anket sorularına yönelik görüşmelerdeki notlar ve kağıtlardan oluşmaktadır. Öğrencilerden elde edilen cevaplarda, gizlilik esas alınarak öğrencilere “O1, O2, O3, O4, O5, O6” gibi kodlar verilmiştir.

4.6.1 Öğrenci Görüşmelerine Ait Bulgular. Bu alt probleme ilişkin bulgular için çalışma grubunda bulunan 6. sınıf öğrencilerinden rastgele seçilen ve gönüllü olarak katılan 6 öğrenci ile gerçekleştirilmiştir. Bu kategorilerin hangi temalarla sınıflandırıldığı Tablo 42 ile gösterilmiştir.

Tablo 42

Öğrenci Görüşmelerine Dair Tema ve Kategoriler

Kategori	Tema
Ders içi uygulama ve etkinliklere dair genel	Motivasyona yönelik Zorluk hissine yönelik Metin komutlarına yönelik Devam etmeye yönelik
Bilgisayarca düşünme beceri düzeylerine yönelik	Bireysel farklılığa yönelik Problem çözmeye yönelik Algoritmik düşünmeye yönelik
Akademik başarıya yönelik	Bilgiye yönelik Uygulamaya yönelik
Blok temelli programlama öz-yeterlilik algılarına yönelik	Özgüvende hissetmeye yönelik Hatırlamaya yönelik Zorluk algısına yönelik

Bu öğrenciler akademik olarak başarılı, iyi ve çok iyi düzeyinde bulunan öğrencilerdir. Yapılan görüşmelerden toplanan verilere göre metin temelli programlama öğretimine dair görüşmeler “Ders içi uygulama ve etkinliklere dair genel”, “Bilgisayarca düşünme beceri düzeylerine yönelik”, “Akademik başarıya yönelik”, “Blok temelli öz-yeterlilik algılarına yönelik” dört kategoriye ayrılmıştır.

4.6.1.1 Ders ii uygulama ve etkinliklere dair genel grüşme. Uygulama iin metin temelli programlama ğretiminde ders ii uygulama ve etkinlikleri sonrasında ğrencilerin genel grüşleri eđlenceli, ğretici, motive edici ve verimli bir sre geirdikleri ynündedir. ğrencilerin bazılarına ait grüşler řu řekildedir:

O1: Kodlama etkinlikleri ok iyiydi. Metin temelli programlamada en ok hořuma giden řey kodları yazmaktı. Evet, beni gayet motive etti. Metin temelli programlamaya devam ederim.

O2: ok iyiydi ve ok fazla řey ğrendim. Metin temelli programlama en ok dřünme becerimi geliřtirmesi hořuma gitti. Metin temelli programlamada zor řeyler vardı fakat ğretmenim ve arkadaşlarım sayesinde her řeyi bařardım. Beni motive etti, eskisinden daha iyi olduđumu dřünmeye bařladım, devam etmek isterim.

O3: Kodlama etkinlikleri eđlenceli ve ğreticiydi. Yazılı kodlama etkinlikleri ile ilgili en ok hořuma giden řey kodların daha kapsamlı olmasıydı, zorlandıđım kısım olmadı. Evet, devam etmek isterim.

O5: Güzeldi, en ok hořuma giden HTML'di. Zor ve sıkıcı gelen bir konu olmadı. Beni motive etti. Metin temelli programlamaya devam ederim.

4.6.1.2 Bilgisayarca dřünme beceri dzeylerine ynelik grüşme. ğrencilerin metin temelli programlama ğretimi sresince bilgisayarca dřünme beceri dzeylerine katkısı olup olmadığına ynelik dřünceleri deđerlendirildiđinde, srecin becerilerine katkısı olduđuna ynelik cevaplar vermiřlerdir. ğrencilerin bazılarına ait grüşler řu řekildedir:

O5: Algoritmik dřünme, problem özme ve yaratıcılık becerime etkisi oldu.

O4: Problem özme, algoritmik dřünme ve iřbirlikli alıřmama katkısı oldu, ünkü anlamayınca bir başkasına sorduđumuz zaman daha iyi anlamakla kalmayarak, birlikte daha iyi dřünüyoruz. Ancak eleřtirel dřünmeme katkısı olduđunu dřünmüyorum.

O2: Algoritmik dřünme ve problem özme becerime katkısı oldu, en ok etkilediđi unsurlardan biri.

4.6.1.3 Akademik başarıya yönelik görüşme. Öğrencilerin metin temelli programlama öğretimi sonrasında akademik başarılarına ve yeni kavramlar öğrendiklerinin bilincinde olup olmadıklarına ilişkin görüşleri değerlendirildiğinde sürecin programlamadaki başarılarını olumlu etkilediğini ve yeni bilgiler öğrendiklerinin farkında oldukları sonucuna ulaşıldı. Öğrencilerin bazılarına ait görüşler şu şekildedir:

O1: Bir sürü yeni bilgi öğrendim, akademik başarıma katkısı oldu.

O2: Metin temelli etkinliklerde yeni bir sürü bilgi öğrendim, önceki öğrendiklerimi daha sonra da kullandım. BTY dersindeki başarıma çok büyük bir katkısı oldu.

O6: Kesinlikle çok şey öğrendim, başlık ekleme gibi temel kodları hep kullandık. Evet, başarıma katkısı oldu.

4.6.1.4 Blok temelli öz-yeterlilik algılarına yönelik görüşme. Öğrencilerin blok temelli öz-yeterlilik algılarını, metin temelli programlama öğretimi süresince kullanıp kullanmadıklarına yönelik tutumları ve görüşlerine göre blok temelli programlamada bazı kavramları hatırladıkları ve blok temelli programlama bilmeden metin temelli programlamada zorlanacaklarına yöneliktir. Öğrencilerin bazılarına ait görüşler şu şekildedir:

O2: Evet, blok temelli programlamada temel kodları hatırladım. Kesinlikle düşünüyorum, blok temelli programlama bilmezken metin temelli yapmak çok zor olurdu.

O1: Bazı kavramları kullandım. Evet düşünüyorum bana çok fazla katkısı oldu.

O4: Scratch programında öğrenmiş olduklarımı metin temelli programlamada kullandım. Kesinlikle bana katkısı olduğunu düşünüyorum.

Bölüm 5

Tartışma ve Sonuçlar

Bu bölümde araştırma kapsamında elde edilen bulgulardan ulaşılan sonuçlara ilişkin tartışmalara ve sonuçlar doğrultusunda geliştirilen önerilere ve yorumlara yer verilmiştir. Çalışmada ortaokul öğrencilerinin metin temelli programlama öğretimi kapsamındaki etkinliklerin bilgisayarca düşünme becerilerine ve akademik başarılarına etkisi araştırılmıştır, öğrencilerin bu uygulamalar hakkındaki görüşleri incelenmiştir. Araştırma kapsamında elde edilen bulgular, asıl amaca ulaşılması doğrultusunda alt amaçlarla sunulmuş ve literatür kapsamında tartışılmıştır. Tartışma, araştırma soruları ile ilişkilendirilerek önce bilgisayarca düşünme becerisi ve akademik başarı üzerindeki etkisi, sonrasında blok temelli öz-yeterlilik algı düzeylerinin bilgisayarca düşünme becerileri ve akademik başarı arasındaki ilişki, öğrencilerin sürece yönelik ifadeleri şeklinde olmuştur.

5.1 Araştırmanın Birinci Alt Problemine Dair Tartışma ve Sonuçlar

Araştırmanın birinci alt problemi olan “Ortaokul öğrencilerinin blok temelli programlama öz-yeterlilik algıları, bilgisayarca düşünme becerileri ve akademik başarıları ne düzeydedir?” olarak ifade edilmiştir.

Uygulama sonrasında elde edilen bulgulara göre bilgisayarca düşünme beceri düzeyleri olan düşük, orta ve yüksek olarak öğrencilerin sayısı ön test ve son test istatistiksel olarak incelenmiştir. BDBST’e sonuçlarına göre yüksek düzeydeki öğrencilerin sayısının BDBOT’e oranla yükseldiği hesaplanmıştır. Orta ve düşük seviyedeki öğrenci sayısının oranlarının ise azaldığı görülmektedir. Buradan yola çıkarak bilgisayarca düşünme becerileri bütünüyle ele alındığında öğrencilerin becerilerinin gelişim gösterdikleri görülmüştür. Bu sonuca paralel olarak alanyazın çalışmalarında bilgisayarca düşünme becerisinin gelişmesine, bilgisayar bilimleri ve programlama öğretiminin yardımcı olduğu belirtilmiştir (Barr ve Stephenson, 2011). Ancak bilgisayarca düşünmenin alt boyutlarından sadece problem çözme becerisinin puanında bir artış olduğu bulgulara dayalı olarak söylenebilmektedir. Kukul (2018)

bilgisayarca düşünmenin üst düzey düşünme biçimi olarak ele alınması gerektiğini belirtmiştir. Bu becerinin gelişmesi için uzun süreli uygulamalar yapılmalıdır; sadece programlama ile tüm becerilerin gelişmesi beklenilmemelidir (Kukul, 2018). Bilgisayarca düşünme aşamasında problem çözme becerisi ise diğer becerilerinde gelişimine katkı sağlar (Korkmaz vd., 2015). Bu sebeple bulgulara göre yaratıcılık, algoritmik düşünme, işbirliklilik ve eleştirel düşünme becerilerinin genel ortalama puanları oldukça yüksektir. Metin temelli programlama öğretimi sonrasında ise problem çözme becerisinde oldukça yükselme görülmektedir. Bu da programlama öğretiminin bilgisayarca düşünmeye katkı sağladığını göstermektedir. Aynı zamanda bilgisayarca düşünmenin bilgisayar bilimlerinde problem çözme yaklaşımı olduğuna dair savı destekler niteliktedir (ISTE, 2019; SLO, 2020). Buna göre Papert'ın (1980) belirttiği gibi bilgisayarca düşünme programlama ile ilgilidir.

Öğrencilerin akademik başarılarını belirlemek için hazırlanan MTPBT istatistiksel sonuçlarına göre öğrencilerin seviyesi 4 kategoride incelenmiştir. Mevcut verilere göre başarısız %2,9, orta seviyede %15,9, iyi seviyede %39,9 ve başarılı %41,3 oranında öğrenci vardır. Buradan yola çıkarak istatistiksel olarak öğrencilerin genel olarak başarılı ve iyi düzeyinde oldukları görülmektedir. Metin temelli programlama öğretimi sonucunda öğretim hedefinin çoğunlukta sağlandığını sonucuna ulaşılabilmektedir.

Uygulamada blok temelli programlamaya ilişkin öz-yeterlilik algılarını ve alt boyutlarının etkilerini araştırmak için BT POT ölçeği kullanılmıştır. Öğrencilerin ortalama puanları istatistiksel olarak incelendiğinde, ortalama puanlarının genel olarak yüksek olduğu ve bulunan değerlerin kız ve erkek öğrencilerde değişkenlik gösterdiği belirlenmiştir. Kız öğrencilerin ortalama puanı $\bar{X}=40,58$ iken, erkek öğrencilerin ortalama puanı $\bar{X}=43,15$ olduğu görülmektedir. Bu durumda erkek öğrencilerin blok temelli programlama öz-yeterlilik algıları istatistiksel olarak daha yüksek olduğu görülmektedir. Öğrencilerin blok temelli programlama aracı hakkındaki sonuçları incelendiğinde aracı sevdikleri sonucuna ulaşılmıştır. Buna göre öğrenciler blok temelli programlama aracına karşı olumlu bir tavır sergilemektedir. Bu durumda Powers ve arkadaşları (2007) blok temelli programlama araçlarının özgüven ve motivasyon sağladığı savını desteklemektedir. Meerbaum (2011), programlama öğretiminde acemi olanlar için motivasyonlarının yüksek tutulmasında ve programlama kavramına alışmaları açısından blok temelli programlama araçları iyi bir

seçim aracı olabilmektedir. Dolayısıyla blok temelli programlama araçları, metin temelli programlama öğretimi öncesinde hedef odaklı çalışmalarla kullanılabilir.

5.2 Araştırmanın İkinci Alt Problemine Dair Tartışma ve Sonuçlar

Araştırmanın ikinci alt problemi olan “Metin temelli programlama öğretiminde öğrencilerin akademik başarıları cinsiyete göre anlamlı farklılık göstermekte midir?” olarak ifade edilmiştir.

Programlama öğretimi sonrasında elde edilen bulgulara göre cinsiyetin akademik başarıları üzerinde anlamlı etkisinin olup olmadığı incelenmiştir. Araştırmaya göre cinsiyetin akademik başarıları üzerinde anlamlı etkisinin olmadığı saptanmıştır. Verilere göre kız öğrencilerin puan ortalaması $\bar{X}=80,25$ iken erkek öğrencilerin puan ortalaması $\bar{X}=78,06$ olarak bulunmuştur. Alanyazında cinsiyet ve programlama öğretimi hakkındaki çalışmalar incelendiğinde kız öğrencilerin akademik başarı olarak daha yüksek sonuçlar aldığı, erkeklerin ise programlamaya yönelik algı ve motivasyonlarının daha yüksek olduğu görülmüştür (Romero-Zaliz, Manoso ve Madrid, 2015). Her ne kadar cinsiyette eşitlik tüm mesleklerde yayılmış olsa dahi, kadınların bilişim sektöründeki sayısının az olması dikkat çekmektedir (The Tech Partnership ve BC, 2016; Vu, 2017). Bu durumun bilgisayar biliminde öncü olan kadınlardan sonra olması ise sıkça araştırılan bir konu haline gelmesine neden olmuştur (Öigaard, 2015). Carter ve Jenkins (1999) programlama öğretiminde seçilen konuların ve öğretim yaklaşımlarının erkek merkezli olduğundan dolayı motivasyonel olarak olumlu etkilendiklerini ancak konu seçimlerinin kızlara yönelik seçilmesi ile algılarının değişebileceğini belirtmişlerdir. Kız öğrencilerin sınıf dışı çalışmalara da katılarak programlamaya yönelik deneyim ve güvenlerinin arttığı gözlemlenmiştir. Genel olarak kızların hedef odaklı ve yenilikçi çalışmalarda erkeklere oranla daha dikkatli ve başarılı olmaları dikkat çekmektedir (Boyle ve Jenkins, 1996).

Kızların öz-yeterlilik algılarının rol model olarak hemcinslerinin olması ile kendilerine olan özgüvenlerinin arttığı ve matematik, fen bilimleri, teknoloji ve programlama gibi alanlarda başarılı olabileceği söylenmektedir (Aldağ ve Tekdal, 2015; Byrne ve Lyons, 2001; Zeldin ve Pajares, 2000). Bu durumda, akademik başarıları yüksek çıkan kız öğrencilerin, blok temelli programlama öz-yeterlilik algılarının düşük olmasının sebebi kendilerine programlama alanında güvenmemeleri olarak düşünülmektedir. Ancak ders içeriklerinde konulardan ve öğretim

yaklaşımlarından akademik başarılarını olumlu etkilenmiş olarak görülmektedir.

Brauner, Leonhardt, Ziefle ve Schroeder (2010) kızların öz-yeterliliğinin düşük olması sonucunda program geliştirme performansının düşebildiğini ifade etmişlerdir. Ancak bu sonuçlar karmaşık programlama görevlerinde öz-yeterliliklerinin düşük olduğunu varsaymıştır. Bu tez çalışmasında ise programlamaya ilişkin öz-yeterlilik algılarının, kızların metin temelli programlamadaki akademik başarılarını etkilemediğini ortaya çıkmıştır. Rubio, Romero-Zaliz, Manoso ve Madrid (2015) programlama dersinde kız ve erkek öğrencilerin arasındaki farklılığı azaltmak için geliştirdikleri öğretim modülünü uygulamışlardır. Bu modülün sonucunda ise kız ve erkek öğrencilerin programlama öz-yeterlilik algıları ve öğrenme çıktıları arasındaki fark ortadan kalkmıştır ve kızlar için programlama öz-yeterlilik algısının artması için aktif stratejiler uygulanmasını önermektedirler. Kızlara programlama kavramlarını gerçeklikle bağlantılar kurularak verilmesi sonucunda daha etkili olacağı önerilmektedir (Othman ve Latih, 2006; Öigaard, 2015).

5.3 Araştırmanın Üçüncü Alt Problemine ve Boyutuna Dair Tartışma ve Sonuçlar

Araştırmanın üçüncü alt problemi ve boyutu olan “Metin temelli programlama öğretiminde, öğrencilerin bilgisayarca düşünme becerilerine göre ön test ve son test puanları arasında anlamlı farklılık var mıdır?” ve “Metin temelli programlama öğretiminde, öğrencilerin bilgisayarca düşünme becerileri alt boyutlarına göre ön test ve son test puanları arasında anlamlı farklılık var mıdır?” olarak ifade edilmiştir.

Programlama öğretimi sonrasında elde edilen bulgulara göre bilgisayarca düşünme becerileri ön test ve son test ortalama puanları arasında anlamlı farklılık olduğu ortaya çıkmıştır. Mevcut ortalama puan verileri incelendiğinde öğrencilerin ön test ortalama puanı $\bar{X}=73,25$ iken son test puan ortalaması $\bar{X}=76,34$ anlamlı bir farklılık gözlemlenmiştir. Bu durum metin temelli programlama öğretimi sonrasında öğrencilerin bilgisayarca düşünme becerilerini olumlu yönde geliştirdiğini göstermektedir. Bu durum programlama öğretiminin, bilgisayarca düşünme kavramlarının kazandırılmasında önemli bir araç olduğunu ortaya koymuştur (Bocconi vd., 2016; Lu ve Fletcher, 2009). Wing (2006) bilgisayarca düşünmenin herkes için gerekli olduğunu savunmuştur. Ancak Denning (2017) bilgisayarca düşünme hareketinin içerdiği kavramlar henüz bir standarta tabi olmadığından dolayı herkes için

bilgisayar bilimi fikrine karşıdır. Buna rağmen 21. yüzyıl beceri setleri ve beceri odaklı müfredat çalışmalarında bilgisayarca düşünme becerilerine ve alt boyutlarına yönelik çalışmalar yapılarak gerekli müfredatlar hazırlanmaktadır (CSTA, 2009, 2017; ISTE, 2019; P21, 2015; SLO, 2019, 2020).

