

T.C.
SAKARYA ÜNİVERSİTESİ
EĞİTİM BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR VE ÖĞRETİM TEKNOLOJİLERİ EĞİTİMİ ANABİLİM
DALI
BİLGİSAYAR VE ÖĞRETİM TEKNOLOJİLERİ EĞİTİMİ BİLİM DALI

BLOK TABANLI PROGRAMLAMA ARACININ 6. SINIF
ÖĞRENCİLERİNİN PROGRAMLAMA BAŞARISI,
ALGORİTMA GELİŞTİRME ÖZ-YETERLİKLERİ VE
GÜDÜLENMELERİNE ETKİSİ

YÜKSEK LİSANS TEZİ

FUNDA BAKIRCI

DANIŞMAN

DOÇ. DR. ÖZCAN ERKAN AKGÜN

HAZİRAN 2019

T.C.
SAKARYA ÜNİVERSİTESİ
EĞİTİM BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR VE ÖĞRETİM TEKNOLOJİLERİ EĞİTİMİ ANABİLİM
DALI
BİLGİSAYAR VE ÖĞRETİM TEKNOLOJİLERİ EĞİTİMİ BİLİM DALI

BLOK TABANLI PROGRAMLAMA ARACININ 6. SINIF
ÖĞRENCİLERİNİN PROGRAMLAMA BAŞARISI,
ALGORİTMA GELİŞTİRME ÖZ-YETERLİKLERİ VE
GÜDÜLENMELERİNE ETKİSİ

YÜKSEK LİSANS TEZİ

FUNDA BAKIRCI

DANIŞMAN

DOÇ. DR. ÖZCAN ERKAN AKGÜN

HAZİRAN 2019

Bildirim

Hazırladığım tezin tamamen kendi çalışmam olduğunu, akademik ve etik kuralları gözeterek çalıştığımı ve her alıntıya kaynak gösterdiğimi taahhüt ederim.



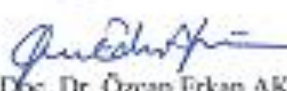
Funda BAKIRCI

JÜRİ ÜYELERİNİN İMZA SAYFASI

'Blok Tabanlı Programlama Aracının 6. Sınıf Öğrencilerinin Algoritma Geliştirmeye Yönelik Başarısı, Öz-Yeterlik Algısı ve Güdülenmelerine Etkisi' başlıklı bu yüksek lisans/doktora tezi, Bilgisayar ve Öğretim Teknolojileri Eğitimi Anabilim/bilim Dalında hazırlanmış ve jürimiz tarafından kabul edilmiştir.

Başkan Dr. Öğr. Üyesi  Ozlem Canan GÜNGÖREN

Öye Dr. Öğr. Üyesi  Zeliha DEMİR KAYMAK

Öye (Danışman) Doç. Dr.  Özcan Erkan AKGÜN

Yukarıdaki imzaların, adı geçen öğretim üyelerine ait olduğunu onaylanmıştır.

01.08.2019


Prof. Dr. Ömer Faruk TUTKUN

Enstitü Müdürü 

ÖN SÖZ

Lisans eğitimimde almaya başladığım programlama dersleri, öğrenmekte istekli olduğum ve başarıyla tamamladığım derslerin başında gelmiştir. Günümüzde, ülkelerin programlama eğitimine yönelik çalışmaları, programlama öğretiminin olası katkıları ve geliştirilen blok tabanlı, görsel programlama araçları öğrencilerin programlama ile daha erken yaşlarda tanışmalarını sağlamıştır. Öyle ki; öğrenciler küçük yaşlarda tasarladıkları oyunları ve uygulamalarını geliştirebilmektedirler. Ancak programlama kavramlarının soyut yapısı, programlama dillerinin karmaşıklığı gibi sorunlar, çocuklar için programlama öğretiminin önünde engel olarak görülmektedir. Bundan dolayı; programlama eğitiminin nasıl verilmesi gerektiği konusu bilim insanları tarafından tartışılmaktadır. Ülkemiz Vizyon 2023 çalışmalarında programlama eğitiminin önemini vurgulamakta ve gerçekleştirilmesi planlanan tasarım-beceri atölyelerinde programlama becerilerinin kazandırılması hedeflemektedir. Bu çalışmada, blok tabanlı Scratch programlama aracı ile verilen programlama öğretiminin öğrencilerin başarı, öz-yeterlik algısı ve güdülenmelerine etkisi araştırılmıştır.

Yüksek lisans eğitimim boyunca gerek aldığım derslerde, gerekse tez sürecinin her aşamasında sabır ve anlayışla benden yardımlarını esirgemeyen, bilgi ve deneyimlerini paylaşan danışmanım, saygıdeğer hocam Doç. Dr. Özcan Erkan AKGÜN' e sonsuz teşekkürlerimi sunarım.