Bilgisayarca düşünme ve programlamanın “zihninde ne olduğu” sorusuna doğrudan cevap verebilecek şekilde çok az sayıda araştırmanın mevcut olduğu görülmektedir. Genellikle yapılan çalışmalar zihinsel süreçlerin çıktıklarına odaklanmaktadır, zihinsel süreçlerin kendilerini ise ihmal etmektedirler (Demir ve Seferoğlu, 2017). Birbiriyle ilişkili alanlar olsa da bilgisayar bilimi, programlama ve bilgisayarca düşünme farklı alanlardır (Denning, 2017; Wing, 2006). Bu yüzden dünyada ve ülkemizde bilgisayar bilimleri dersi programlarında güncellemeler yapılmıştır. BTY dersi müfredatlarına programlama ve problem çözme konusu alt kavramlarından hesaplamalı düşünme adı altında bilgisayarca düşünme kavramları verilmektedir. Bilgisayar bilimleri ve programlamanın temelinde ise bilgisayarca düşünme yer almaktadır (Demir ve Seferoğlu, 2017; Tekinarslan ve Çetin, 2018). CSTA, ISTE ve SLO gibi uluslararası kurumlar ve MEB müfredatlarında, blok temelli programlama öğretiminden sonra metin temelli programlama öğretimi verilmektedir. Müfredatların kazanım ve içeriklerine göre öğrencilerin bilgisayarca düşünme becerilerinin gelişmesi amaçlanmaktadır. Böylelikle metin temelli programlama dilleri ile bilgisayarca düşünme becerilerinin ilerlediği savını destekler niteliktedir (Kandemir, 2018b).

Programlama öğretimi sonrasında elde edilen bulgulara göre bilgisayarca düşünme beceri alt basamakları olan yaratıcılık, algoritmik düşünme, işbirliklilik ve eleştirel düşünme ön test ve son test ortalama puanları arasında anlamlı bir fark bulunamamıştır. Ancak problem çözme ön test ve son test ortalama puanları arasında anlamlı farklılık olduğu ortaya çıkmıştır. Mevcut problem çözme ortalama puan verileri incelendiğinde öğrencilerin ön test ortalama puanı $\bar{X}=18,56$ iken son test puan ortalaması $\bar{X}=22,21$ olmuştur. Bu durum metin temelli programlama öğretimi sonrasında problem çözme becerisi eğilimlerini olumlu yönde geliştirdiğini göstermektedir. Papert (1980), bilgisayarca düşünme, bilgisayar bilimi ve programlama ile ilişkili olduğunu söylemiştir. Öte yandan bu araştırmanın sonucu bilgisayarca düşünmenin, programlama becerisinin kendisi olmadığı savını da desteklemektedir (Wing, 2006). Ayrıca bilgisayarca düşünme, bilgisayar bilimini tarif

etmede yetersizdir (Denning, 2013, 2017). Bu sebeple programlama öğretimi, bilgisayarca düşünme sürecinin gelişimine katkıda bulunur; fakat programlamanın kendisi bilgisayarca düşünme değildir. Bundy (2007) tarafından yapılan çalışmalarda bilgisayarca düşünmenin problem çözme ve algoritmik becerileri ile her alanda kavramsal anlama için gerekli olduğu ifade edilmiştir. Czerkowski ve Lyman (2015) bilgisayarca düşünme ile bireylerin problem çözme becerilerinin ilerleyeceğini ve daha verimli bireyler haline geleceklerine dikkat çekmiştir. Ek olarak Wing (2006) ve Barr, Harrison ve Conery (2011) tarafından bilgisayarca düşünme becerisi, teknoloji ile düşünce birleşimini güçlendiren bir problem çözme yaklaşımı olarak özetlenmiştir, öğrencilerin bu becerileri eğitim-öğretim süreçleri içerisinde kazanmaları ve geliştirmeleri önemlidir.

Araştırma kapsamında öğrencilerin metin temelli programlama öğretimi öncesinde yaratıcılık, işbirliklilik, algoritmik düşünme ve eleştirel düşünme becerileri maksimum puana oldukça yakın ve yüksektir. Ancak problem çözme becerisi metin temelli programlama öğretimi sonrasında oldukça gelişmiştir. Bu durumda programlama öğretimi bilgisayarca düşünmenin alt boyutlarını etkilemektedir. Bilgisayarca düşünmenin alt basamaklarından yaratıcılık, algoritmik düşünme, işbirliklilik, eleştirel düşünme, problem çözme ve iletişim becerileri incelendiğinde öğrencilerin okul çağında bu becerilere sahip olmaları, hem onların kendilerini geliştirmelerini hem de teknoloji çağı öğrenme kültürüne sahip olacakları çeşitli araştırmalarda dile getirilmektedir (Barr vd., 2011; Brown, 2015; ISTE, 2019; Lye ve Koh, 2014). Brown (2015); günlük hayatta algoritmalar ve karmaşık problemlerle karşılaşıldığı düşünüldüğünde okullarda bu becerilerin geliştirilmesinin önem arz ettiğini vurgulamıştır.

Problem çözüm sürecinde bilgisayara adım adım ve mantıklı bir sırada anlatma işlemi, programlama sürecidir ve bu süreç oldukça karmaşık bir süreçtir (Helminen ve Malmi, 2010; Kert ve Uğraş, 2009). Ayrıca programlamanın problem çözme, eleştirel düşünme ve yaratıcılık gibi becerileri kapsamaması, programlama dillerinin karmaşık söz dizimi ve ara yüzlerinin kullanımının zor olması bu becerilerin kazanılamamasının sebeplerindedir. Ancak programlama dili öğretiminde gerekli beceriler arasında problem çözme becerisi önemli yer almaktadır. Saygıner ve Tüzün (2017) araştırmasında programlamanın bir problem çözme süreci olduğunu ifade etmiştir. Bu yüzden problem çözme becerisi programlama becerisini ve başarıyı etkileyen önemli

faktörlerden biridir. Programlama süreci bireyin, problem çözüme ve tasarlama algılarını geliştirmektedir.

Sonuç olarak bulgular bilgisayarca düşünmenin alt basamakları olan yaratıcılık, işbirlikli çalışma, algoritmik ve eleştirel düşünmenin yerini almadan problem çözüme becerisi ile tüm becerileri geliştirilir düşüncesini destekler niteliktedir (Korkmaz vd., 2015). Balanskat ve Engelhardt (2015) problem çözüme, algoritmik ve eleştirel düşünme kavramlarının programlama öğretimi ile olacağını belirtmiştir. Şendurur (2018) problem çözüme ve soyutlama yeteneklerinin metin temelli programlama dilleri ile arttığını söylemektedir.

5.4 Araştırmanın Dördüncü Alt Problemine Dair Tartışma ve Sonuçlar

Araştırmanın dördüncü alt problemi olan “Ortaokul öğrencilerinin, blok temelli programlama öz-yeterlilik algıları, bilgisayarca düşünme becerileri ve akademik başarıları, blok temelli programlama aracını kullanmaya devam etme durumuna göre anlamlı farklılık göstermekte midir?” olarak ifade edilmiştir.

Araştırma sonucunda elde edilen bulgulara göre blok temelli programlama öz-yeterlilik algıları, bilgisayarca düşünme becerileri ve akademik başarıları arasında anlamlı bir etki söz konusudur. Bu araştırma sorusuna yönelik öğrencilerin blok temelli programlama aracını kullanmaya devam etme durumlarının blok temelli programlama öz-yeterlilik algıları, bilgisayarca düşünme becerileri ve akademik başarıları üzerindeki etkisi incelenmiştir. Öğrencilerin blok temelli programlama aracını kullanmaya devam etme durumlarının bilgisayarca düşünme becerileri ve programlamaya ilişkin öz-yeterlilik algıları ve alt boyutları ile daha ilgili olduğu söylenebilmektedir.

Blok temelli programlama öğretiminin alt boyutları olan basit ve karmaşık blok temelli programlama öz-yeterlilik algıları üzerinde anlamlı bir etkiye sahip olduğu belirlenmiştir. Blok temelli programlama öz-yeterlilik algısına ait boyutların etki büyüklükleri ayrı olarak incelendiğinde aracı kullanmaya devam etme durumu yüksek düzeyde etki büyüklüğüne sahiptir. Blok temelli programlama aracını kullanmaya devam eden öğrencilerin programlama öz-yeterlilik algılarının daha yüksek olduğu ortaya çıkmıştır. Bu durumda blok temelli programlama aracı ile programlama yapmaya devam eden öğrenciler kendilerini daha öz güvenli hissetmektedir. Programlamaya yeni başlayan öğrencilerin blok temelli araçlar ile hedefe yönelik

problem çözmeleri ve hazır kod bloklarını kullanmaları, programlama becerilerini ve motivasyonlarını etkilemektedir (Maloney vd., 2008). Bu durumda, BT POT ölçeğinin metin temelli programlama öğretimi öncesinde uygulanmış olmasından dolayı blok temelli programlamaya devam eden öğrencilerin programlama kavramı konusunda öz-yeterlilik algıları dolayısıyla motivasyonları ve öz güvenleri oldukça yüksektir. Saygıner ve Tüzün (2017) blok temelli programlama araçlarının öğrencilerin üretken ve yaratıcı bireyler olmaya teşvik ettiğini söylemektedir. Bu araçların kod yazmaya teşvik etmesinden dolayı öğrencilerin programlama temel becerileri gelişmektedir. Bu nedenle genellikle blok temelli programlama araçları temel programlama kavramlarını öğretmek için kullanılmakla birlikte problem çözme becerilerinin ve motivasyonlarının gelişmesi için tercih edilebilmektedir (Kandemir 2018a; Kert 2018b). Programlamanın temel kavramları, sıralama, koşul, döngü, değişken gibi tüm geleneksel programlama dillerinde kullanılan kavramlardır (Saygıner ve Tüzün, 2017). Blok temelli programlamaya devam eden öğrencilerin bu temel kavramlara da hakim olması beklenmektedir. Elde edilen bulgulara göre blok temelli programlamaya devam eden öğrencilerin bu kapsamda öz-yeterlilikleri daha yüksek olarak bulunmuştur.

Araştırmada blok temelli programlama aracını kullanmaya devam etme durumlarının akademik başarıları üzerinde anlamlı bir etkisi olmadığı sonucu ortaya çıkmıştır. Powers ve arkadaşları (2007) metin temelli programlama öğretiminde blok temelli programlama dillerinin özgüven ve motivasyon sağladığını; ancak metin temelli programlamanın söz dizimsel kurallarında ve yazılımı derlemede zorlandıklarını belirtmişlerdir. Weintrop ve Wilensky (2015) blok temelli programlamanın arayüzü kullanımının farklı olması, metin temelli programlama dili olmaması ve sınırlı kod yazma gibi özelliklerinden dolayı metin temelli programlama öğrenen öğrencilerin başarılarını olumsuz etkileyebileceğini söylemektedir. Blok temelli programlama aracı kullanan öğrencilerin metin temelli programlama ile karşılaştıklarında zorluklar yaşadığı belirtilmiştir (Powers vd., 2007). Ancak Meerbaum ve arkadaşlarına (2011) göre programlamaya yeni başlayan acemilerin motivasyonlarını yüksek tutmaları için blok temelli programlama araçları iyi bir seçimdir. Programlama öğretiminde blok temelli programlama araçlarının, öğrencilerin blok temelli programlama akademik başarısını etkilemediği Erdem ve Kalelioğlu (2018) tarafından belirtilmiştir. Bu araştırmanın sonucu da blok temelli

programlamanın öğrencilerin metin temelli programlamadaki akademik başarılarında önemli bir etken olmadığını göstermektedir.

Dorling ve White (2015) programlama öğretimi için geliştirdikleri modelde blok temelli programlama aracında metin temelli programlama aracına ilk geçişte bu iki aracın birlikte kullanılacağı ortamlar ya da blok temelli programlama aracını kavramları öğretmek için değil metin temelli programlama diline destekleyici olarak kullanılmaktadır. Bu çalışma grubunda bulunan öğrenciler 5. sınıf düzeyinde iken blok temelli programlama aracı olan Scratch aracını kullandıktan sonra eğitsel programlama aracı olan Code Monkey ile hedefe yönelik kodlama yapmışlardır. 2018-2019 eğitim-öğretim birinci yarısında ise eğitsel programlama aracı olan Code Combat ile JavaScript programlama dili ile hedefe yönelik programlama yapmışlardır. Buna rağmen araştırmacının gözlem notlarına göre metin temelli programlama öğretiminde söz dizimi kuralları ve programlama kavramlarında öğrenciler ilk 2 hafta zorluk yaşamışlardır.

Blok temelli programlama aracını kullanmaya devam etme durumlarının bilgisayarca düşünme becerilerinin birleşik alt boyutlarında anlamlı bir etkisi olmadığı ortaya çıkmıştır. Ancak bilgisayarca düşünme becerisinin alt boyutları tek incelendiğinde sadece işbirliklilik becerisinde blok temelli programlama aracını kullanmaya devam etmenin anlamlı bir etkisi olduğu ortaya çıkmıştır. Bunun nedeni bu öğrencilerin programlamaya ilişkin öz-yeterlilik algılarının da yüksek olması olmuş olabilir. Ders esnasında öğretmenin izniyle akran yardımlaşmasına izin verilmiştir. Sonuç olarak programlama konusunda öz güveni ve öz-yeterliliği yüksek olan öğrenciler diğer arkadaşlarına yardım etmiş olabilirler. Blok temelli programlamaya devam eden öğrencilerin bu şekilde işbirliklilik becerileri de olumlu etkilenmiştir.

5.5 Araştırmanın Beşinci Alt Problemine Dair Tartışma ve Sonuçlar

Araştırmanın beşinci alt problemi olan “Ortaokul öğrencilerinin, blok temelli programlama öz-yeterlilik algıları, bilgisayarca düşünme becerileri ve akademik başarıları, ön düşünme durumuna göre anlamlı farklılık göstermekte midir?” olarak ifade edilmiştir.

Araştırma sonucunda elde edilen bulgulara göre öğrencilerin ön düşünme durumları blok temelli programlama öz-yeterlilik algıları, bilgisayarca düşünme becerileri ve akademik başarıları üzerindeki etkisi incelenmiştir. Bulgulara göre

öğrencilerin ön düşünme durumlarının sadece bilgisayarca düşünme becerisi ve alt boyutları ile daha ilgili olduğu söylenebilmektedir. Çıkan verilere göre yaratıcılık, algoritmik ve eleştirel düşünme becerilerine orta düzeyde etki sağlarken, problem çözme becerisine yüksek düzeyde etki sağlamaktadır. Ön düşünme durumları yüksek olan öğrencilerin orta olanlara göre yaratıcılık, algoritmik düşünme, eleştirel düşünme ve problem çözme becerilerine daha etkili olduğu ortaya çıkmıştır. Yükseltürk ve Altıok (2018) öğrencilerin bilgisayarca düşünmenin alt basamakları olan analitiklik ve eleştirel düşünebilen, problem çözme becerisi yüksek, iletişim becerileri gelişmiş bireylerin yetişmesinin programlama öğretimi ile gerçekleşeceğini ifade etmektedirler. Bocconi ve arkadaşları (2016) bilgisayarca düşünme becerilerinin ilerlemesinde programlama öğretiminin önemli bir araç olduğunu vurgulamışlardır. Araştırma kapsamındaki öğrencilerin bilgisayarca düşünme becerilerinin ve ön düşünme durumlarının oldukça yüksek olmasının nedeni daha önceden blok temelli programlama öğretimi almış olmalarından kaynaklı olabilmektedir. Bu doğrultuda metin temelli programlama öğretiminden sonra bilgisayarca düşünme becerileri ve alt boyutlarında daha iyi bir gelişim görülmesi söz konusu olmuştur. Ancak ön düşünme durumlarının, öğrencilerin akademik başarılarına ve blok temelli programlama öz-yeterlilik algılarına etkisi yoktur. Weintrop ve Wilensky (2015) çalışmasındaki sınırlı komut yapılarının olması, gerçek (metin) programlama dillerinin yapısına benzememesi ve gelişmiş programlama yapılarının kullanılmaması gibi özelliklerin blok temelli programlamanın olumsuz yanları olduğunu destekler niteliktedir. Meerbaum ve arkadaşları (2011), blok temelli programlama aracının özellikle algoritmik düşünme becerisi ve yazılım tasarımı seviyesinde düşünmeden komutlar yazıldığından dolayı problem çözme becerisinin gelişmeyeceğini belirtmişlerdir. Bu tez çalışmasında benzer şekilde öğrencilerin problem çözme becerilerinde, metin temelli programlama ile daha çok gelişme kaydedilmiştir. Yapılan literatür araştırmalarına göre blok temelli programlama araçlarının kısıtlı program geliştirme özelliğinden dolayı bilgisayarca düşünme becerileri, yaratıcılık ve problem çözme becerileri daha az etkilenebilmektedir (Kandemir, 2018b). Ancak programlama öğretimi bilgisayarca düşünme becerisinin gelişmesinde en etkin yöntemlerden biridir ve uzun süreli çalışmalarla desteklenmelidir (Bocconi vd., 2016; Kukul, 2018; Lu ve Fletcher, 2009). Bu doğrultuda bilgisayarca düşünme becerisinin ve alt boyutlarının gelişimi için programlama öğretiminin uzun süreçlere yayılarak hazırlanması gerektiği söylenebilmektedir.