Yüksek lisans ders sürecinde verdikleri eğitimleri ile bana çok şey katan Doç. Dr. Mehmet Barış HORZUM ve Doç. Dr. Mübin KIYICI hocalarıma, tez sürecinde deneyimlerini paylaşarak yardımlarını esirgemeyen değerli arkadaşlarım Dr. Ebru ALBAYRAK ÖZER'e ve Ayşe GÖNÜLTAŞ'a sonsuz teşekkürlerimi sunarım.

Son olarak bugünlere gelmemde en büyük rolü oynayan, aldığım her kararda, yaptığım herşeyde arkamda duran canım annem Hatice BAKIRCI ve canım babam Mehmet BAKIRCI'ya, kardeşlerim, Mustafa BAKIRCI ve Fatma DAĞDELEN'e sonsuz teşekkürlerimi sunarım.

ÖZET

BLOK TABANLI PROGRAMLAMA ARACININ 6. SINIF ÖĞRENCİLERİNİN PROGRAMLAMA BAŞARISI, ALGORİTMA GELİŞTİRME ÖZ-YETERLİKLERİ VE GÜDÜLENMELERİNE ETKİSİ

Bakırcı, Funda

Yüksek Lisans Tezi, Bilgisayar ve Öğretim Teknolojileri Eğitimi Anabilim Dalı
Bilgisayar ve Öğretim Teknolojileri Eğitimi Bilim Dalı

Danışman: Doç. Dr. Özcan Erkan Akgün

Haziran, 2019. ix+111 Sayfa.

Bu çalışmada, “Ortaokul Bilişim Teknolojileri ve Yazılım” dersinde programlama öğretiminde blok tabanlı Scratch programlama aracının kullanımının öğrencilerin algoritma geliştirmeye ilişkin öz-yeterlik algıları, derse yönelik güdülenmeleri ve programlama başarılarına etkisinin olup olmadığının tespit edilmesi amaçlanmıştır.

Çalışma iki aşamadan oluşmuştur. Birinci aşamada 2016-2017 yılında bilişim teknolojileri ve yazılım dersinin öğretim programında yer alan programlama konusunun kazanımlarına uygun olarak başarı testi geliştirilmiştir. Öğrencilerin öz-yeterlik algısı düzeyini belirlemek için “Algoritma geliştirmeye ilişkin öz yeterlik algısı ölçeği” geliştirilmiştir. Öğrencilerin derse ilişkin güdülenme düzeylerini belirlemek için “Güdülenme ve Öğrenme Stratejileri Ölçeği” kullanılmıştır.

Araştırmanın deneysel sürecinin yürütülmesine yönelik olarak ortaokul altıncı sınıf bilişim teknolojileri ve yazılım dersi “Algoritma ve Strateji Geliştirme” konusunda 4 haftalık öğretim programı hazırlanmıştır. Çalışmada kullanılacak ölçekler ve öğretim planı uzman görüşleri doğrultusunda hazırlanmış ve uygulanmıştır. Çalışmanın ikinci aşamasında ise hazırlanan öğretim planlarına uygun olarak 4 haftalık deneysel süreç gerçekleştirilmiştir.

Araştırmanın deseni, yarı deneysel desenlerden ön-test son-test deney – kontrol gruplu desendir. Deney grubu öğrencilerine birinci hafta Scratch programının arayüzü tanıtılmış, sonraki dört hafta süresince öğrenciler Scratch ortamında oyun geliştirerek programlama öğrenmişlerdir. Kontrol grubu öğrencileri ise dört hafta boyunca geleneksel anlatıma dayalı yöntemle akış şemaları oluşturarak programlama öğrenmişlerdir. Araştırmanın çalışma grubunu Sakarya ili merkez ilçedeki bir

ortaokulda 2016-2017 Eğitim-Öğretim döneminde 6. sınıfta eğitim gören, bir deney bir kontrol grubu olmak üzere toplam 64 öğrenci oluşturmaktadır. Verilere ulaşmak için öz-yeterlik algısı ölçeği, başarı testi ve güdülenme ölçeği kullanılmıştır. Uygulamalardan önce ve sonra öğrencilerden bu ölçekleri doldurmaları istenmiş ve başarı testi uygulanmıştır.

Araştırmada elde edilen veriler SPSS 25.0 ile analiz edilmiştir. Araştırma sonucunda elde edilen verilerin dağılımlarını belirlemek için normal dağılım analizlerinden Shapiro-Wilk analizi yapılarak dağılımın normal olmaması sonucu 2'li gruplar arasındaki farkın anlamlılığına bakarken Mann Whitney U testi, ön test-son test düzeylerine göre farklılıklar bakılırken Wilcoxon Sıralı İşaretler Sıralı İşaretler testi yapılmıştır.

Araştırmanın sonucunda; hem deney hem de kontrol grubunda öğrenen öğrencilerde başarı puanı, öz-yeterlik algısı düzeyi sonuçlarında istatistiksel olarak anlamlı bir artış gerçekleşmiştir. Bununla birlikte Scratch ortamında öğrenen öğrencilerin güdülenme düzeylerinde artış görülürken; geleneksel yöntemlerle öğrenen öğrencilerin güdülenme düzeylerinde azalma görülmektedir. Ancak başarı puanı, öz yeterlik algısı ve güdülenme puanları açısından deney ve kontrol grubu arasında anlamlı bir farklılık görülmemiştir. Bu bulgular doğrultusunda algoritmik düşünme becerilerinin geliştirilmesi ve algoritma öz-yeterlik algısının artırılması için derslerde programlama etkinliklerine daha çok yer verilmesi önerilmektedir.

Anahtar kelimeler: 21. Yüzyıl Becerileri, Programlama, Algoritma Geliştirme, Öz-yeterlik Algısı, Güdülenme, Blok Tabanlı Programlama Aracı, Scratch.

ABSTRACT

THE EFFECTS OF BLOCK BASED PROGRAMMING TOOL ON 6th YEAR STUDENTS' PROGRAMMING ACHIEVEMENT, ALGORITHM DEVELOPMENT SELF-COMPETENCIES AND MOTIVATION

Bakırcı, Funda

Post Graduate Thesis, Department of Computer Education and Instructional Technologies

Computer Education and Instructional Technologies

Supervisor: Associate Prof. Dr. Özcan Erkan AKGÜN

June, 2019. ix+111 Pages

In this study, it is aimed to determine whether the use of block-based Scratch programming tool in programming teaching in Information Technologies and Software course affects self-efficacy perceptions, course motivation, and programming achievement. The study consisted of two stages. In the first stage, an achievement test was developed in 2016-2017 in accordance with the intended learning outcomes of programming in the curriculum of Information Technologies and Software course. In order to determine the level of self-efficacy perception of the students, a Self-efficacy Perception Scale for Developing Algorithms was developed. Motivation and Learning Strategies Scale was used to determine the students' motivation level. In order to carry out the experimental process of the research, a 4-week teaching program on Algorithm and Strategy Development was prepared. The scales and instructional plan to be used in the study were prepared and applied in accordance with expert opinions. In the second stage of the study, a 4-week experimental process was conducted in accordance with the prepared teaching plans.

The research model is the quasi-experimental design model with pre-test post-test control groups. Scratch program was introduced to the experimental group students in the first week. The control group students learned to program for four weeks by creating flow charts using the traditional lecture method. The study group consists of a total of 64 students in a secondary school in the central district of Sakarya. Self-efficacy perception scale, achievement test, and motivation scale were used to reach

the data. Before and after the treatments, students were asked to complete these scales and an achievement test was applied.

The data were analyzed with SPSS 25.0. In order to determine the distribution of the data obtained from the study as a result of the normal distribution analysis Shapiro-Wilk analysis was done by the normal distribution of the difference between the groups as a result of the difference between the two groups while looking at the Mann Whitney U test, pre-test results were examined by looking at the Wilcoxon Signed Rank test.

As a result of the research, it was seen that there was a statistically increase in the achievement score and self-efficacy perception results of the students who were learning in both the experimental group and the control group. In addition, the motivation level of students learning in the Scratch programming environment has increased; the level of motivation of students learning traditional methods is reduced. However, no significant difference was found between the experimental and control groups in terms of achievement score, self-efficacy perception, and motivation scores. Based on these findings, it is suggested to improve the algorithmic ability skills and to improve the algorithm self-efficacy perception and to offer more space for programming activities in the courses.

Keywords: 21st Century Skills, Programming, Algorithm Development, Self-efficacy Perception, Motivation, Block Based Programming Tool, Scratch.