5.6 Araştırmanın Altıncı Alt Problemine Dair Tartışma ve Öneriler

Araştırmanın altıncı alt problemi olan “Ortaokul öğrencilerinin, tasarlanan ders etkinliklerine, bilgisayarca düşünme becerilerine ve blok temelli programlama öz-yeterlilik algılarına yönelik betimsel görüşleri nelerdir?” olarak ifade edilmiştir.

Araştırma kapsamında uygulanan öğrenci görüşmelerinin analizi ile elde edilen bulgular doğrultusunda çalışma sonunda metin temelli programlama öğretimi ile tasarlanan ders etkinliklerinde kendilerini olumlu yönde etkilediği ve oldukça eğlendikleri yönündedir. Etkinlikler süresince araştırmacı tarafından alınan gözlem notları ve öğrencilerin ders etkinliklerini eğlenceli olarak değerlendirmesinin doğrultusunda etkinliklerin içeriklerinin güncel olduğunun ve öğrencilerin ilgisini çekmede önemli rol aldığı düşünülebilir. Gupta (2004) metin temelli programlama dili seçilirken hedef kitle ve kullanılacak programlama aracının öğrenmeye odaklı olması gerektiğini belirtmektedir. Öğrenciler metin temelli programlama dilinde zorlanmadan keyifli zaman geçirdiklerini belirterek bu yöndeki sonuçları doğrulanmıştır. Metin temelli programlama öğretiminde bilişsel gelişim dönemlerine uygun ve soyutlama, problem çözme gibi becerilerde zorlanmadan programlama kavramlarında ilerleme göstermeleri ve söz dizimsel kuralları uygulamaları için öğrencilerin hazırbulunmuşluk düzeyleri iyi bilinmeli, hedef kitle analiz edilmelidir (Kandemir, 2018a, 2018b). Buna göre hazırlanan ders uygulamaları hedef kitlenin seviyesinde verilmiştir. Programlama öğretimi sürecinde öğretmenin hedef kitlenin analizi konusunda dikkat etmesi gerekmektedir.

Yapılan görüşmeler sonunda öğrenciler bilgisayarca düşünme alt boyutlarından problem çözme, algoritmik düşünme ve işbirliklilik becerilerinde gelişme gördüklerini belirtmişlerdir. Yapılan analizler sonucunda öğrencilerin sadece problem çözme becerilerinde ilerleme olduğu ortaya çıkmıştır. Problem çözme ve algoritmik düşünme becerilerinin bilgisayarca düşünmenin her alanında kavramsal anlama için gerekli olduğu Bundy (2007) tarafından belirtilmiştir. Bunun dışında Czerkowski ve Lyman (2015) problem çözme becerilerinin, bilgisayarca düşünme ile gelişeceği fikrini desteklemektedir. Programlama ile düşünce, problem çözme becerisini güçlendiren ve bilgisayarca düşünme becerilerini kazanmalarını sağlayan önemli etkenlerdendir (Barr, 2011; Wing, 2006). İşbirliklilik becerilerinin geliştiğini söylemeleri sınıf ortamında akran yardımlaşmasına izin verilmesi ve bulgularda da ortaya çıkan öz-yeterlilik algısı yüksek olan öğrencilerin işbirliklilik içinde olmasından kaynaklı

olabilmektedir. Araştırma grubunun genel algoritmik düşünme becerisinin ise maksimum seviyeye oldukça yakın olduğu ortaya çıkmıştır. Bu yüzden öğrenciler algoritma geliştirme sürecinde becerilerinin geliştiğini belirtmiş olabilmektedirler.

Yapılan çalışmalarla öğrencilerin programlama kavramlarına hakimiyet düzeylerinin geliştiği ve akademik başarılarına katkı sağladığı yönündeki olumlu görüşleri ile metin temelli programlama öğretimindeki kazanımlara yönelik hedeflerin gerçekleştiği düşünülebilir. Burada bir diğer önemli faktör ise öğrencilerin programlama öğretimi sonunda akademik başarı testi ile değerlendirme sonuçlarını net bir şekilde görmesinden kaynaklı olabilmektedir. Ristić ve arkadaşları (2016) metin temelli programlama dili ve aracının programlama kavramlarını içermesi, güncel ve modern bir dil olması, söz dizimi kuralları basit ve derleyicisi kullanabilir olması görüşünü desteklemektedir. Çoklu paradigmaya sahip script (betik) bir dil olan JavaScript, web tabanlı olduğu için hem güncel hem de modern programlama dili olarak öğrencilerin meslek hayatlarında da işe yarayacak ve ilgilerini çeken bir programlama dilidir. Kandemir (2018b), öğrencilerin bilişsel gelişim dönemlerinden soyut dönemde nesne tabanlı ve fonksiyonel metin temelli programlama dilleri hedef kitleye yönelik doğru bir seçim olacağı görüşünü desteklemektedir. Gupta (2004) göre, programlama dilinin söz dizimi içerisindeki kurallardan daha çok tekniklere odaklanması gerektiği yönündedir; öğrencilerin görüşmelerindeki bulgulara göre programlama kavramları ve problem çözme odaklı etkinliklerin amacına ulaştığı söylenebilmektedir.

Uygulama kapsamında öğrencilerle yapılan görüşmelerde blok temelli programlama öz-yeterliliklerini metin temelli programlama ile programlama kavramlarını kullandıkları yönündedir. Öğrencilerin önceden blok temelli programlama dili öğrenmiş olmaları, daha özgüvenli hissetmelerini ve motivasyonlarının yüksek olmasını sağladığını göstermektedir. Kandemir (2018a, 2018b) programlama becerilerini kolaylıkla vermek ve motivasyonun yüksek olmasını sağlamak için blok temelli programlama araçlarının iyi bir seçim olacağını söylemektedir. Buna rağmen blok temelli programlama aracı ile programlama yapmaya alışkanlık kazanmış öğrencilerin metin temelli programlama dillerinde bazı olumsuzluklar yaşamaktadır (Meerbaum, 2011). Bununla birlikte bu çalışmada yapılan uygulamalar ile öğrenciler eğitsel programlama araçlarını kullanarak hedefe yönelik programlama ile metin temelli programlama diline hazırlanmıştır. Bu doğrultuda öğrencilerin programlamaya ilişkin öz-yeterlilik algıları, motivasyon ve öz

güvenleri yüksektir. Böylece bu çalışmada Dorling ve White (2015) blok temelli programlama araçlarını temel programlama kavramlarını öğretmek için, destekleyici bir araç olarak kullanılmasını ifade ettiğini destekler niteliktedir. İleri seviye programlama kavramlarının öğretimi için metin temelli programlama dillerine ihtiyaç olduğu görülmektedir.

5.4 Öneriler

Bu bölümde, yapılan çalışma sonucunda elde edilen bulgular doğrultusunda araştırmacılara ve uygulayıcılara yönelik öneriler verilmektedir.

5.4.1 Araştırmacılara yönelik öneriler.

- Araştırmada blok temelli programlama öz-yeterlilik algısına sahip olan tek grup ön test-son test çalışması ile yapılmıştır. Metin temelli programlama öğretiminin etkisini ölçmek için blok temelli programlama öz-yeterlilik algısına sahip olan ve olmayan deney-kontrol grubu ile gerçekleştirilebilir.
- Araştırmada tüm gruba metin temelli programlama öğretimi yapılmıştır, bir grup ile blok temelli programlama öğretimine devam edilerek diğer grupta metin temelli programlama öğretimi yapılarak öğrencilerin akademik başarısı, motivasyonu ve bilgisayarca düşünme becerileri arasındaki farklılık incelenebilir. Bu kapsamda iki gruba ait hazırlanacak olan ders planlarının ve kazanımlarının aynı olmasına dikkat edinilmesi gerekmektedir.
- Blok temelli programlama öz-yeterlilik algısına sahip olan deney-kontrol grubu ile metin ve robot temelli programlama aracının bilgisayarca düşünme becerilerine etkisi araştırılabilir. Bu sayede bilgisayarca düşünmenin alt boyutlarında somut programlamanın etkisi ölçülebilir.
- Bu çalışmada öğrencilerin blok temelli akademik başarı puanları bulunmamaktadır. Öğrencilerin blok temelli programlama akademik başarılarına göre metin temelli akademik başarıları arasındaki farklılık incelenebilir.
- Bu araştırmada öğrencilerin blok temelli programlamaya ilişkin

öz-yeterlilik algıları ölçülmüştür. Ancak araştırma sonucunda öğrencilerin metin temelli programlamaya yönelik öz-yeterlilik algıları, kaygı ve motivasyon düzeylerine bakılabilir.

- Metin temelli programlama dili farkının akademik başarıya etkisi araştırılabilir. Nesne tabanlı, fonksiyonel veya dinamik programlama dillerinin öğrencilerin metin temelli programlamaya yönelik öz-yeterlilik algıları, kaygı ve motivasyon düzeylerine bakılabilir.
- Farklı bağımsız değişkenler ile bağımlı değişkenlere olan etkisi araştırılabilir.
- Metin temelli programlama dili bir katalizör olarak kullanılıp bilişsel yüke etkisi araştırılabilir.
- Bu araştırma aynı kademede bulunan ortaokul öğrencilerine uygulanmıştır. Çalışma farklı eğitim kademelerinde uygulanarak bilişsel gelişim dönemlerine ve bilgisayarca düşünme becerilerine etkisi araştırılabilir.

5.4.2 Uygulamacılara Yönelik Öneriler.

- Metin temelli programlama öğretimi öncesinde, eğitsel programlama araçları ile öğrencilere hedefe yönelik programlama araçları (Code Monkey, Code Combat, Blockly vs.) kullanılmalıdır. Bu şekilde metin temelli programlamada artan bilişsel yük azaltılmış olacaktır. Öğrenciler bu sayede metin temelli programlamada bulunan sözel kod yapısı ve derlemede hata ayıklama gibi genel kurallara dolaylı yoldan alışacaktır.
- Programlamada acemi olan öğrenciler için blok temelli programlama araçları programlamanın temel kavramlarını öğretmek, motivasyonu, öz güveni ve öz-yeterlilik algısını sağlamak için kullanılabilir. Ancak ileri programlama kavramlarının öğretiminde metin temelli programlama araçlarının bilişsel gelişim dönemine ve hedef kitleye göre seçilerek uygulanmalıdır.

- Programlama, bilgisayarca düşünme becerisinin ve alt boyutlarının geliştirilmesinde kullanılabilir. Bilgisayarca düşünme becerisi gelişiminin bir düşünce süreci olduğu bilinmelidir ve ürün odaklı çalışmalarla değerlendirilmemelidir. Uzun süreli uygulamalar tasarlanması daha etkili olabilir.
- Metin temelli programlama aracı seçilirken programlama öğretimindeki pedagojik yaklaşımlara ve programlama dillerinin paradigmalarına önem gösterilmelidir. Programlama öğretimi zor bir süreçtir ve öğrencilerin bu süreci verimli bir şekilde kullanmaları gerekmektedir. 21. yüzyıl bireyleri olarak programlama ve bilgisayarca düşünme becerilerini edinmelidirler.
- Metin temelli programlama öğretiminde öğrencilerin bilişsel gelişim dönemlerine ve programlama hazırbulunuşluk düzeylerine göre yaratıcılık, algoritmik düşünme, işbirliklilik, eleştirel düşünme ve problem çözme becerilerine odaklı etkinlikler ve uygulamalar hazırlanmalıdır.
- Metin temelli programlama öğretiminde hazırlanan ders etkinlikleri öğrencilerin ihtiyaçlarına ve ilgilerine yönelik olmasına özen gösterilmelidir.
- Metin temelli programlama dillerinin popüleritesi devam etmektedir. Bu programlama dillerine hakim olan BT uzmanları ya da programcılara ihtiyaç gün geçtikçe artmaktadır. Verimli uzmanların yetiştirilmesinde ve mesleki yönlendirmelerde programlama ve bilgisayarca düşünme becerisinin değerlendirilmeleri önemli olabilir.
- Metin temelli programlama dilinin modern ve güncel olmasına özen gösterilmelidir. Böylece öğrenciler öğrendikleri programlama dilini gerçek hayat ile eşleştirebileceklerdir.
- Seçilecek olan metin temelli programlama dilinin kullanıcı dostu derleyiciye sahip olmasına dikkat edinilmelidir. Bu sayede ders esnasında derleyiciden hata alma ve ayıklamaya yeterli zaman kalacaktır. Blok ve metin temelli programlama derleyici

araçlarında hata ayıklama yöntemlerinin farklı olmasından dolayı öğrenciler metin temelli programlama araçlarını kullanırken zaman kaydebilirler.

5.4.3 Tasarlanan Ders Uygulamalarına Yönelik Öneriler.

- Metin temelli programlama öğretimine yönelik uygulamalar geliştirirken hedef kitle analizi dikkatle yapılmalıdır. Etkinlikler gerçek hayattan örneklerle ilişkilendirilmelidir.
- Metin temelli programlama öğretimine yönelik etkinlikler planlanırken bilgisayarca düşünme becerisi boyutlarına ilişkin daha ayrıntılı ders etkinlikleri hazırlanabilir.
- Metin temelli programlama öğretimine yönelik uygulamalarda hedef kitleye ve bilişsel gelişim dönemine uygun farklı programlama dilleri ve derleyici araçları tercih edilebilir. Bu çalışmada seçilen programlama dili zümre içi toplantıda alınan kararla kullanılmıştır. Aynı zamanda güncel ve popüler bir programlama dili olması, dünyada web programlama dillerine ait müfredatların ön planda olması da etken olmuştur. Ancak Small Basic, Phyton gibi programlama dilleri de tercih edilebilir.
- Metin temelli programlama akademik başarı testi değerlendirilmesine ek olarak uygulama testi de yapılabilir. Bu sayede öğrencilerin uygulamaya dair hakimiyet durumları ve programlama becerisi ölçülebilir.

KAYNAKLAR

- Aldağ, H., & Tekdal, M. (2015). Bilgisayar kullanımı ve programlama öğretiminde cinsiyet farklılıkları. *Proceeding of 1. Uluslararası Çukurova Kadın Çalışmaları Kongresi*, 236-243.
- Alsancak Sırakaya, D. (2019). *Programlama Öğretiminin Bilgi İşlemsel Düşünme Becerisine Etkisi*. 575–590.
- Alp, Y. (2019). *Blok Tabanlı Programlama Öğretiminin Ortaokul Öğrencilerinin Problem Çözme Becerisine ve Bilgisayara Yönelik Tutumuna Etkisi*. (Yayınlanmış yüksek lisans tezi). İnönü Üniversitesi Eğitim Bilimleri Enstitüsü, Malatya.
- Altun, A., ve Kasalak, İ. (2018). Blok Temelli Programlamaya İlişkin Öz-Yeterlik Algısı Ölçeği Geliştirme Çalışması: Scratch Örneği. *Eğitim Teknolojisi Kuram ve Uygulama*, 8(1).
- Amaral, M. A., Emer, M. C. F. P., Bim, S. A., Setti, M. G. ve Gonçalves, M. M. (2017). *Investigating Gender Issues in an Undergraduated Computing Program*. <https://doi.org/10.1590/1806-9584.2017v25n2p857> adresinden 5 Ocak 2020 tarihinde edinilmiştir.
- Angeli, C., Voogt, J., Fluck, A., Webb, M., Cox, M., Malyn-Smith, J., ve Zagami, J. (2016). A K-6 computational thinking curriculum framework: Implications for teacher knowledge. *Educational Technology and Society*, 19(3), 47–57.
- Atiker, B. (2019). *Programlama Öğretiminde Ortaokul Öğrencilerinin Bilgi İşlemsel Düşünme Becerilerinin Başarıya Etkileri*. İstanbul Üniversitesi Fen Bilimleri Enstitüsü, İstanbul.
- 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.

<https://dergipark.org.tr/asead/issue/40925/494055> adresinden 29 Nisan 2019 tarihinde edinilmiştir.

Bağcı, B. B. (2017). *The Effect of Visual and Text Interfaces in Teaching Robot Programming*. (Yayınlanmış yüksek lisans tezi). İstanbul Teknik Üniversitesi Bilgisayar Mühendisliği Enstitüsü, İstanbul.

Balanskat, A., ve Engelhardt, K. (2015). *Computing our future - computer programming and coding*. (Ekim).

Balanskat, A., Engelhardt, K., ve Lict, A. H. (2018). *Strategies to include computational thinking in school curricula in Norway and Sweden*.

Barr, D., Harrison, J., ve Conery, L. – ISTE (2011). March-2011-Computational Thinking-LL386.pdf. *ISTE Learning and Leading with Technology*, 5191(Mart/Nisan), 21–22.

Barr, V., ve Stephenson, C. (2011). Bringing computational thinking to K12. *ACM Inroads*, 2(1), 48. <https://doi.org/10.1145/1929887.1929905>

Baştemur-Kaya, C. (2018). *Programlama Dili Öğretiminde Alice Programının Öğrencilerin Akademik Başarıları, Problem Çözme Becerisi Algıları, Motivasyonları ve Programlamaya Hazır Bulunuşluk Düzeyleri Üzerine Etkisi*. Gazi Üniversitesi Eğitim Bilimleri Enstitüsü, Ankara.

Baştuğ, İ. (2019). *Jquery Tabanlı Blok Programlama Öğretiminin Programlamaya Yönelik Tutuma Etkisinin Değerlendirilmesi (Başiskele Örneği)*. (Yayınlanmış yüksek lisans tezi). Sakarya Üniversitesi Eğitim Bilimleri Enstitüsü, Sakarya.

Bayman, P., ve Mayer, R. E. (1988). Using Conceptual Models to Teach BASIC Computer Programming. *Journal of Educational Psychology*, 80(3), 291–298. <https://doi.org/10.1037/0022-0663.80.3.291>.

Bennedsen, J. (2008). *Teaching and Learning Introductory Programming*. 227.

Bergin, J., Brodlie, K., Goldweber, M., Jimenez-Peris, R., Khuri, S., Patino-Martinez, M., Wilson, J. (1996). Overview of visualization: its use and design.

Report of the Working Group on Visualization. *SIGCSE Bulletin (Association for Computing Machinery, Special Interest Group on Computer Science Education)*, 28(Special Issue), 192–200.

Blackwell, A. F. (2002). First steps in programming: A rationale for attention investment models. *Proceedings - IEEE 2002 Symposia on Human Centric Computing Languages and Environments, HCC 2002*, 2–10.
<https://doi.org/10.1109/HCC.2002.1046334>.

Bloom, B. S., Engelhart, M. D., Furst, E. J., Hill, W. H., ve Krathwohl, D. R. (1956). The Classification of Educational Goals. *Taxonomy of educational objectives*, 207.

Bocconi, S., Chiocciariello, A., Dettori, G., Ferrari, A., Engelhardt, K., Kampylis, P., ve Punie, Y. (2016). Exploring the Field of Computational Thinking as a 21st Century Skill. *EDULEARN16 Proceedings, 1*. (Haziran), 4725–4733.
<https://doi.org/10.21125/edulearn.2016.2136>.

Boechler, P., Artym, C., Dejong, E., Carbonaro, M., ve Stroulia, E. (2014). Computational thinking, code complexity, and prior experience in a videogame-building assignment. *Proceedings - IEEE 14th International Conference on Advanced Learning Technologies, ICALT 2014*, 396–398.
<https://doi.org/10.1109/ICALT.2014.118>.

Boyle, R., & Jenkins, T. (1996). *University of Leeds School of Computer Studies. Generating Motivation in New Students of IT*. (Haziran), 27–30.

Brauner, P., Leonhardt, T., Ziefle, M., ve Schroeder, U. (2010). *Teaching Fundamentals Concepts of Informatics*. 5941(Ocak), 35.
<https://doi.org/10.1007/978-3-642-11376-5>.

Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. *In Proceedings of the 2012 annual meeting of the American Educational Research Association (pp. 1–25)*. Vancouver, Canada.

- Bundy, A. (2007). Computational Thinking is Pervasive. *Journal of Scientific and Practical Computing*, 1(2), 67–69.
- Büyüköztürk, Ş. (2019). *Sosyal Bilimler için Veri Analizi El Kitabı İstatistik, Araştırma Deseni SPSS Uygulamaları ve Yorum*. Ankara: PEGEM Akademi.
- Büyüköztürk, Ş., Kılıç, E., Akgün, Ö. E., Karadeniz, Ş., ve Demirel, F. (2016). *Bilimsel araştırma yöntemleri*. Ankara: PEGEM Yayınları.
- Byrne, P., & Lyons, G. (2001). The Effect of Student Attributes on Success in Programming. *SIGCSE Bull.*, 33(3), 49-52.
- Can, A. (2019). *SPSS ile Bilimsel Araştırma Sürecinde Nicel Veri Analizi*. Ankara: PEGEM Akademi.
- Carter, J., ve Jenkins, T. (1999). Gender and programming: What’s going on? *SIGCSE Bulletin (Association for Computing Machinery, Special Interest Group on Computer Science Education)*, 31(3), 1–4.
<https://doi.org/10.1145/384267.305824>.
- Çaycı, B., Demir, M. K., Başaran, M., ve Demir, M. (2007). *Sosyal Bilgiler Dersinde İşbirliğine Dayalı Öğrenme ile Kavram Öğretimi*. (2), 619–630.
- Cevahir, H., ve Özdemir, M. (2017). Programlama Öğretiminde Karşılaşılan Zorluklara Yönelik Öğretmen Görüşleri ve Çözüm Önerileri. *Kastamonu Eğitim Dergisi (Mayıs)*, 320–335.
- Çetin, İ., Toluk Uçar, Z. (2018). Bilgi İşlemsel Düşünme Tanımı ve Kapsamı. Y. Gülbahar (Ed.), *Bilgi İşlemsel Düşünmeden Programlamaya (ss. 41-74)*. Ankara: PEGEM Akademi.
- Civelek, M. (2014). *Ölçme ve Değerlendirme*.
<https://kaanonaran.files.wordpress.com/2015/02/olcme-ve-degerlendirme-22-07-2013.pdf> adresinden 25 Kasım 2019 tarihinde edinilmiştir.
- Cohen, J. (1988). *Statistical Power Analysis for the Behavioral Sciences Second Edition*. New York: Lawrence Erlbaum Associates.

- Creswell, J. W. (2017). *Araştırma Deseni: Nitel, Nicel ve Karma Yöntem Yaklaşımları*. Ankara: Eğiten Kitap Yayıncılık.
- CSTA, (2009). A Model Curriculum for K12 Computer Science Level I Objectives and Outlines. *Computer Science Teachers Association*. Retrieved on January, 4, 2011.
- CSTA. (2017). *CSTA K–12 Computer Science Standards - Revised 2017*. (Haziran), <https://www.csteachers.org/page/standards> adresinden 30 Aralık 2019 tarihinde edinilmiştir.
- Cuny, J., Snyder, L., & Wing, J. M. (2010). Demystifying computational thinking for non-computer scientists. Unpublished manuscript in progress. <http://www.cs.cmu.edu/~CompThink/resources/TheLinkWing.pdf> adresinden 30 Aralık 2019 tarihinde edinilmiştir.
- Czerkawski, B. C., ve Lyman, E. W. (2015). Exploring Issues About Computational Thinking in Higher Education. *TechTrends*, 59(2), 57–65. <https://doi.org/10.1007/s11528-015-0840-3>.
- Demir, F. (2015). *Programlama Öğretiminde Eğitsel Programlama Dilinin Farklı Kullanımlarının Programlama Başarısı ve Kaygısına Etkisi*. (Yayınlanmış doktora tezi). Atatürk Üniversitesi Eğitim Bilimleri Enstitüsü, Erzurum.
- Demir, Ö., ve Seferoglu, S. S. (2017). Yeni Kavramlar, Farklı Kullanımlar: Bilgi-İşlemsel Düşünmeyle İlgili Bir Değerlendirme. H. F. Odabaşı, B. Akkoyunlu ve A. İşman (Ed.), *Eğitim Teknolojileri Okumaları 2017* (ss. 801-830).
- Denning, P. J. (1997). Computer Science: The Discipline. *Encyclopedia of Computer Science*. (Ağustos).
- Denning, P. J. (2013). *Peter J. Denning*. 98 (Ekim 2010), 198–202.
- Denning, P. J. (2017). Remaining trouble spots with computational thinking. *Communications of the ACM*, 60(6), 33–39. <https://doi.org/10.1145/2998438>.

- Denning, P. J., ve Rosenbloom, P. S. (2009). The profession of IT Computing. *Communications of the ACM*, 52(9), 27. <https://doi.org/10.1145/1562164.1562176>.
- Doğan, U. (2015). *Ortaokul Öğrencilerinde Bilgisayar Oyunu Geliştirme Sürecinin Öğrencilerin Eleştirel Düşünme Becerilerine Ve Algoritma Başarılarına Etkisi*. (Yayınlanmış yüksek lisans tezi). Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü, İstanbul.
- Dorling, M., ve White, D. (2015). Scratch: A way to logo and python. *SIGCSE 2015 - Proceedings of the 46th ACM Technical Symposium on Computer Science Education*, 191–196. <https://doi.org/10.1145/2676723.2677256>.
- Ekici, G. (2016). *Öğrenme - Öğretme Kuramları ve Uygulamadaki Yansımaları*. Ankara: Pegem Akademi.
- Erdem, E., ve Kalelioğlu, F. (2018). *Blok Tabanlı Ortamlarda Programlama Öğretimi Sürecinde Farklı Öğretim Stratejilerinin Çeşitli Değişkenler Açısından İncelenmesi*. (Yayınlanmış yüksek lisans tezi). Başkent Üniversitesi Eğitim Bilimleri Enstitüsü, Ankara.
- Erdinç, Z., Erdinç, F. (2017). *Uygulamalarla Programlama Öğreniyorum*. İstanbul: Abaküs Yayınları.
- Erümit, A. K., Berigel, M. (2018). Programlama Dillerinin Tarihi ve Programlama Öğretimi. Y. Gülbahar, H. Karal (Ed.), *Kuramdan Uygulamaya Programlama Öğretimi* (ss. 2-36). Ankara: PEGEM Akademi.
- Farooq, M. S., Khan, S. A., Ahmed, F., ve Islam, S. (2014). *Choice of Pedagogical Approaches towards First Programming Languages*. 4, 311–317.
- Fatih, M. Y. (2016). *Görsel Programlama Ortamı ile Öğretimin Öğrencilerin Bilgisayar Programlamayı Öğrenmesine ve Programlamaya Karşı Tutumlarına Etkisinin İncelenmesi*. (Yayınlanmış yüksek lisans tezi). Ondokuz Mayıs Üniversitesi Eğitim Bilimleri Enstitüsü, Samsun.
- Fidan, P. N. (2012). *Okulda Öğrenme ve Öğretme*. Pegem Akademi: İstanbul.

Finnish National Board of Education. (2016). New national core curriculum for basic education: focus on school culture and integrative approach. *Finnish National Board of Education - New National Core*, 2.

<https://www.oph.fi/sites/default/files/documents/new-national-core-curriculum-for-basic-education.pdf> adresinden 2 Kasım 2019 tarihinde edinilmiştir.

Futschek, G. (2006). Algorithmic thinking: The key for understanding computer science. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 4226 LNCS, 159–168. https://doi.org/10.1007/11915355_15.

Gagne, R. M., ve M, G. R. (1984). The condition of learning. *Gagne R M. Domains of learning. Interchange*, 2(21), 1–8.

Google Exploring Computational Thinking-ECT, (2016). *Google Exploring Computational Thinking*. <https://edu.google.com/resources/programs/exploring-computational-thinking/> adresinden 3 Ekim 2019 tarihinde edinilmiştir.

Gölbucaklı, İ. (2016). APOS. <https://prezi.com/og7ojceywcwx/apos/> adresinden 25 Kasım 2020 tarihinde edinilmiştir.

Grover, S. (2018). A Tale of Two CTs (and a Revised Timeline for Computational Thinking) | blog@CACM | Communications of the ACM. *ACM Blog*, 10–13, <https://cacm.acm.org/blogs/blog-cacm/232488-a-tale-of-two-cts-and-a-revised-timeline-for-computational-thinking/fulltext> adresinden 12 Ekim 2019 tarihinde edinilmiştir.

Gülbahar, Y. (2018). *Bilgi İşlemsel Düşünmeden Programlamaya*. Ankara: Pegem Akademi.

Gülbahar, Y., Karal, H. (2018). *Kuramdan Uygulamaya Programlama Öğretimi*. Ankara: Pegem Akademi.

Günay, D., Şişman, B. (2019). Bilgi ve İletişim Teknolojileri Okuryazarlığı. A. D. Öğretim Özçelik, M. N. Tuğluk (Ed.), *Eğitimde ve Endüstride 21. Yüzyıl Becerileri (ss. 257-273)*. Ankara: Pegem Akademi.

- Gürer, D., & Camp, T. (2002). An ACM-W literature review on women in computing. *SIGCSE Bull*, 34(2), 121-127.
- Green, S. B., Salkind, N. J. (2005). *Using SPSS for Windows and Macintosh: Analyzing and Understanding Data (4th Edition)*. New Jersey: Pearson.
- Gupta, D. (2004). What is a good first programming language? *Crossroads*, 10(4), 7–7. <https://doi.org/10.1145/1027313.1027320>
- Halpern, D. F. (1996). *Thinking Critically about Critical Thinking*. Mahwah, NJ: Lawrence Erlbaum Associates.
- Hamarat, E. (2019). *21. Yüzyil Becerileri Odağında Türkiye'nin Eğitim Politikaları*. (Nisan).
- Helminen, J., ve Malmi, L. (2010). Jype - A program visualization and programming exercise tool for python. *Proceedings of the ACM Conference on Computer and Communications Security*, 153–162. <https://doi.org/10.1145/1879211.1879234>
- Hoc, J. M., Green, T. R. G., Samurçay, R., ve Gilmore, D. J. (1990). Introduction to Theoretical and Methodological Issues. *Psychology of Programming*, 1–7. <https://doi.org/10.1016/b978-0-12-350772-3.50005-7>.
- International Society for Technology in Education. – ISTE (2019). ISTE Standards for computer science educators. *Journal of Ecperiential Education*, 1–2.
- ISTE. (2016). *ISTE Standards for Students*. www.iste.org/standards sitesinden 9 Ekim 2019 tarihinde edinilmiştir.
- ISTE. (2019). *ISTE standards for educators: Computational thinking competencies*. www.iste.org/standards sitesinden 9 Ekim 2019 tarihinde edinilmiştir.
- Jenkins, T. (2004). The First Language - A Case for Python. *Innovation in Teaching and Learning in Information and Computer Sciences*, 3(2), 1–9.

- 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.
- Kaleliođlu, F. (2018). Trkiye'de Programlama đretimi. Y. Glbahar, H. Karal (Ed.), *Kuramdan Uygulamaya Programlama đretimi* (ss. 38-65). Ankara: PEGEM Akademi.
- Kandemir, C.M. (2018a). Metin Tabanlı Programlama. Y. Glbahar (Ed.), *Bilgi İşlemsel Dşnmeden Programlamaya* (ss. 267-292). Ankara: PEGEM Akademi.
- Kandemir, C.M. (2018b). Metin Tabanlı Programlama. Y. Glbahar, H. Karal (Ed.), *Kuramdan Uygulamaya Programlama đretimi* (ss. 299-336). Ankara: PEGEM Akademi.
- Kert, S.B. (2018a). Bilgisayar Bilimi Eđitimine Giriş. Y. Glbahar (Ed.), *Bilgi İşlemsel Dşnmeden Programlamaya* (ss. 1-20). Ankara: PEGEM Akademi.
- Kert, S.B. (2018b). Bilişsel, Duyuşsal ve Sosyal Aıdan Programlama đretimi. Y. Glbahar, H. Karal (Ed.), *Kuramdan Uygulamaya Programlama đretimi* (ss. 133-157). Ankara: PEGEM Akademi.
- Kert, S. B., ve Uđraş, T. (2009). Programlama Eđitiminde Sadelik ve Eđlence: Scratch rneđi. *The First International Congress of Educational Research, Çanakkale*.
- Kim, H.-Y. (2013). *Statistical notes for clinical researchers: assessing normal distribution (2) using skewness and kurtosis*. Restorative Dentistry ve Endodontics, 38(1), 52–54. <http://doi.org/10.5395/rde.2013.38.1.52>
- Korkmaz, ., Çakır, R., ve zden, M. Y. (2017). A validity and reliability study of the computational thinking scales (CTS). *Computers in Human Behavior*, 72, 558–569. <https://doi.org/10.1016/j.chb.2017.01.005>

- Korkmaz, Ö., Çakır, R., ve Özden, M. Y. (2015). Computational thinking levels scale (CTLS) adaptation for secondary school level. *Gazi Journal of Education Sciences, 1*(2), 143–162.
- Korkmaz, Ö., Çakır, R., Özden, M. Y., Oluk, A., ve Sarıoğlu, S. (2015). Bireylerin Bilgisayarca Düşünme Becerilerinin Farklı Değişkenler Açısından İncelenmesi. *Ondokuz Mayıs Üniversitesi Eğitim Fakültesi Dergisi, 34*(2), 68–87.
- Krathwohl, D. R., Anderson, L. W., ve Bloom, B. S. (2001). *A taxonomy for learning, teaching, and assessing: a revision of Bloom's taxonomy of educational objectives. 41*(4), 352.
<http://books.google.com/books?id=JPKXAQAAMAAJvepgis=1> adresinden 2 Aralık 2019 tarihinde edinilmiştir.
- Kukul, V. (2018). *Programlama Öğretiminde Farklı Yapılandırılan Süreçlerin Öğrencilerin Bilgi İşlemsel Düşünme Becerilerine, Özyeterliliklerine ve Programlama Başarılarına Etkisi*. (Yayınlanmış doktora tezi). Gazi Üniversitesi Eğitim Bilimleri Enstitüsü, Ankara.
- Kurihara, A., Sasaki, A., Wakita, K., ve Hosobe, H. (2015). A programming environment for Visual block-based domain-specific languages. *Procedia Computer Science, 62* (Aralık), 287–296.
<https://doi.org/10.1016/j.procs.2015.08.452>
- Lee, I., Martin, F., ve Apone, K. (2014). Integrating computational thinking across the K–8 curriculum. *ACM Inroads, 5*(4), 64-71.
- Lin, C., ve Zhang, M. (2003). The use of computer animation in teaching discrete structures course. *MICS 2003 Proceedings The 36th Annual Midwest Instruction and Computing Symposium*.
- Lu, J. J., ve Fletcher, G. H. L. (2009). Thinking about computational thinking. *ACM SIGCSE Bulletin, 41*(1), 260. doi:10.1145/1539024.1508959.