İÇİNDEKİLER

Bildirim.....	i
Jüri Üyelerinin İmza Sayfası	ii
Önsöz	iii
Türkçe Özet	iv
İngilizce Özet.....	vi
İçindekiler	viii
Tablolar Listesi	xi
Şekiller Listesi	xiii
BÖLÜM I: Giriş	1
1.1 Problem.....	6
1.2 Alt Problemler.....	6
1.3 Önem.....	7
1.4 Sınırlılıklar	9
1.5 Tanımlar.....	10
1.6 Kısaltmalar.....	11
BÖLÜM II: Araştırmanın Kuramsal Çerçevesi ve İlgili Araştırmalar	12
2.1 21. Yüzyıl Becerileri.....	12
2.2 Programlama.....	14
2.3 Programlama Öğretimi	17
2.4 Ülkemizde Programlama Eğitimi	21
2.5 Öz-yeterlik	23
2.5.1 Programlamaya İlişkin Öz-yeterlik.....	24

2.5.2 Algoritma Geliştirmeye İlişkin Öz-yeterlik.....	24
2.6 GÜdülenme.....	25
2.7 Scratch	26
2.8 İlgili Araştırmalar	31
2.8.1 Scratch'ın Programlama Başarısı ve GÜdülenmeye Etkisi İle İlgili Araştırmalar	31
2.8.2 Scratch'ın 21. Yüzyıl Becerilerine Etkisi İle İlgili Araştırmalar	34
2.8.3 Scratch'ın Öz-yeterlik Algısına Etkisi ile İlgili araştırmalar	36
2.9 Alanyazın Taramasının Sonucu	37
BÖLÜM III: Yöntem	39
3.1 Araştırmanın Modeli.....	39
3.2 Çalışma Grubu	40
3.3 Deneysel İşlemler	41
3.3.1 Kontrol Grubu Ders İçeriğinin Hazırlanması ve Uygulanması	41
3.3.2 Deney Grubu Ders İçeriğinin Hazırlanması ve Uygulanması	42
3.4 Veri Toplama Araçları	46
3.4.1 Kişisel Bilgi Formu.....	47
3.4.2 Öz-yeterlik Algısı Ölçeği.....	48
3.4.3 GÜdülenme ve Öğrenme Stratejileri Ölçeği	56
3.4.4 Başarı Testi	57
3.5 Verilerin Analizi	61
BÖLÜM IV: Bulgular	62
4.1 Araştırmada Kullanılan Ölçeklere İlişkin Betimsel Bulgular.....	62
4.2 Başarı Testi, Öz-yeterlik Algısı ve GÜdülenme Puanlarının Deney ve Kontrol Gruplarının Karşılaştırılmasına Yönelik Bulguları	63
4.4 Başarı Testi, Öz-yeterlik Algısı ve GÜdülenme Puanlarının Deney Grubuna Yönelik Bulguları	65

4.5. Başarı Testi, Öz-yeterlik Algısı ve Gdlenme Puanlarının Kontrol Grubuna Ynelik Bulguları	68
BLM V: Sonu, Tartıřma ve neriler.....	72
5.1 Sonu ve Tartıřma	72
5.1.1 Deney ve Kontrol Gruplarının Bařarı Puanı, z-yeterlik Algısı ve Gdlenme Dzeylerinin Karřılařtırılmasına Ynelik Ortaya ıkan Sonular.....	73
5.1.2 Scratch Programlama Aracı İle ğrenen ğrencilerin Bařarı Puanı, z-yeterlik Algısı ve Gdlenme Dzeylerine Gre Ortaya ıkan Sonular.....	74
5.1.3 Programlama Aracı Kullanılmayan Ortamda ğrenen ğrencilerin Bařarı Puanı, z-yeterlik Algısı ve Gdlenme Dzeylerine Gre Ortaya ıkan Sonular.....	75
5.2 neriler	78
5.2.1 Uygulayıcılara Ynelik neriler	78
5.2.2 Arařtırmacılara Ynelik Olarak neriler	79
Kaynaka	81
Ekler	94
zgemiř ve İletiřim Bilgisi.....	111