- Malan, D.J. ve Leitner, H.H. (2007). Scratch for budding computer scientists. *38th ACM Technical Symposium on Computer Science Education*. Covington, Kentucky.
- Maloney, J., Resnick, M., Rusk, N., Pepler, K. A., ve Kafai, Y. B. (2008). Media designs with scratch: What urban youth can learn about programming in a computer clubhouse. *Computer-Supported Collaborative Learning Conference, CSCL, (PART 3)*, 81–82.
- Mayer, R. E. (1998). Cognitive, metacognitive, and motivational aspects of problem solving. *Instructional Science*, 26(1–2), 49–63.
- MEB. (2006). *İlköğretim Bilgisayar Dersi (1–8. Sınıflar) Öğretim Programı*.
- MEB. (2012). *Ortaokul ve İmam Hatip Ortaokulu Bilişim Teknolojileri ve Yazılım Dersi (5, 6, 7 Ve 8. Sınıflar) Öğretim Programı*.
- MEB. (2012). *Bilişim Teknolojileri ve Yazılım Dersi Öğretim Programı (Ortaokul 5 ve 6. Sınıflar)*.
- MEB. (2017). *Bilişim Teknolojileri ve Yazılım Dersi Öğretim Programı (Ortaokul 5 ve 6. Sınıflar)*.
- MEB. (2018). *Bilişim Teknolojileri ve Yazılım Dersi Öğretim Programı (İlkokul 1, 2, 3 ve 4. Sınıflar)*.
- MEB. (2018). *Özel Eğitim Hizmetleri Yönetmeliği*.
https://orgm.meb.gov.tr/meb_iys_dosyalar/2018_07/09101900_ozel_egitim_hizmetleri_yonetmeli_07072018.pdf adresinden 3 Ekim 2019 tarihinde edinilmiştir.
- Meerbaum-Salant, O., Armoni, M., ve Ben-Ari, M. (2011). Habits of programming in scratch. *ITiCSE'11 - Proceedings of the 16th Annual Conference on Innovation and Technology in Computer Science*, 168–172.
<https://doi.org/10.1145/1999747.1999796>.

- Metcalf, M. (1985). *Fortran, Programming Language*. CERN-Data Handling Division: Geneva, Switzerland.
- Mıhçı, C. (2014). *Programlama Eğitiminde Görsel Blok Programlama Ve Mobil Uygulama Geliştirme Araçlarının Karşılaştırılması*. (Yayınlanmış yüksek lisans tezi). Marmara Üniversitesi Eğitim Bilimleri Enstitüsü, İstanbul.
- Morris, D., & Trushell, J. (2014). Computer programming, ICT and gender in the classroom: a male-dominated domain or a female preserve? *Research in Teacher Education*, 4(1), 4–9. <https://doi.org/10.3109/10826084.2014.984850>.
- Morvan, C. (2018). *European Schoolnet's 2018 Annual Report*. <http://www.eun.org/resources/country-reports> adresinden 11 Kasım 2019 tarihinde edinilmiştir.
- Nouri, J., Zhang, L., Mannila, L., & Norén, E. (2019). Development of computational thinking, digital competence and 21st century skills when learning programming in K-9. *Education Inquiry*, 1–17. doi:10.1080/20004508.2019.1627844.
- Oluk, A., Korkmaz, Ö., ve Oluk, H. A. (2018). Effect of Scratch on 5th Graders' Algorithm Development and Computational Thinking Skills. *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, 9(1), 54–71. <https://doi.org/10.16949/turkbilmat.399588>.
- Othman, M., & Latih, R. (2006). Women in computer science: No shortage here. *Communications of the ACM*, 49(3), 111–114. <https://doi.org/10.1145/1118178.1118185> adresinden 1 Ocak 2020 tarihinde edinilmiştir.
- Öigaard, S. (2015). *Programming language and Gender*. <http://lnu.diva-portal.org/smash/get/diva2:846305/FULLTEXT01.pdf> adresinden 5 Ocak 2020 tarihinde edinilmiştir.
- Partnership for 21st Century Skills-P21. (2017). *21St Century Skills Early Learning 21St Century Skills Early*.

http://static.battelleforkids.org/documents/p21/P21_ELF_Framework_Final_20pgs.pdf adresinden 20 Kasım 2019 tarihinde edinilmiştir.

Papert, S. (1980). *Mindstorms: Children, Computers and Powerful Ideas*. New York: Basic Books. <http://worrydream.com/refs/Papert%20-%20Mindstorms%201st%20ed.pdf> adresinden 3 Aralık 2019 tarihinde edinilmiştir.

Papert, S., ve Harel, I. (1991). Situating Constructionism. *Constructionism*, 1–12.

Papert, S., ve Harel, I. (2017). Situating Constructionism. *Cell*, 151(1), 2.

Pijpers, R. (2017). *Alles wat u moet weten over 21e eeuwse vaardigheden*. <https://www.kennisnet.nl/artikel/alles-wat-je-moet-weten-over-21e-eeuwse-vaardigheden/> adresinden 8 Ocak 2020 tarihinde edinilmiştir.

Polat, C., ve Odabaş, H. (2008). Bilgi toplumunda yaşam boyu öğrenmenin anahtarı: Bilgi okuryazarlığı. *Küreselleşme, Demokratikleşme ve Türkiye Uluslar arası Sempozyumu*, 143–151.

Powers, K., Ecott, S., ve Hirshfield, L. M. (2007). Through the looking glass: Teaching CS0 with Alice. *SIGCSE 2007: 38th SIGCSE Technical Symposium on Computer Science Education*, 213–217. <https://doi.org/10.1145/1227310.1227386>

Raadt, M. de. (2008). *Teaching programming strategies explicitly to novice programmers*. 214. http://eprints.usq.edu.au/4827/2/de_Raadt_2008_whole.pdf adresinden 12 Aralık 2019 tarihinde edinilmiştir.

Resnick, M., Silverman, B., Kafai, Y., Maloney, J., Monroy-Hernández, A., Rusk, N., Silver, J. (2009). Scratch: Programming for Everyone. *Communications of the ACM*, 52(11), 60. <https://doi.org/10.1145/1592761.1592779>.

Ristić, O., Milošević, D., ve Urošević, V. (2016). The importance of programming languages in education. *6th International Conference, Faculty of Technical Sciences*, (May), 243–249.

- Rose, S., Habgood, J., & Jay, T. (2017). An exploration of the role of visual programming tools in the development of young children's computational thinking. *Electronic Journal of E-Learning*, 15(4), 297–309.
- Rubio, M. A., Romero-Zaliz, R., Mañoso, C., ve De Madrid, A. P. (2015). Closing the gender gap in an introductory programming course. *Computers and Education*, 82, 409–420. <https://doi.org/10.1016/j.compedu.2014.12.003>.
- Şahiner, A. (2017). *Komputasyonel Düşünme Kavramı ile İlgili 2006-2016 Yılları Arasındaki Bilimsel Yayınların İncelenmesi: Doküman Analizi Çalışması*. (Yayınlanmış yüksek lisans tezi). Yıldız Teknik Üniversitesi Eğitim Bilimleri Enstitüsü, İstanbul.
- Şahiner, A., ve Kert, S. B. (2016). Komputasyonel Düşünme Kavramı ile İlgili 2006-2015 Yılları Arasındaki Çalışmaların İncelenmesi. *Avrupa Bilim ve Teknoloji Dergisi*. 5(9), 38–43.
- Saritepeci, M., ve Durak, H. (2017). Analyzing the effect of block and robotic coding activities on computational thinking in programming education. *Educational research and practice*, (Mayıs), 438.
- Saygıner, Ş. (2017). *Programlama Öğretimlerinin Erişi, Mantıksal Düşünme Ve Motivasyona Etkileri*. (Yayınlanmış yüksek lisans tezi). Hacettepe Üniversitesi Eğitim Bilimleri Enstitüsü, Ankara.
- Saygıner, Ş., ve Tüzün, H. (2017). Programlama Eğitiminde Yaşanan Zorluklar ve Çözüm Önerileri*. *Uluslararası Bilgisayar ve Öğretim Teknolojileri Sempozyumu, İnönü Üniversitesi*.
- Sedgewick, R., ve Wayne, K. (2011). *Algorithms*. Princeton University: Pearson Education.
- Seferoğlu, S. S., & Akbıyık, C. (2006). *Teaching Critical Thinking* [in Turkish]. Hacettepe University Journal of Education, 30, 193-200.
- Senemoğlu, N., (1997). *Gelişim Öğrenme ve Öğretim*. Ankara: Ertem Matbaacılık.

- SLO., (2019a). *Voorbeeld Leermiddelen Computational Thinking*.
<https://slo.nl/vakportalen/vakportaal-digitale-geletterdheid/computational-thinking/voorbeeld-leermiddelen/> adresinden 6 Ocak 2020 tarihinde edinilmiştir.
- SLO., (2019b). *Voorbeeld Leerdoelen Computational Thinking*.
<https://slo.nl/vakportalen/vakportaal-digitale-geletterdheid/computational-thinking/voorbeeld-leerdoelen/> adresinden 6 Ocak 2020 tarihinde edinilmiştir.
- SLO., (2020). *Computational Thinking*. <https://slo.nl/vakportalen/vakportaal-digitale-geletterdheid/computational-thinking/> adresinden 6 Ocak 2020 tarihinde edinilmiştir.
- Şendurur, P. (2018). Bilişsel Araçlar ve Bilgi İşlemsel Düşünme. Y. Gülbahar (Ed.), *Bilgi İşlemsel Düşünmeden Programlamaya (ss. 79-98)*. Ankara: PEGEM Akademi.
- Tekinarslan, E., Çetin, İ. (2018). Bilişsel, Duyuşsal ve Sosyal Açından Programlama Öğretimi. Y. Gülbahar, H. Karal (Ed.), *Kuramdan Uygulamaya Programlama Öğretimi (ss. 38-65)*. Ankara: PEGEM Akademi.
- Tok, E. (2008). *Düşünme Becerileri Eğitimi Programının Okul Öncesi Öğretmen Adaylarının Eleştirel, Yaratıcı Düşünme Ve Problem Çözme Becerilerine Etkisinin İncelenmesi*. (Yayınlanmış doktora tezi). Marmara Üniversitesi Eğitim Bilimleri Enstitüsü, İstanbul.
- The Tech Partnership, & BCS. (2016). *Women in IT scorecard*. 1–56.
https://www.thetechpartnership.com/globalassets/pdfs/research-2016/womeninit_scorecard_2016.pdf adresinden 1 Ocak 2020 tarihinde edinilmiştir.
- Tutkun, Ö. F. (2012). Bloom'un Yenilenmiş Taksonomisi Üzerine Genel Bir Bakış. *Sakarya University Journal of Education*, 2(1), 14–22.
- Trees, F. P. (2010). *A meta-analysis of pedagogical tools used in introductory programming courses*. Doc- toral dissertation, Pace University.

- Vujošević-Janičić, M., ve Tošić, D. (2008). The role of programming paradigms in the first programming courses. *The Teaching of Mathematics*, XI, 2, 63–83. <http://elib.mi.sanu.ac.rs/files/journals/tm/21/tm1122.pdf> adresinden 24 Kasım 2019 tarihinde edinilmiştir.
- Waite, J. (2017). Pedagogy in teaching Computer Science in schools: A Literature Review. *The Royal Society*, 1–90. <https://royalsociety.org/~media/policy/projects/computing-education/literature-review-pedagogy-in-teaching.pdf> adresinden 24 Kasım 2019 tarihinde edinilmiştir.
- Wang, T.-C., Mei, W.-H., Lin, S.-L., Chiu, S.-K., ve Lin, J. M.-C. (2009). Teaching programming concepts to high school students with Alice. *2009 39th IEEE Frontiers in Education Conference*.
- Weintrop, D., ve Wilensky, U. (2015). To block or not to block, that is the question: Students' perceptions of blocks-based programming. *Proceedings of IDC 2015: The 14th International Conference on Interaction Design and Children*, 199–208. <https://doi.org/10.1145/2771839.2771860>
- Wing, J. M. 2006 Computational thinking. *Commun. ACM* 49, 33–35.
- Vu, S. (2017). *Cracking the code: Why aren't more women majoring in computer science*. <https://bit.ly/2V20pWN> adresinden 24 Kasım 2019 tarihinde edinilmiştir.
- Yalçın, S. (2018). 21. Yüzyıl Becerileri ve Bu Becerilerin Ölçülmesinde Kullanılan Araçlar ve Yaklaşımlar. *Ankara Üniversitesi Eğitim Bilimleri Fakültesi Dergisi*, 183–201. <https://doi.org/10.30964/auebfd.405860>.
- Yıldız Durak, H. (2018). The Effects of Using Different Tools in Programming Teaching of Secondary School Students on Engagement, Computational Thinking and Reflective Thinking Skills for Problem Solving. *Technology, Knowledge and Learning*, (0123456789). <https://doi.org/10.1007/s10758-018-9391-y>.

Yılmaz, U. (2018). *Eğitimde Ölçme ve Değerlendirme*.

<https://docplayer.biz.tr/129849541-Egitimde-olcme-ve-degerlendirme-ugur-yilmazer-1.html> adresinden 15 Aralık 2019 tarihinde edinilmiştir.

Yurdabakan, I. (2012). The Effects of Bloom' s Revised Taxonomy on Measurement and Evaluation in Education. *Gaziantep Üniversitesi Sosyal Bilimler Dergisi*, 11(2), 327–348.

Yükseltürk, E., Altıok, S. (2018). Blok Tabanlı Programlama. Y. Gülbahar (Ed.), *Bilgi İşlemsel Düşünmeden Programlamaya (ss. 241-263)*. Ankara: PEGEM Akademi.

Yükseltürk, E., Üçgöl, M. (2018). Blok Tabanlı Programlama. Y. Gülbahar, H. Karal (Ed.), *Bilgi İşlemsel Düşünmeden Programlamaya (ss. 273-296)*. Ankara: PEGEM Akademi.

Zeldin, A. L., ve Pajares, F. (2000). Against the odds: Self-efficacy beliefs of women in mathematical, scientific, and technological careers. *American Educational Research Journal*, 37(1), 215–246.
<https://doi.org/10.3102/00028312037001215>.

EKLER

A. 8. Hafta Bilişim Teknolojileri ve Yazılım Ders Planı

Tarih	27-31/05/2019
Sınıf	6 A-B-C-D-E-F-G
Süre	40 dakika
Öğrenme Alanı	Problem Çözme ve Programlama
Kazanımlar	JavaScript hazır fonksiyonları kullanır. JavaScript kodları ile fonksiyonları kullanır. Tüm programlama yapılarını içeren özgün bir proje oluşturur.
Öğretme-Öğrenme Yöntem ve Teknikleri	Sunuş Yöntemi, Soru-Cevap, Kodlama Taktiği, Gösterip Yaptırma, Uygulama, Bireysel Öğrenme Etkinlikleri
Kullanılan Eğitim Teknolojileri-Araç, Gereçler ve Kaynakça	24 adet bilgisayar, interaktif akıllı tahta, yardımcı kitap, Notepad++, Google Classroom
Giriş	
Öğretme-Öğrenme Etkinlikleri	
Dikkat çekme	Öğrencilerden 5 öğrenci seçilir. Öğretmenin hafızasından 1 ile 10 arasından rasgele bir sayı tuttuğu ve bu sayıyı sırası ile tahmin etmeye çalışmaları istenir.
Öğrenciyi hedeften haberdar etme	Programlama dillerinde de rastgele sayı tutma fonksiyonunun çok sık kullanıldığı ve JavaScript dilinde de bu özelliğin kullanıldığından bahsedilir.

Güdüleme	Derste bu özelliği kullanarak hem JavaScript dilindeki hazır fonksiyonları kullanarak hem de kendileri fonksiyon oluşturarak “Sayı Tahmini Oyunu” hazırlayacakları söylenir.
Gelişme Uyarıcıların Sunulması Öğrencilerden Notepad++ uygulamasını açmaları istenir. HTML kemik kodların hatırlanması ve yazılması istenir. Script kodlarının Body bölümüne yazılarak kodlar eklenir. Form elementlerinden text ve buton kullanılacağından bahsedilir. “random” komutunun rasgele sayı tutturmak için eklendiği belirtilir. Kavramların Tanıtılması “parseInt” fonksiyonunun iki parametre kullanarak bir seçim atadığından bahsedilir. JavaScript dilinde fonksiyonların çok kullanıldığı anlatılır. Çünkü programlamada çok sık kullanılan komutların sürekli yazılmasının doğru olmadığı, komutların ne kadar kısa tutulursa o kadar hızlı çalışacağından bahsedilir. Aynı zamanda zamandan kazanırken, çok fazla komut yazmaktan da kurtulmuş olduğu anlatılır.	

Etkinlik

Sayı Tahmini Oyunu

```
1
2
3 <!doctype html>
4
5 <html>
6   <head>
7     <title> Sayı Tahmini Oyunu </title>
8   </head>
9
10  <body>
11
12    <script type="text/javascript">
13
14      function SayiTahmini() {
15        var girilenSayi = document.getElementById("sayiId").value;
16
17        var TahminiSayi=Math.round(Math.random()*10);
18
19        if (girilenSayi<TahminiSayi)
20        {
21          document.getElementById("sonuc").innerHTML ="Daha büyük bir sayı giriniz.";
22        }
23        if (girilenSayi>TahminiSayi)
24        {
25          document.getElementById("sonuc").innerHTML ="Daha küçük bir sayı giriniz.";
26        }
27        if (girilenSayi==TahminiSayi)
28        {
29          document.getElementById("sonuc").innerHTML ="Tebrikler bildiniz. Tuttuğum sayı: "+ Tahmi
30        }
31      }
32
33    </script>
34
35    1-10 arasında bir sayı giriniz: <input type="text" id="sayiId">
36
37    <button onclick="SayiTahmini()">Gönder</button>
38
39    <p id="sonuc"></p>
40  </body>
41 </html>
42
43
44
```

Ölçme-Değerlendirme	Kontrol Çizelgesi, öğrencilerin web siteleri kontrol edilir.
Dönüt Sağlama	Öğrencilerin çalışmalarının doğruları için “harika, aferin” gibi dönütler verilir. Hataları olan öğrenciler için hatalarını düzelterek tekrar kayıt etmeleri istenir.
Nur Faden KARAÇAM DUMAN Bilişim Teknolojileri ve Yazılım Dersi Öğretmeni	

B. Bilgisayarca Düşünme Beceri Düzeyleri Ölçeği

Sevgili Öğrenciler,

Aşağıdaki maddeler bilgisayarca düşünme becerilerini ölçmeye dönük hazırlanmış ve bir araştırmada kullanılacaktır. Araştırma dışında başka hiçbir amaçla kullanılmayacaktır. Lütfen her bir maddeyi dikkatle okuyup, sizi yansıtan düzeyini en olumludan (5) en olumsuz (1) doğru puanlayınız. Katılımınızdan dolayı şimdiden teşekkür ederiz.