TABLolar LİSTESİ

Tablo 1. Arařtırmada Kullanılacak Ön Test- Son Test Kontrol Gruplu Yarı Deneysel Desen.....	40
Tablo 2 Deneysel Sürecin Haftalık Konu, Kazanım ve Etkinlikler Tablosu	45
Tablo 3. Öğrencilerin Demografik Özelliklerine Ait Dağılımları	47
Tablo 4. Öğrencilerin Deney ve Kontrol Grubuna Ait Dağılımları	48
Tablo 5. Öz-yeterlik Algısı Ölçeđi Maddelerine İlişkin İstatistikler	49
Tablo 6. Öz-yeterlik Algısı Ölçeđinin Madde ve Toplam Ölçek Korelasyonu Deđerleri.....	50
Tablo 7 : Öz yeterlik Ölçeđi KMO ve Barlett Testi Sonucu.....	51
Tablo 8. Öz yeterlik Ölçeđinin Öz Deđerleri ve Açıkladıkları Varyans Düzeyi	52
Tablo 9. Öz-yeterlik Algısı Ölçeđi Maddeleri Faktör Yük Deđerleri	53
Tablo 10. Algoritma Geliřtirmeye İlişkin Öz -yeterlik Algısı Ölçeđi Alt-Üst % 27'lik Gruplar İçin t-Testi Sonuçları	55
Tablo 11. Ölçeklere İlişkin Düzey Sınırları	56
Tablo 12. Başarı Testine İlişkin Madde Analizi Sonuçları	58
Tablo 13. Belirtke Tablosunda Bulunan Kazanımlar ve Soru Dağılımları	60
Tablo 14. Programlama Başarı Testinin Belirtke Tablosu.....	60
Tablo 15. Arařtırmada kullanılan Ölçeklere İlişkin Betimsel Bulgular.....	62
Tablo 16. Verilerin Dağılımına Ait Normallik Testi Sonuçları	63
Tablo 17. Öğrencilerin Başarı Testi Düzeyleri Deney-Kontrol Grubuna Göre Farklılıklarına Ait Mann Whitney U Testi Sonucu.....	64
Tablo 18. Öğrencilerin Öz-yeterlik Algısı Düzeylerinin Deney-Kontrol Grubuna Göre Farklılıklarına Ait Mann Whitney U Testi Sonucu.....	64
Tablo 19. Öğrencilerin Güdöleme Düzeyleri Deney-Kontrol Grubuna Göre Farklılıklarına Ait Mann Whitney U Testi Sonucu.....	65
Tablo 20. Deney Grubu Öğrencilerin Öntest-Sontest Başarı Testi Puanlarının Farklılıklarına Ait Wilcoxon Sıralı İşaretler Testi Sonucu	66

Tablo 21. Deney Grubu Öğrencilerin Öntest-Sontest Öz-yeterlik Algısı Puanlarının Farklılıklarına Ait Wilcoxon Sıralı İşaretler Testi Sonucu	67
Tablo 22. Deney Grubu Öğrencilerinin Öntest-Sontest Güdüleme Puanlarının Farklılıklarına Ait Wilcoxon Sıralı İşaretler Testi Sonucu	68
Tablo 23. Kontrol Grubu Öğrencilerinin Öntest-Sontest Ders Başarısı Puanlarının Farklılıklarına Ait Wilcoxon Sıralı İşaretler Testi Sonucu	69
Tablo 24. Kontrol Grubu Öğrencilerin Öntest-Sontest Öz yeterlik Algısı Puanlarının Farklılıklarına Ait Wilcoxon Sıralı İşaretler Testi Sonucu	70
Tablo 25. Kontrol Grubu Öğrencilerinin Öntest-Sontest Güdüleme Puanlarının Farklılıklarına Ait Wilcoxon Sıralı İşaretler Testi Sonucu	71

ŞEKİLLER LİSTESİ

Şekil 1. Programlama Dili Öğretim Materyallerinin Sınıflandırılması (Gültekin, 2006).	19
Şekil 3. Scratch'in Kod Panelindeki Algılama Kodlarının Görünümü.....	29
Şekil 4. Kontrol Grubu Öğrenci Görseli	42
Şekil 5. Deney Grubu Öğrenci Görseli	43
Şekil 6. Deney Grubunda Hazırlanan Proje Örneği	44
Şekil 7. Deney Grubunda Hazırlanmış Bir Projeye İlişkin Akran Dönütleri.....	44
Şekil 8. Öz-yeterlik Ölçeğine Ait Scree Plot Grafiği	53

BÖLÜM I

GİRİŞ

21. yüzyılın ilk dönemini yaşadığımız günümüzde bilim ve teknolojideki gelişmeler hayatın neredeyse her alanını etkilediği gibi eğitim dünyasında da değişikliklere sebep olmuştur. Eğitim öğretim faaliyetlerinde kullanılan araç gereçler teknolojik özellik bakımından büyük bir değişime uğramıştır. Öyle ki; geleneksel beyaz tahtaların yanı sıra etkileşimli tahtalar, tablet bilgisayarlar ve çeşitli eğitim yazılımları öğretim ortamlarında kullanılmaya başlanmıştır.

Günümüzde, eğitim kurumlarında çocukların bilişim teknolojilerini bilinçli bir şekilde kullanmaları, 21. yüzyıl becerilerinin kazandırılmasında ve gelecekte karşılaşılabilecekleri gerçek hayat problemlerini çözebilecek donanımla yetiştirilmesinde büyük öneme sahip olduğu görülmektedir. Ancak çocukların bu teknolojileri nasıl ve ne amaçla kullanmaları gerektiğinin bilincinde olmamalarından dolayı üretim amacıyla kullanmak yerine çoğunlukla eğlence odaklı kullandıkları söylenebilir (Coşar, 2013).