Nur Faden Karaçam Duman

C1	Kararlarının çoğundan emin olan insanları severim	1	2	3	4	5
C4	Yeni bir durumla karşılaştığımda ortaya çıkabilecek sorunları çözebileceğime inancım vardır.	1	2	3	4	5
C5	Bir sorunumu çözmek üzere plan yaparken o planı yürütebileceğime güvenirim.	1	2	3	4	5
C8	Bir sorunla karşılaştığımda, başka konuya geçmeden önce durur ve o sorun üzerinde düşünürüm.	1	2	3	4	5
A1	Bir problemin çözümünü verecek denklemi hemen kurabilirim	1	2	3	4	5
A3	Matematiksel sembol ve kavramlar yardımıyla yapılan anlatımları daha kolay öğrendiğimi düşünürüm	1	2	3	4	5
A4	Sayılar arasındaki ilişkileri kolaylıkla yakalayabildiğime inanırım	1	2	3	4	5
A6	Sözel olarak ifade edilen bir matematik problemini sayısallaştırabilirim.	1	2	3	4	5
O1	Grup arkadaşlarımla birlikte işbirlikli öğrenme deneyimleri yaşamaktan hoşlanırım.	1	2	3	4	5
O2	İşbirlikli öğrenmede, grupla çalıştığım için daha başarılı sonuçlar elde ettiğimi/edeceğimi düşünüyorum.	1	2	3	4	5
O3	İşbirlikli öğrenmede grup arkadaşlarımla birlikte grup projesi ile ilgili problemleri çözmekten hoşlanırım.	1	2	3	4	5
O4	İşbirlikli öğrenmede daha çok fikir ortaya çıkıyor.	1	2	3	4	5
T1	Karmaşık problemlerin çözümüne yönelik düzenli planlar geliştirmede iyiyimdir.	1	2	3	4	5
T2	Karmaşık problemleri çözmeye çalışmak eğlencelidir.	1	2	3	4	5
T3	Zorlayıcı şeyler öğrenmeye istekliyimdir.	1	2	3	4	5
T5	Elimdeki seçenekleri karşılaştırırken ve karar verirken kullandığım sistematik bir yöntem vardır.	1	2	3	4	5
P1	Problemin çözümünü zihnimde canlandırma konusunda sıkıntı yaşarım.	1	2	3	4	5
P2	Problem çözümünde X, Y gibi değişkenleri nerede ve nasıl kullanmam gerektiği konusunda sıkıntı yaşarım.	1	2	3	4	5
P3	Tasarladığım çözüm yollarını sırasıyla aşamalı bir şekilde uygulayamam.	1	2	3	4	5
P4	Bir soruna yönelik olası çözüm yollarını düşünürken çok fazla seçenek üretemem.	1	2	3	4	5
P5	İşbirlikli öğrenme ortamında kendi düşüncelerimi geliştiremem.	1	2	3	4	5
P6	İşbirlikli öğrenme grup arkadaşlarıma bir şeyler öğretmeye çalışmak beni yoruyor.	1	2	3	4	5

C. Blok Temelli Programlamaya İlişkin Öz-Yeterlilik Algısı Ölçeği

Sevgili Öğrenciler,

Bu ölçek sizin Blok temelli programlamaya yönelik kendinize ilişkin öz-yeterlilik algınızı belirlemek üzere hazırlanmıştır. Araştırmaya yönelik katkınızdan dolayı teşekkür ederiz.

Uygun olan seçenekleri tıklayarak işaretleyiniz.

Nur Faden Karaçam Duman

1. E-posta adresi *

2. Adınız Soyadınız *

3. Sınıfınız *

Yalnızca bir şıkkı işaretleyin.

- 6A
 6B
 6C
 6D
 6E
 6F
 6G

4. Cinsiyetiniz *

Yalnızca bir şıkkı işaretleyin.

- Kız
 Erkek

5. Evinizde bilgisayar var mı? *

Yalnızca bir şıkkı işaretleyin.

- Evet
 Hayır
 Var, ama kullanmama izin verilmiyor veya bilgisayar arızalı olduğu için çalışmıyorum

6. Evinizde internet bağlantınız var mı? *

Yalnızca bir şıkkı işaretleyin.

- Evet
 Hayır
 Evet var, ama kullanmama izin verilmiyor veya bilgisayar arızalı olduğu için internete bağlanamıyorum

7. Evde Scratch programı ile uygulama yapmaya devam ediyor musunuz? *

Yalnızca bir şıkkı işaretleyin.

- Evet
 Hayır

8. Scratch uygulaması ile programlama yapmayı seviyor musunuz? *

Yalnızca bir şıkkı işaretleyin.

- Evet
 Kismen
 Hayır

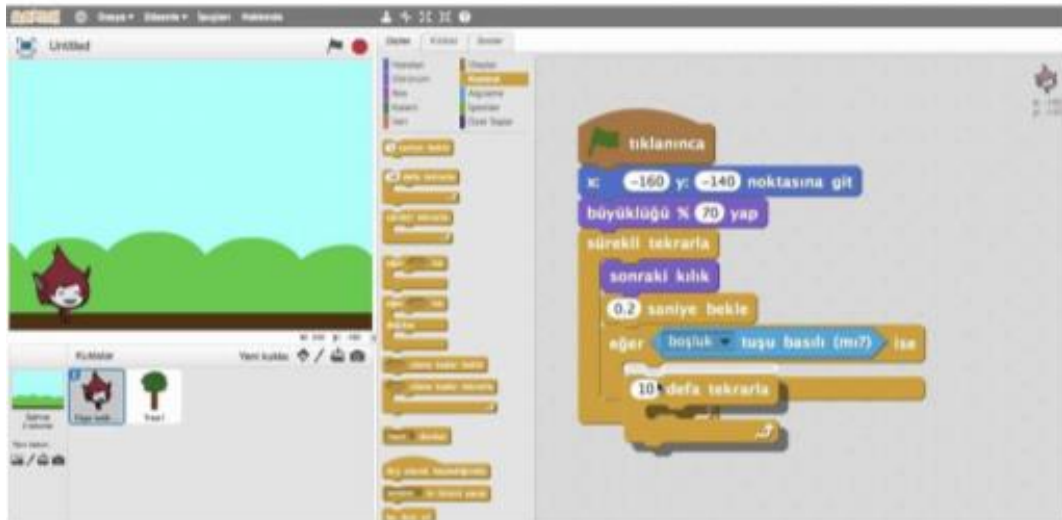
1. Bölüm

Aşağıdaki Scratch programına ilişkin verilen görevleri yaparken kendinize olan güveninizi 1 ile 5 arasında derecelendirerek belirtiniz. Anlamadığınız soru olursa boş bırakınız.

- 1 - Hiç Güvenmiyorum
2 - Biraz Güveniyorum
3 - %50 / %50
4 - Oldukça Güveniyorum
5 - Tamamen Güveniyorum

Scratch programını kullandığınız zamanları düşünerek, aşağıdaki ifadeleri okuyunuz. Scratch programını, yazılar bloğunu hatırlamanız gayesiyle ekran görünümü sunulmuştur.

Scratch Ekran Görünümü - Diziler (Kodlar) Bloğu Orta Bölümde Açıktır



9. **1. Scratch'te yazılmış bir program (kodlar) gördüğümde, çalıştırıldığında neler olacağını söyleyebilirim.**

Yalnızca bir şıkkı işaretleyin.

	1	2	3	4	5	
Hiç Güvenmiyorum	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Tamamen Güveniyorum

10. **2. Başkası tarafından hazırlanan bir programı (kodları) okuyup anlayabilirim.**

Yalnızca bir şıkkı işaretleyin.

	1	2	3	4	5	
Hiç Güvenmiyorum	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Tamamen Güveniyorum

11. **3. Bir kuklaya herhangi bir hareket vermek istediğimde, Scratch'te bunu nereden yapabileceğimi bilirim.**

Yalnızca bir şıkkı işaretleyin.

	1	2	3	4	5	
Hiç Güvenmiyorum	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Tamamen Güveniyorum

12. **4. Sahnedeki kuklayı istediğim hızda hareket ettirebilirim.**

Yalnızca bir şıkkı işaretleyin.

	1	2	3	4	5	
Hiç Güvenmiyorum	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Tamamen Güveniyorum

13. **5. Sahnedeki kuklayı sürekli hareket ettirebilirim.**

Yalnızca bir şıkkı işaretleyin.

	1	2	3	4	5	
Hiç Güvenmiyorum	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Tamamen Güveniyorum

14. **6. Scratch'te bir kuklanın görünümünü (kostüm, renk, boyut, konuşma gibi) bir koşula bağlı olarak (örneğin: eğer ise) değiştirebilirim.**

Yalnızca bir şıkkı işaretleyin.

	1	2	3	4	5	
Hiç Güvenmiyorum	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Tamamen Güveniyorum

15. **7. Scratch'te bir kuklanın hareketini (hızı, yönü, konumu gibi) bir koşula bağlı olarak (örneğin: eğer ise) değiştirebilirim.**

Yalnızca bir şıkkı işaretleyin.

	1	2	3	4	5	
Hiç Güvenmiyorum	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Tamamen Güveniyorum

16. **8. Bir oyunda kullanıcının elde ettiği puan değerinin tutulacağı bir değişken oluşturabilirim.**

Yalnızca bir şıkkı işaretleyin.

	1	2	3	4	5	
Hiç Güvenmiyorum	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Tamamen Güveniyorum

17. **9. Bir oyunda istenilenler başarılıldıkça "Puan" veya "Skor" değerinin arttığı veya azaldığı bir program hazırlayabilirim.**

Yalnızca bir şıkkı işaretleyin.

	1	2	3	4	5	
Hiç Güvenmiyorum	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Tamamen Güveniyorum

18. **10. İstenilenler açıkça tanımlandığında oldukça karmaşık ve uzun kodlardan oluşan bir oyun hazırlayabilirim.**

Yalnızca bir şıkkı işaretleyin.

	1	2	3	4	5	
Hiç Güvenmiyorum	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Tamamen Güveniyorum

19. **11. Scratch'te hazırlanan bir programdaki hataları bulabilirim.**

Yalnızca bir şıkkı işaretleyin.

	1	2	3	4	5	
Hiç Güvenmiyorum	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Tamamen Güveniyorum

20. **12. Scratch'te hazırlanan bir programdaki hataları düzeltip çalışabilir hale getirebilirim.**

Yalnızca bir şıkkı işaretleyin.

	1	2	3	4	5	
Hiç Güvenmiyorum	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Tamamen Güveniyorum

D. Metin Temelli Programlama Başarı Testi

6. SINIF METİN TEMELLİ PROGRAMLAMA BAŞARI TESTİ

1. “Sayfa içindeki metinlerin tarayıcıda nasıl görüneceğini ve işlevsel olacağını düzenleyen programlama dili _____ dir.”

Bu parçada boş bırakılan yerlere aşağıdakilerden hangisi sırasıyla getirilmelidir?

- a. JavaScript
- b. XML
- c. Java
- d. HTML

```
<!doctype html>  
<html lang="tr">  
<meta charset="utf-8">
```

2. **Aşağıdakilerden hangisi, yukarıda yer alan komut satırlarından birinin görevidir?**

- a. Sayfa ile ilgili açıklama bulundurur.
- b. Web sitesinin konu başlığını gösterir.
- c. Sayfanın HTML5 ile tasarlandığını belirtir.
- d. Sayfanın başlangıcı ve sonudur.



Sayfama Hoşgeldiniz

3. Yukarıdaki görselde yer alan web sayfasının ad ve başlığının oluşmasını sağlayan komutlar, aşağıdaki seçeneklerin hangisinde doğru olarak verilmiştir?

a. <!DOCTYPE html>

```
<html>
  <head>
    <title> Sayfama Hoşgeldiniz </title>
  </head>
  <body>
    <h1> Sayfama Hoşgeldiniz </h1>
  </body>
</html>
```

b. <!DOCTYPE html>

```
<html>
  <head>
    <title> Sayfama Hoşgeldiniz </title>
  </head>
  <body>
  </body>
</html>
```

c. <!DOCTYPE html>

```
<html>
  <head>
    <title> Sayfama Hoşgeldiniz </title>
  </head>
</html>
```

d. <!DOCTYPE html>

```
<html>
  <body>
    <h1> Sayfama Hoşgeldiniz </h1>
  </body>
</html>
```

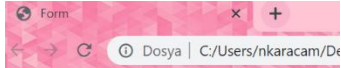
4. Bir web sayfasında görüntülenmek istenen içerikler, aşağıdaki etiketlerin hangisinin içerisinde yer almalıdır?

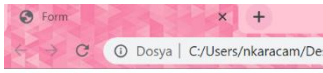
- a. <html> ... </html>
- b. <body> ... </body>
- c. <head> ... </head>
- d. <script> ... </script>


```
7 <!DOCTYPE html>
8 <html>
9 <head>
10 <title> Form </title>
11 </head>
12 <body>
13 <form name="email" enctype="text/plain"
14 action="mailto:mail@sunucu.com?subject="Siteden Email">
15 Ad:<input type="text" name="ad" required="required">
16 <br>
17 Soyad:<input type="text" name="soyad" required="required">
18 </form>
19 </body>
20 </html>
```

5. Yukarıdaki komutlarla oluşturulan web sayfası, aşağıdakilerden hangisidir?

a. 
Ad Soyad :
Email :

b. 
Ad Soyad :

c. 
Ad Soyad :
Email :
Gönder

d. 
Email :

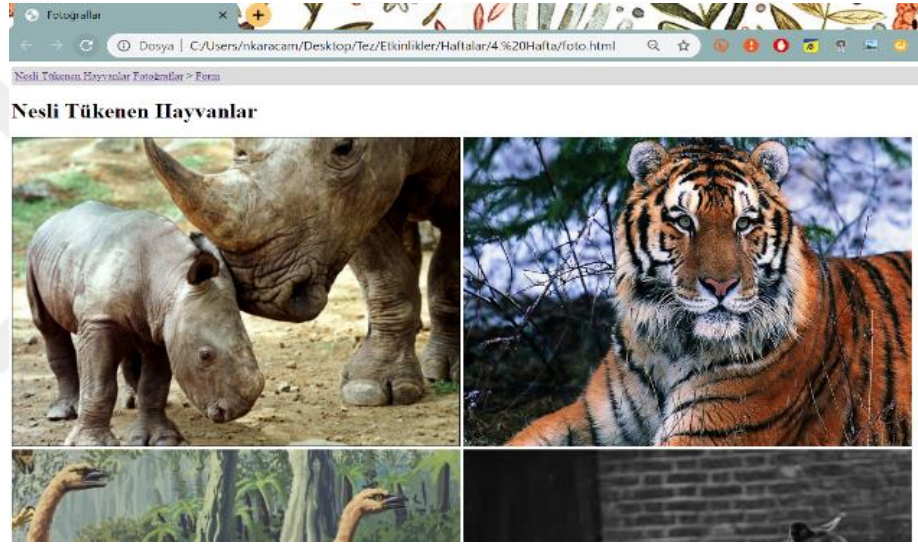
6. Link (bağlantı) oluşturmak için kullanılan etiket (komut) ---- dir. Yukarıdaki cümlede boş bırakılan yere, aşağıdaki HTML etiket komutundan hangisi getirilmelidir?

a.

b.

c. <a href>

d. <p>



*Şekilde yer alan web sayfasını inceleyiniz.

7. Aşağıdaki HTML komutlarından hangisi, web sayfasına ait fotoğraflardan herhangi birini eklemek için kullanılabilir?

a.

b.

c. <img="f1.jpg" style="width: 640px; height: 480px ">

d.

8. Yukarıdaki açıklamaya göre, JavaScript komutlarını HTML sayfaları içerisinde kullanabilmek için script komutlarının aşağıdaki etiketlerden hangisine yerleştirilmesi gerekir?

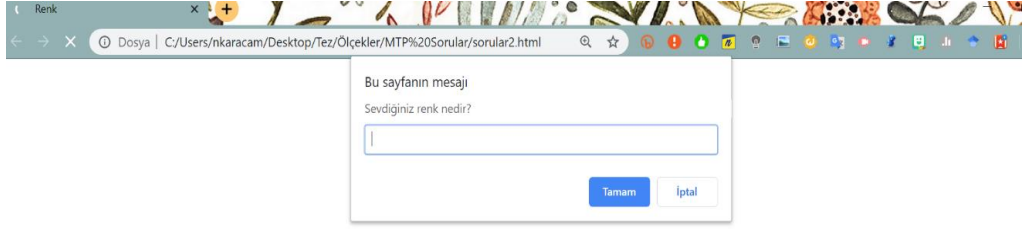
- a. <html>
- b. <head>
- c. <body>
- d. <title>

9. JavaScript komutları ile kullanıcıdan / ziyaretçiden bilgi alabilmek için aşağıdaki komutlardan hangisi kullanılabilir?

- a. prompt()
- b. alert()
- c. innerHTML()
- d. document.write()

10. Bir web sayfasının “div” etiketinin arka plan rengini değiştirmek için aşağıdaki komutlardan hangisi kullanılmalıdır?