Günümüz koşullarının gerektirdiği niteliklere sahip bir öğrencide olması gereken becerileri Milli Eğitim Bakanlığı “MEB 21. Yüzyıl Öğrenci Profili” çalışması doğrultusunda açıklamıştır. Özetle; 21. yüzyılda yetişen öğrenci, etkili iletişim kurabilen, işbirliği yapabilen, problem çözme, analitik ve eleştirel düşünme, yaratıcı düşünme becerilerine sahip, araştıran ve sorgulayan, öğrenmeye açık, bilgiye ulaşabilen ve bilgiyi yönetebilen, teknolojiyi etkili ve verimli kullanabilen, teknoloji kullanımında seçici olan, kendini yenileyebilme, kendini yeniden oluşturabilme becerisine sahip, nasıl öğreneceğini bilen, kendine güvenen birey olmalıdır. Fark oluşturabilmeli ve bilimsel okur-yazar olabilmelidir (EARGED, 2011).

Ülkemizin vizyon 2023 çalışmalarında öğrencilerin bilişim teknolojilerini çevrim içi ve çevrimdışı ortamlarda üretim, problemlere çözüm geliştirme ve hayallerini hayata geçirme aracı olarak kullanmaları amaçlanmaktadır. Okullarda kurulması planlanan “Tasarım-Beceri Atölyeleri”nde bilmekten çok tasarlamanın, yapmanın, üretmenin ön plana çıkarılmasını amaçlamaktadır. 2019-2020 yılında gerçekleştirilmesi planlanan bu atölyeler yeni çağın gerektirdiği problem çözme, eleştirel düşünme, üretkenlik, takım çalışması ve çoklu okuryazarlık becerilerinin kazandırılması için somut mekânlar olarak düzenlenecektir. Bu amaçlar doğrultusunda, teknoloji kullanımının temel bilgisayar kullanımından daha öteye taşınarak öğrencilerin var olan ürünleri geliştirebilmeleri ve yeni bilgi ve teknolojiler üretebilmeleri vurgulanmaktadır. Bu durum bilişim teknolojileri eğitiminde programlama eğitiminin önemini ortaya çıkarmaktadır (MEB, 2018).

Programlama, bir problemin çözümünün programlama dili kullanılarak oluşturulan kod satırlarına verilen isimdir (Arabacıoğlu, Bülbül ve Filiz 2007). Sayın ve Seferoğlu (2016), günümüzde, programlamanın bir ihtiyaçtan çok zorunluluk olmaya başladığını ifade etmişlerdir. Yapılan araştırmalar da 21. yy becerilerini kazandırmanın yollarından birinin programlama eğitimi olduğunu göstermektedir (Akpınar ve Altun, 2014; Çakıroğlu, Sarı ve Akkan, 2011). Programlama eğitiminin soyut düşünme becerilerine (Erdoğan, 2005), eleştirel düşünme becerilerine (Coşar, 2013), öğrencilerin problem çözme ve analitik düşünme becerilerine (Akpınar ve Altun, 2014; Çakıroğlu ve diğerleri, 2011) ayrıca erken yaşlarda başlayan programlama etkinliklerinin çocukların bilişsel gelişimi üzerine (Akpınar ve Altun, 2014; Atmatzidou ve Demetriadis, 2016; Lewis, 2010), farklı düşünme, yaratıcılık yetenekleri ile üst biliş becerilerine (Clements ve Gullo, 1984), kompütasyonel düşünme (bilişimsel düşünme) becerilerine (Kalelioğlu ve Gülbahar, 2014; Kasalak, 2017; ScratchEd Team, 2011) olumlu etkisi olduğu görülmektedir.

Bu becerilere ek olarak; programlama sürecinde öğrencilerin ürüne dönük büyük projeler yapma, yaparak öğrenme ve bilgisayara öğreterek öğrenme alışkanlıkları gelişmektedir (Akpınar ve Altun, 2014; Çakıroğlu ve diğerleri, 2011). Programlama eğitimi öğrencilerin bilgisayar tutumuna, akademik başarılarına (Coşar, 2013) ve programlamaya yönelik algılarına (Lewis, 2010; Kalelioğlu ve Gülbahar, 2014) olumlu katkı sağlamaktadır.

Yapılan çalışmalardan görülmüştür ki; programlama eğitimi, 21. yüzyıl becerisi olarak sayılan eleştirel düşünme, problem çözme, analitik düşünme, algoritmik düşünme becerilerinin gelişimine olumlu katkı sağlamaktadır. Bu bağlamda programlama öğretiminin etkili ve verimli bir şekilde gerçekleştirilmesi ve hedeflerine ulaşması öğrencilere birçok alanda yarar sağlayabilir.

Kert ve Uğraş (2009) programlamanın bireysel gelişime olan olumlu etkilerinden dolayı programlama eğitimine küçük yaşlardan itibaren önem verilmesi gerektiğini vurgulamaktadır. Programlama ile öğrenciler kendi geliştirdikleri programlarıyla belki de ileride geleceğe yön veren projelere imza atacaktırlar (Karabak ve Güneş, 2013).

BTE Derneği (2012) hazırladığı “Bilişim Teknolojileri Eğitimi Durum Raporu” ‘nda, Bilişim Teknolojileri dersinin sanılanın aksine sanal ortamlardaki tehlikelerden sağlığa; kaliteli bilgiye ulaşmaktan, programcılığa; web tasarımcılığında iletişime, güvenlikten bilgi okuryazarlığına çok çeşitli içerikleri barındıran öğretim programı yenilikleri gerektirdiği vurgulanmıştır. Programlama eğitimin önemini artması ile birlikte ülkemizde Bilişim Teknolojileri Dersi Öğretim Programı’na programlama konularını eklemenin zorunluluğu ortaya çıkmıştır. 2012 yılında Bilişim Teknolojileri dersinin adı değişerek “Bilişim Teknolojileri ve Yazılım” dersi olmuş ve dersin içeriğine yazılım ve programlama eğitimi konuları eklenmiş ve 5. sınıftan itibaren öğrencilere temel programlama eğitimi zorunlu olarak verilmeye başlanmıştır (BTE Derneği, 2013).

Fakat programlama öğretiminin önünde bir takım zorluklar bulunmaktadır. Başer’ e (2013) göre öğrenciler programlamayı öğrenmesi zor bir iş olarak görmektedir. Özmen ve Altun (2014), yaptıkları çalışmada bu zorlukların; öğrencilerin başta pratik eksikliği ve algoritma oluşturamama, bunlara ilaveten hata ayıklama, programlama bilgisi eksikliği ve programlama becerileri eksikliğinden dolayı kaynaklandığını ifade etmişlerdir. Programlama deneyimi yüksek olan öğrencilerin programlama başarılarının ve programlamaya ilişkin öz yeterlik algılarının yüksek olduğu görülmüştür (Özmen ve Altun, 2014). Hawi’nin (2010) yaptığı çalışmada, programlama öğretiminde, öğrencilerden başarısızlıklarının sebebini açıklamalarını istediğinde, öğrenciler öğrenme stratejilerinden, öğretme yöntemlerinden ve konunun zor olmasından kaynaklandığını ifade etmişlerdir. Programlama eğitiminin önündeki

bu engellerden dolayı öğrenciler programlamaya yönelik olumsuz tutum geliştirmektedirler (Korkmaz, 2016).

Programlama dilinin karmaşık olması öğrenmeyi zorlaştırabilir. Bu derste algoritma mantığı kazandırıldıktan sonra seçilen programlama dilinde deneyim kazandırılmalıdır (Ersoy, Madran ve Gülbahar, 2011). Özoran, Çağıltay ve Topallı'ya (2012) göre de programlama öğrenimindeki bu karışıklık öğrencilerin programlamayı öğrenmelerine engel oluşturmaktadır. Programlama eğitiminin başarısı göz önüne alındığında programlama eğitimine başlandığında öğrencilere programlama dilinden önce programlama mantığının kazandırılması büyük önem arz etmektedir. (Gökoğlu ve Yüksel, 2016; Ersoy ve diğerleri, 2011; Futschek, 2006.). Çünkü programlama dillerinin komut yapıları birbirinden farklılık gösterse de programlama mantığı tüm programlama dillerinde benzerdir. (Futschek, 2006). Arabacıoğlu ve diğerlerine (2007) göre algoritmalar programlama eğitiminin temeli olarak görülmektedir.

Yukarıda belirtilen araştırma sonuçları ve ortaya çıkan bazı durumlara göre, programlama eğitimi ile 21. yüzyıl becerileri kazandırılmakta veya geliştirilebilmektedir. Ancak programlama dillerinin görsel ve mantıksal karmaşıklığı, programlama öğrenmeye engel oluşturabilmektedir. Hedeflenen kazanımlara ulaşılabilmesi için programlama eğitiminin ilk aşamasında algoritma mantığının kazandırılmasının büyük öneme sahip olduğu savunulmaktadır.