- a. <div style= back-color: “red”>
- b. <div style= “back-color: red”>
- c. <div style= “background-color: red”>
- d. <div background-color: “red”>



11. Yukarıdaki web sayfasına göre, uygulamanın çalışabilmesi için aşağıda verilen komut sıralamasından hangisi doğru olarak verilmiştir?

a. `<script>`

```
var renk;  
renk=prompt("Sevdiğiniz renk nedir?");  
</script>
```

b. `<script>`

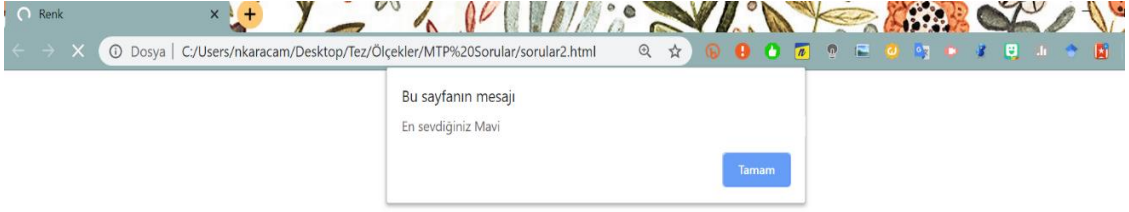
```
var renk;  
renk=alert("Sevdiğiniz renk nedir?");  
</script>
```

c. `<script>`

```
var renk;  
renk=prompt("Sevdiğiniz renk nedir?");  
</script>
```

d. `<script>`

```
var renk  
renk=prompt("Sevdiğiniz renk nedir?")  
</script>
```



* Şekilde yer alan web sayfasını inceleyiniz.

12. Yukarıda yer alan web sayfasındaki veri çıkışının çalışmasını sağlayan komut, aşağıdakilerden hangisidir?

- a. alert("En sevdiğiniz "+renk);
- b. prompt("En sevdiğiniz "+renk);
- c. innerHTML("En sevdiğiniz "+renk);
- d. document.write("En sevdiğiniz "+renk);

- I. var sayi1, sayi2, sonuc;
- II. sayi1 = prompt("1. Sayıyı giriniz: ");
- III. sayi2 = Number(prompt("2. Sayıyı giriniz: "));
- IV. sonuç = (sayi1+sayi2) / 2;
- V. alert("Sonuç: " + sonuc);

*13 ve 14. soruları yukarıdaki komut satırlarına göre yanıtlayınız.

13. Yukarıda verilmiş "script" komutlarında bazı hatalar vardır. Hatalı olan satırlar, aşağıdaki seçeneklerin hangisinde doğru olarak verilmiştir?

- a. II-V
- b. I-IV
- c. II-IV
- d. III-V

14. Yukarıda yer alan komut satırları, aşağıdaki işlemlerden hangisini yapmaktadır?

- a. İki sayının karesini hesaplama
- b. İki sayının ortalamasını hesaplama
- c. İki sayının toplamını hesaplama
- d. İki sayının farkını hesaplama

```
var not;  
if (not<=50) {  
    alert("Notunuz zayıf.");  
} else {  
    alert("Notunuz yüksek.");  
}
```

15. Yukarıdaki script komutlarına göre aşağıdaki veri çıkışlarından hangisi yanlıştır?

- a. Eğer not=58 ise, "Notunuz yüksek." yazar.
- b. Eğer not=41 ise, "Notunuz zayıf." yazar.
- c. Eğer not=88 ise, "Notunuz yüksek." yazar.
- d. Eğer not=55 ise, "Notunuz zayıf." yazar.

```
<script>  
var yas, yil, bilet, cevap;  
  
cevap=prompt("Konser bileti almak ister misiniz?");  
  
if(cevap == "evet") {  
yil=prompt("Lütfen doğum yılınızı giriniz: ");  
yas=2019-(yil);  
  
    if(yas>18){  
alert("Sitemize hoş geldiniz!");  
    }  
  
    else{  
alert("Lütfen ebeveyn ile giriş yapınız.");  
    }  
}  
  
else{  
cevap=alert("Görüşmek üzere!");  
}  
  
</script>
```

* 16 ve 17. soruları yukarıdaki görsele göre yanıtlayınız.

16. Yukarıda verilen script komutlarına göre aşağıdakilerden hangisi yanlıştır?

- a. Veri girişi, “Doğum yılınız 2002” ise veri çıkışı, “Lütfen, ebeveyn ile giriş yapınız!” olur.
- b. Veri girişi, “Doğum yılınız 2011” ise veri çıkışı, “Sitemize hoş geldiniz!” olur.
- c. Veri girişi, “Doğum yılınız 2005” ise veri çıkışı, “Lütfen, ebeveyn ile giriş yapınız!” olur.
- d. Veri girişi, “Doğum yılınız 1991” ise veri çıkışı, “Sitemize hoş geldiniz!” olur.

- i. Doğum yılının girişi
- ii. Yaşın hesaplanması
- iii. Bilet isteme sorusunun cevabının kontrol edilmesi
- iv. Yaşın kontrol edilmesi

17. Görseldeki komutlara göre, script komutlarının işleyiş sıralaması aşağıdaki seçeneklerin hangisinde doğru olarak verilmiştir?

- a. iii-ii-i-iv
- b. iv-i-iii-ii
- c. iii-i-ii-iv
- d. iv-ii-i-iii

18. Script komutlarını kullanarak 1’den 12’ye kadar ardışık sıralama yapan komut, aşağıdakilerden hangisidir?

- a. `for(i=1; i<=12 ;i++)`
- b. `for(i=0; i<=12 ;i++)`
- c. `for(i=1; i<=13 ;i++)`
- d. `for(i=0; i<13 ;i++)`

```
function saha () {  
    .....  
    var kkenar, ukenar, alan;  
  
    kkenar=Number (document.getElementById ("Genişlik:").value) ;  
    ukenar=Number (document.getElementById ("Yükseklik:").value) ;  
  
    alan=ukenar*kkenar;  
    document.getElementById ("Sonuc").innerHTML="alan="+alan;  
}
```

***19, 20 ve 21. soruları yukarıdaki görsele göre yanıtlayınız.**

19. Yukarıdaki fonksiyona göre “alanın hesaplanmış olan değerini ekrana yazdıran komut” aşağıdakilerden hangisidir?

- a. document.getElementById("Sonuc").innerHTML="alan="+alan;
- b. kkenar=Number(document.getElementById("Genişlik:").value);
- c. alan=ukenar*kkenar;
- d. ukenar=Number(document.getElementById("Yükseklik:").value);

20. Yukarıdaki fonksiyona göre “değişkenleri tanımlayan komut” aşağıdakilerden hangisidir?

- a. kkenar=Number(document.getElementById("Yükseklik:").value);
- b. document.getElementById("Sonuc").innerHTML="alan="+alan;
- c. alan=ukenar*kkenar;
- d. var kkenar, ukenar, alan;

21. Yukarıdaki fonksiyona göre “girilen değerleri okuyan komut” aşağıdakilerden hangisidir?

- a. alan=ukenar*kkenar;
- b. var kkenar, ukenar, alan;
- c. kkenar=Number(document.getElementById("Yükseklik:").value);
- d. document.getElementById("Sonuc").innerHTML="alan="+alan;

E. Metin Temelli Programlama Başarı Testi Belirtke Tablosu

6. SINIF BİLİŞİM TEKNOLOJİLERİ VE YAZILIM DERSİ		BİLİŞSEL ALAN					
KAZANIMLAR		HATIRLATMA	ANLAMA	UYGULAMA	ANALİZ	DEĞERLENDİRME	YARATMA
1. Metin tabanlı programlama aracının arayüzünü ve özelliklerini tanımlar.	1,2	*					
2. Metin tabanlı programlama aracında sunulan bir programın işlevlerini açıklar.	1,2,3		*				
3. Yazım kriterlerine uygun temel HTML etiketlerini kullanır.	3,4,7,10,11,13			*			
4. Doğrusal mantık yapısını içeren programlar oluşturur.	3,5,11						*
5. Web tasarım editörüne uygun ara yüzü kullanır.	2,5,7,11,12,16,17,19,20,21			*			
6. Web tasarım editörüne uygun metin işlemlerini kullanır.	3,4,5			*			
7. Web tasarım editörüne uygun tablo işlemlerini yapar.	5						*
8. HTML komut yazım kriterlerine uygun formlar oluşturur.	5						*
9. HTML komut yazım kriterlerine uygun bağlantılar oluşturur.	6						*
10. HTML komut yazım kriterlerine uygun çerçeveler oluşturur.	5,7						*
11. Stil şablonunun(CSS) yapısına uygun CSS yapısı oluşturur.	10						*

12. Web tasarım editörüne uygun resim işlemlerini yapar.	7			*			
13. Web tasarım editörüne uygun çoklu ortam işlemlerini yapar.	10			*			
14. Problemin çözümü için bir algoritma geliştirir.	13,14,15						*
15. Sabitleri ve değişkenleri problem çözümünde kullanır.	11,13,14,15,16,17,20			*			
16. JavaScript kodlarını Html kodları içinde kullanır.	8,16,17,19,20,21			*			
17. JavaScript kodları ile karar yapılarını kullanır.	15,16,17,19,20,21			*			
18. Bir problemi alt problemlere ayrıştırır.	15,16,17				*		
19. JavaScript kodları ile Html bileşenlerini kullanır.	11,12						*
20. JavaScript kodları ile değişkenleri ve mantıksal işlem operatörlerini kullanır.	13,14						*
21. JavaScript kodları ile çoklu karar yapıları içeren programlar oluşturur.	15,16,17						*
22. Bir algoritmayı uyarlamak için en uygun karar yapılarını seçer.	16,17			*			
23. JavaScript kodları ile döngü yapılarını kullanır.	18			*			
24. JavaScript hazır fonksiyonları kullanır.	19,20,21			*			

25. JavaScript kodları ile fonksiyonları kullanır.	19,20,21			*			
26. Tüm programlama yapılarını içeren özgün bir proje oluşturur.	16,17,19,20,21						*

F. Öğrenci Görüşme Formu

BÖLÜM 1: Ders içi uygulama ve etkinliklere dair genel görüşme

1. Bilişim Teknolojileri ve Yazılım dersinde programlama (kodlama) etkinlikleri nasıldı?
2. Metin temelli programlamada en çok hoşuna giden neydi?
3. Metin temelli programlamada zor veya sıkıcı gelen şeyler var mıydı?
4. Metin temelli programlama etkinlikleri seni motive etti mi?
5. Metin temelli programlamaya evde devam eder misin?

BÖLÜM 2: Bilgisayarca düşünme beceri düzeylerine yönelik görüşme

1. Metin temelli programlamanın yaratıcılığına katkısı olduğunu düşünüyor musun? Neden?
2. Metin temelli programlamanın algoritmik düşünme becerine etkisi olduğunu düşünüyor musun? Neden?
3. Metin temelli programlamanın işbirlikli çalışma becerine katkısı olduğunu düşünüyor musun? Neden?
4. Metin temelli programlamanın eleştirel düşünme becerine etkisi olduğunu düşünüyor musun? Neden?
5. Metin temelli programlamanın problem çözme becerine katkısı olduğunu düşünüyor musun? Neden?

BÖLÜM 3: Akademik başarıya yönelik görüşme

1. Metin temelli programlama etkinlikleri süresince hangi yeni kavramları öğrendin?
2. Bir önceki etkinlikte öğrendiğin kavramları bir sonrakinde kullandığın oldu mu?
3. Metin temelli programlamanın Bilişim Teknolojileri ve Yazılım dersindeki başarına katkısı olduğunu düşünüyor musun, neden?


BÖLÜM 4: Blok temelli öz-yeterlilik algılarına yönelik görüşme

1. Scratch programında öğrenmiş olduğun kavramları metin temelli programlamada kullandığın oldu mu, hangilerini hatırladın?
2. Scratch programı ile programlamayı daha önceden bilmenin sana katkısı olduğunu düşünüyor musun?

G. Bilgisayarca Düşünme Beceri Düzeyleri Ölçeği İzin Maili

Bilgisayarca Düşünme Beceri Düzeyleri Ölçeği'ni geliştiren araştırmacılara gönderilen mail aşağıdaki gibidir:

Bilgisayarca Düşünme Ölçeği Kullanım İzni Gelen Kutusu x

 **Nur Faden Karaçam** <nurfaden@gmail.com>
Alıcı: myozden, Burak, s.sevdakucuk


18 Şubat Pzt 12:21

Merhabalar,

Ben Nur F. Karaçam Duman, Bahçeşehir Üniversitesi Eğitim Teknolojisi bölümünde yüksek lisans öğrencisiyim. İstanbul Üniversitesi'nde Yrd.Doç.Dr. Burak ŞİŞMAN danışmanlığında Ortaokul öğrencilerinde metin temelli programlamaya yönelik problem çözüme için bilgisayarca düşünmenin etkilerini araştırdığım tezimde sizin geliştirdiğiniz "Bilgisayarca Düşünme Ölçeği"ni kullanmak istiyorum.

Hazırlamış olduğunuz ölçeği izninizle kullanabilir miyim?
İyi çalışmalar dilerim.

Nur Faden Karaçam
Bilişim Teknolojileri Öğretmeni ve Eğitim Teknolojileri Uzmanı
www.nurfaden.com

 **M. Yasar Ozden** <myozden@gmail.com>
Alıcı: Özgen, Özgen, recep, ben

18 Şubat Pzt 12:40

Sayın Nur Faden Karaçam;
Öncelikle mesajınız teşekkür ederim. Ölçeğimizi kullanmak istemenizden mutluluk duyarız. Çalışmanız tamamlandığında paylaşımlarınızdan haberdar olmak isteriz.

Saygılarımla
M. Yaşar Özden

Saygılarımla,

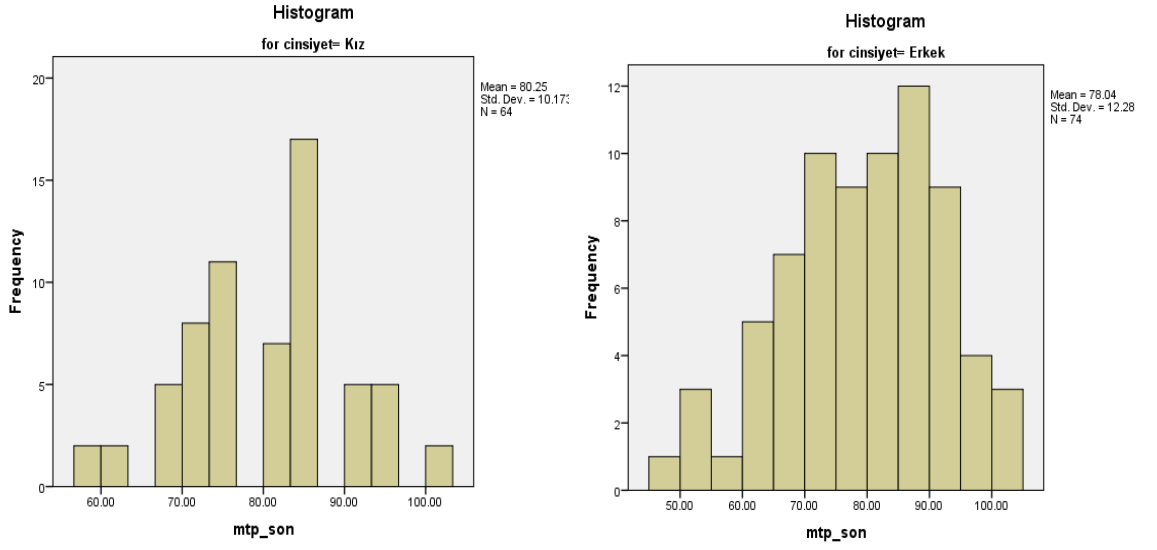
Prof. Dr. M. Yaşar Özden,
Müdür
Uzaktan Eğitim Enstitüsü
Doğu Akdeniz Üniversitesi,
Gazi Magusa
KKTC

H. Blok Temelli Programlamaya İlişkin Öz-Yeterlilik Algısı Ölçeği İzin Maili

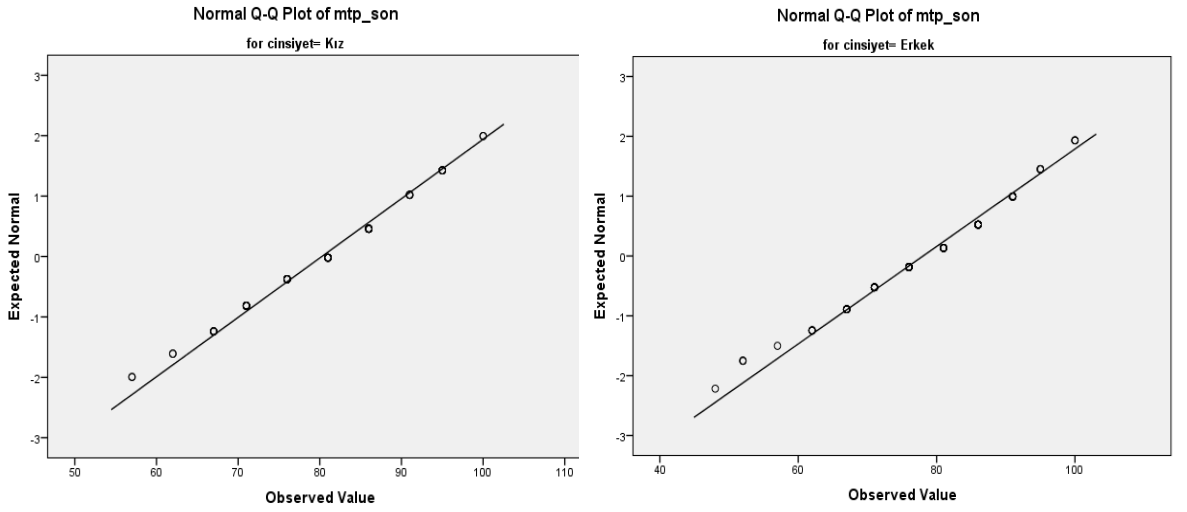
Blok Temelli Programlamaya İlişkin Öz-Yeterlilik Algısı Ölçeği'ni geliştiren araştırmacılara gönderilen mail aşağıdaki gibidir:



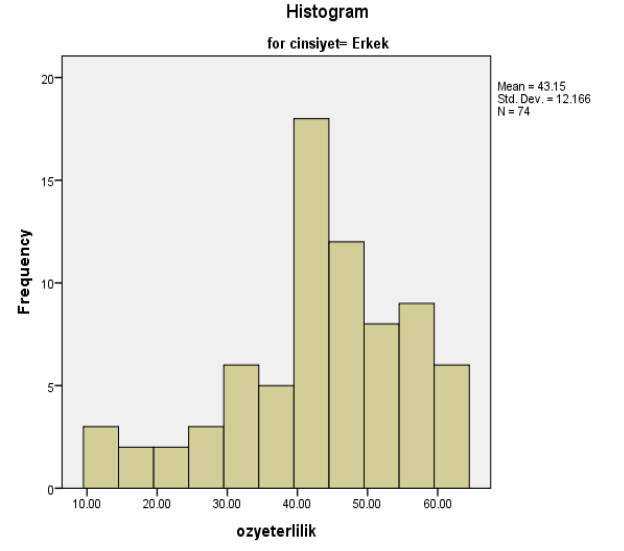
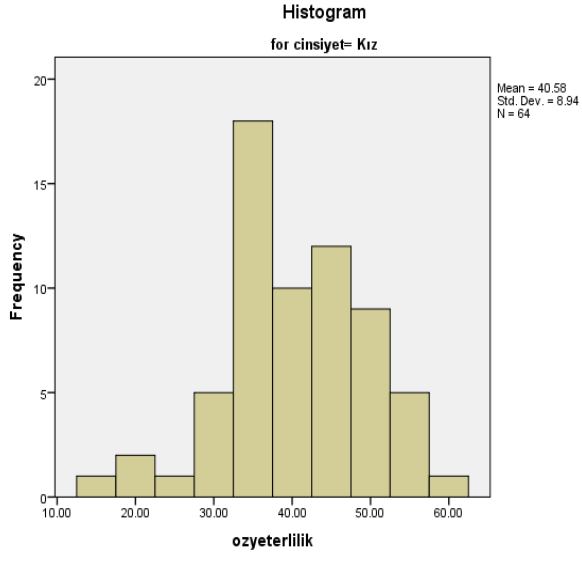
I. Histogram ve Normal Kantil Grafikleri



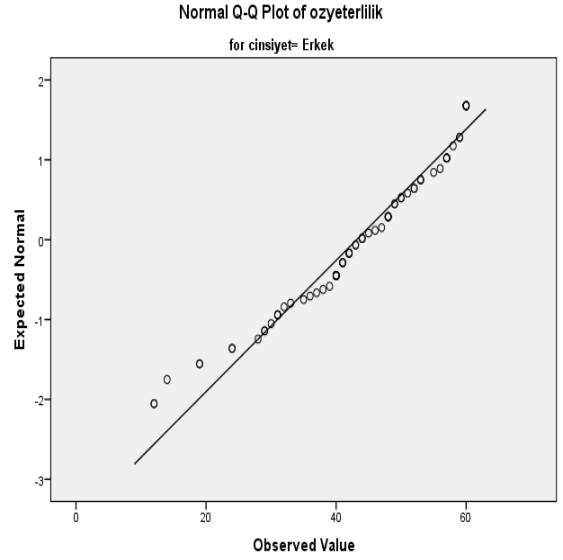
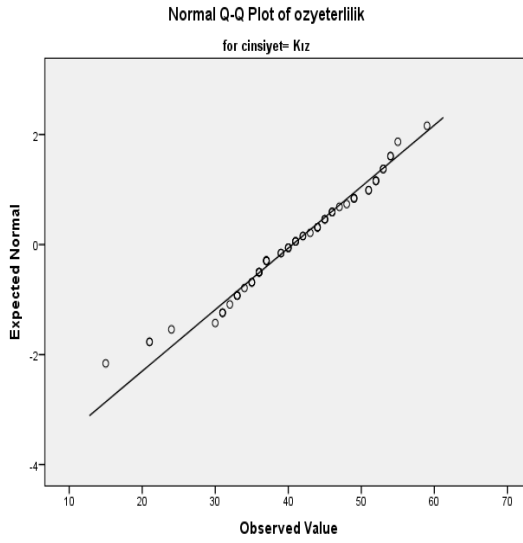
Şekil 8.5. MTPBT'nin cinsiyete göre histogram grafiği.



Şekil 8.6. MTPBT'nin cinsiyete göre normal kantil grafiği.

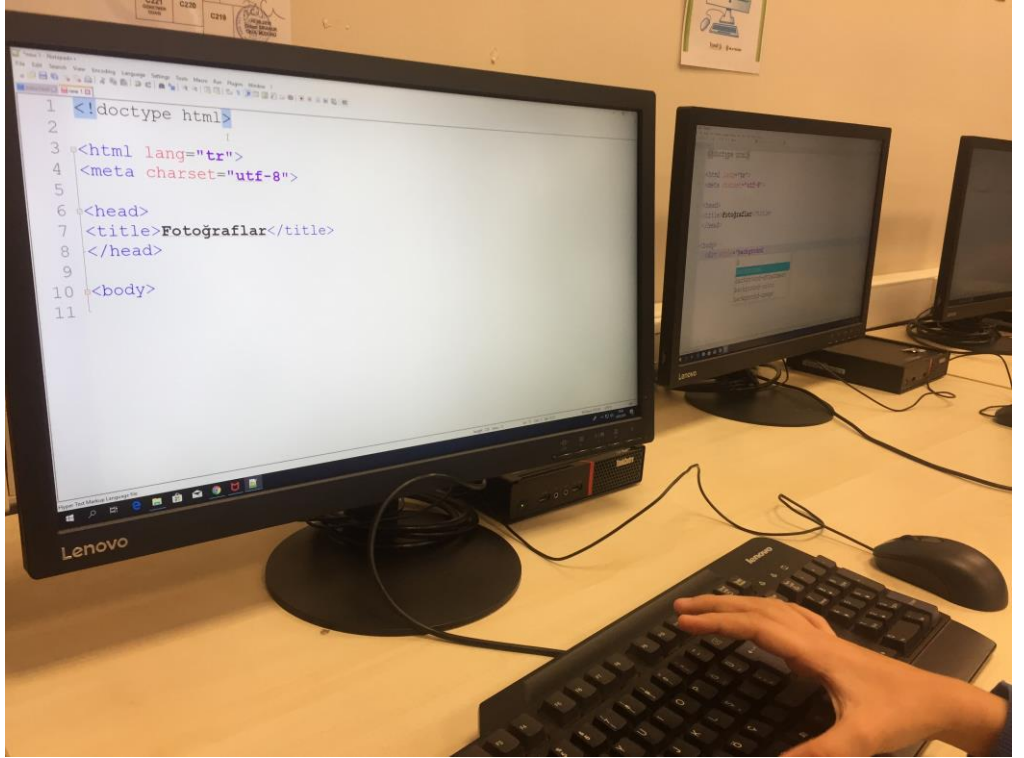


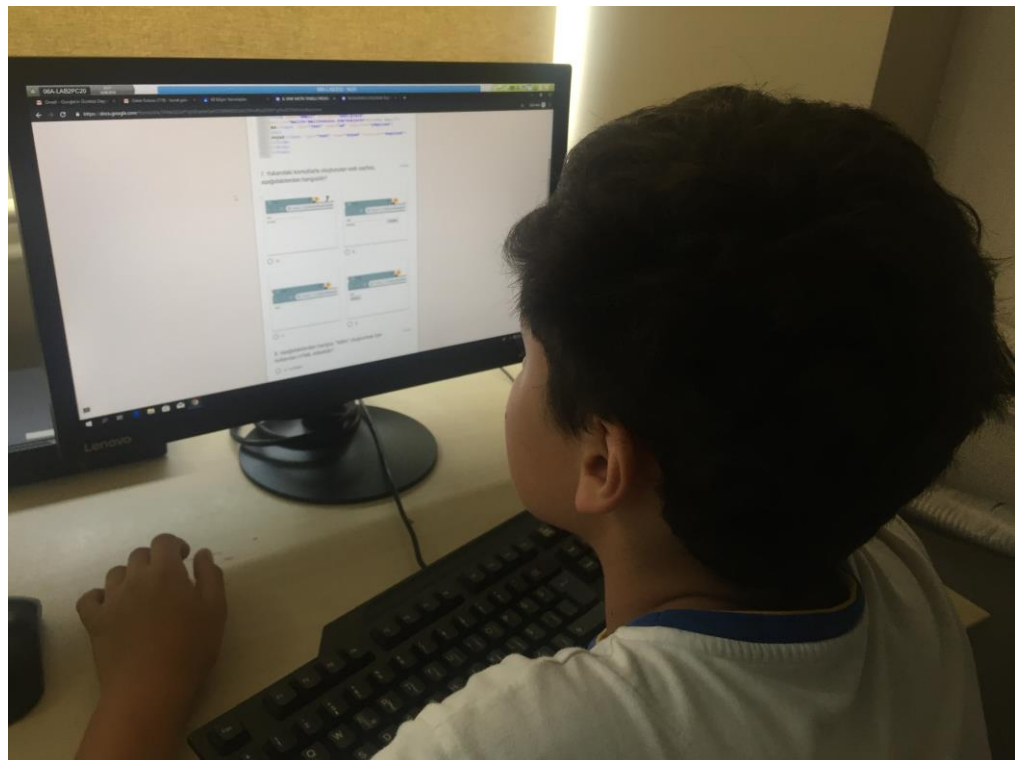
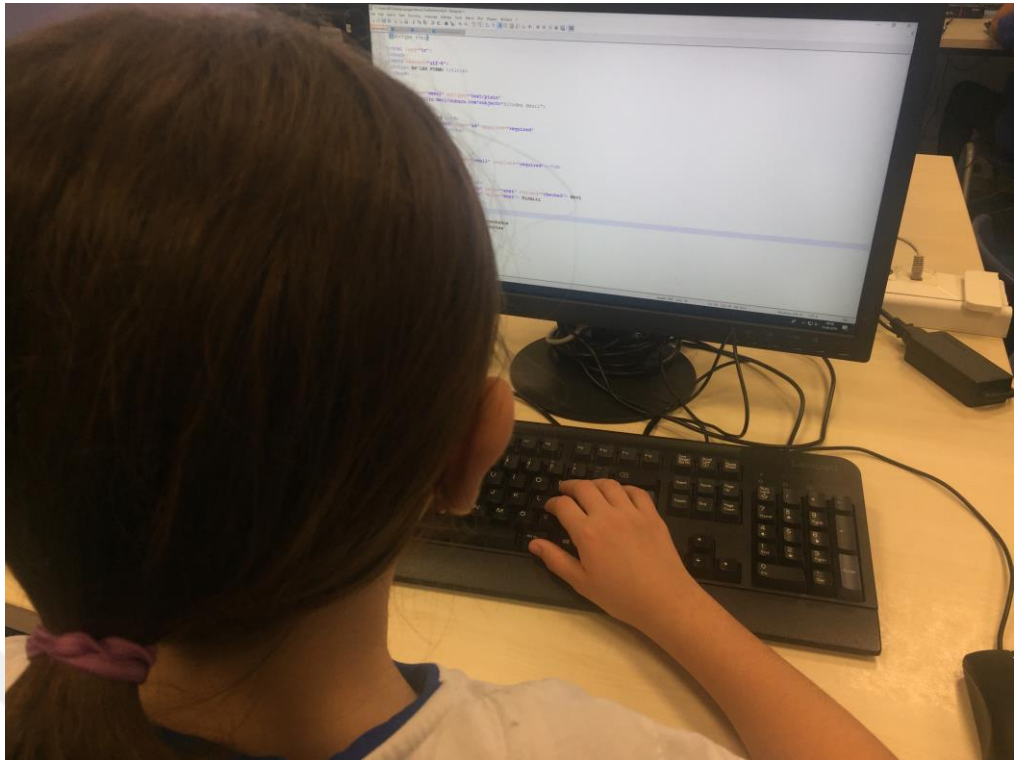
Şekil 8.7. BTPOT'nin cinsiyete göre histogram grafiği.

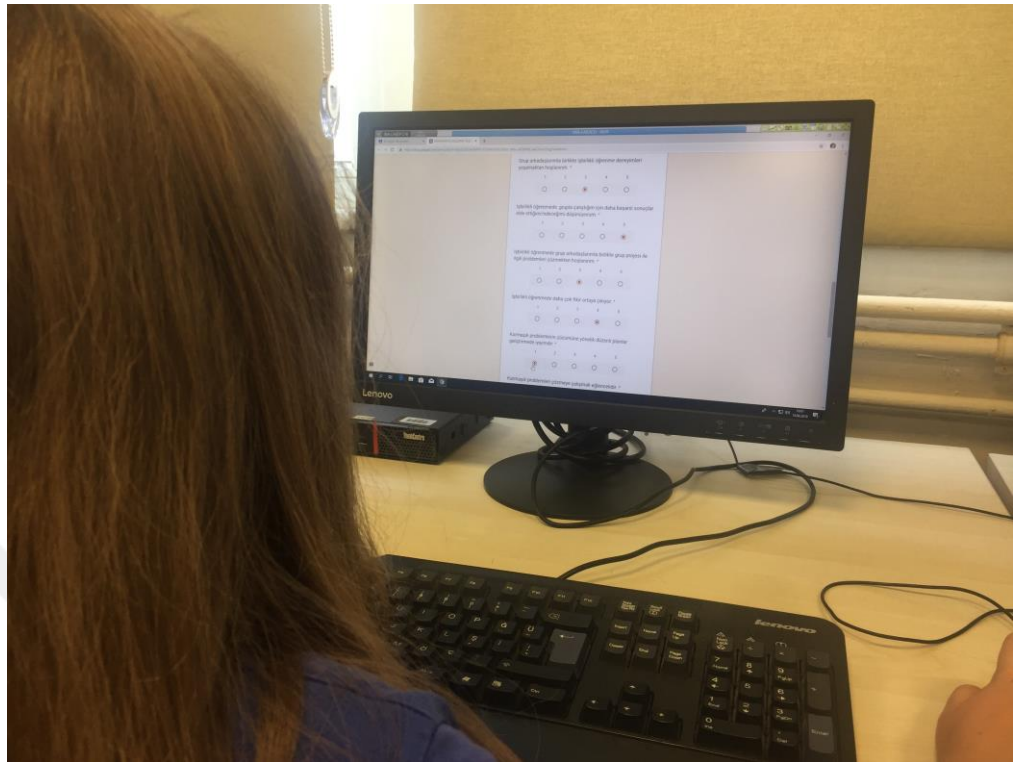


Şekil 8.8. BTPOT'nin cinsiyete göre normal kantil grafiği.

K. Araştırmaya İlişkin Fotoğraflar







L. Ders Etkinliklerine İlişkin Fotoğraflar



```
2 <!DOCTYPE html>
3 <html>
4 <head>
5   <title>Fotoğraflar</title>
6 </head>
7 <body>
8   <div style="background-color: #DCDCDC; padding: 5px;">
9     <a href="index.html">Nesli Tükenen Hayvanlar</a>
10    <a href="foto.html">Fotoğraflar</a> >
11    <a href="form.html">Form</a>
12  </div>
13  <div>
14    <h1> Nesli Tükenen Hayvanlar </h1>
15    
16    
17    
18    
19    
20    
21  </div>
22 </body>
23 </html>
24
```

```
2 <!DOCTYPE html>
3 <html>
4 <head>
5   <title> Futbol Sahası </title>
6 </head>
7 <body>
8   <script>
9     var kkenar, ukenar, alan, cevre;
10
11     kkenar=Number(prompt("Genişlik:"));
12     ukenar=Number(prompt("Yükseklik:"));
13
14     alan=kkenar*ukenar;
15     cevre=2*(kkenar+ukenar);
16
17     alert("Futbol sahasının alanı:"+alan+"metrekaredir, çevresi:"+cevre+"metredir.");
18
19     if(alan<1501)
20       alert("Futbol sahanız halı sahadır.");
21     alert("Futbol sahanız stadyumdur.");
22
23   </script>
24 </body>
25 </html>
26
```

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Soyad, Ad: Karaçam Duman, Nur Faden

Uyruk: Türk (T.C.)

Doğum Tarihi: 1 Ağustos 1991, Isparta

Medeni Durum: Evli

Telefon: +447387820702

email: nurfaden@gmail.com

EĞİTİM

Derece	Kurum	Mezuniyet Yılı
Lisans	Yıldız Teknik Üniversitesi	2014
Lise	Burdur Anadolu Lisesi	2009

İŞ DENEYİMİ

Yıl	Kurum	Görev
2017-2019	FMV Özel Ayazağa Işık Ortaokulu	Bilişim Tek. Öğrt.
2015-2017	Özel Arel Ortaokulu	Bilişim Tek. Öğrt.
2014-2015	Özel Ataşehir Amerikan Kültür Ortaokulu	Bilişim Tek. Öğrt.

YABANCI DİL

İngilizce (İyi Düzey), Almanca (Başlangıç Düzey)

SERTİFİKALAR

Temel HTML Kursu Başlangıç ve İleri Seviye, 2 saat, 2019

JavaScript, 11 saat, 2019

YAYINLAR

Bozkurtlar Peçe, S., Karaçam Duman, N. F., Yıldız, M. (2019). *Kodlama Korsanları Yolculuk Başlıyor*. İstanbul: Pusula Yayıncılık.