Programlama eğitiminde kullanılmak üzere alternatif olarak çeşitli kullanıcı profiline uygun olarak programlama ortamları oluşturulmuştur. Çocuklara ve yeni başlayanlara daha basit bir şekilde programlama algoritması öğretmek amacıyla onların seviyesine hitap eden çeşitli blok tabanlı programlama ortamları eğlenceli, öğretici ve görselliği ön planda olarak geliştirilmiştir. Bu araçlardan bazıları çevrimiçi olarak programın kendi web sitesinden kullanılmakta (code.org vb.), bazıları da bilgisayara kurulduktan sonra çevrimdışı kullanılabilir (scratch, alice vb.).

Öğretimde kullanılacak eğitim yazılımının seçimi oldukça önemli ve özen gösterilmesi gereken bir konudur. Yazılımın hangi amaca yönelik kullanılacağı belirlenmesi ve öğrenci özelliklerinin iyi bir şekilde tanımlanması gerekmektedir. Doğru materyalin tercih edilmesi; öğretimin hedeflerine ulaştırabilmesi, öğrencilerin

kendi hızına göre ilerleyebilmesine katkı sağlayabilmesi, öğrencilerin derse yönelik ilgi ve motivasyonlarını artırabilmesi açısından önemli görülmektedir (Özmen ve Varol, 2011).

Programlama eğitiminde oyun, hikâye tasarlama etkinlikleri öğrencilerin programlama mantığını kavramalarını kolaylaştırmaktadır (Cooper ve diğerleri, 2003; Moreno, 2012; akt. Erol, 2015). Alice ve Scratch gibi programlar oyun ve hikâye tasarlama olanakları sağlamasından dolayı algoritma geliştirme sürecini daha kolay ve anlaşılır hale getirmekte ayrıca problem çözme sürecini daha ilgi çekici hale getirerek öğrencilerin somut ve özgün bir ürün oluşturmaya olanak tanımaktadır (Utting, Cooper, Kölling, Maloney ve Resnick, 2010).

Programlama eğitimine öğrenilmesi zor olan programlama dillerinin öğretimi ile başlanıldığında yapılan etkinlikler çoğunlukla öğrencilerin ilgisini çekmez. Ve sonuçta toplumun sadece küçük bir kesimi için programlama uygun bir etkinlik olmaya devam eder. Ancak programlama öğrenmeye blok tabanlı görselleştirici bir programlama aracı ile başlandığında yapılan etkinliklerin herkes için için daha kolay ve anlaşılır olması mümkündür. Blok tabanlı Scratch programlama aracı, arayüzünde hazır bulunan kod bloklarının sürüklenip bırak özelliği ve kodların görsel bir sahnede canlandırılması ile programlamayı kolaylaştırmaktadır. Çünkü yapılandırmacı yaklaşıma dayanarak geliştirilen Scratch; basit yapısı, oyun ve hikâye gibi tasarım yapabilme olanağı sağlaması ve programlama yapılarına uygunluğu ile programlama öğretiminde kullanılabilir (Resnick ve diğerleri, 2009; Resnick, 2012).

Bu özelliklerinden dolayı bu çalışmada Scratch aracı tercih edilmiştir. Bilişim teknolojileri ve yazılım dersinde programlama öğretiminde yardımcı araç olarak Scratch kullanımının öğrencilerin ders başarıları ve derse yönelik güdülenme düzeyleri üzerindeki etkisi araştırılacaktır.

Ancak programlama öğretiminin hedeflerine ulaşmasında başarıyı etkilediği düşünülen bilişsel becerilerin ve güdülenmenin yanı sıra başka duyuşsal faktörler de göz önüne alınmalıdır. Bu faktörlerden biri de bireyin bir görevi yerine getirmede kendine yönelik olan öz-yeterlik algısıdır.

Bireyin yeterliklerine ilişkin inançları, bireysel hedeflerini, bu hedeflerin koşullarını ve düzeylerini belirleme sürecini etkileyecektir. Öz-yeterlik inancı daha yüksek olan

öğrenciler daha yüksek hedef belirleyecekler, hedeflere ulaşmak için güdülenecek ve daha çok başarı ortaya koyacaklardır (Akgün, 2008).

Öğretmenler, öğrencilerin akademik başarısının artmasında öz yeterlik algısının önemini farkında olup; dersin plan ve programını hazırlarken, öğrenme etkinliklerini düzenlerken öğrencilerin söz konusu olan konuya ilişkin öz yeterlik düzeylerini göz önüne almalıdırlar (Ekici, 2006).

Öz-yeterlik algısı ile ilgili alanyazında aynı anlama gelen farklı kullanımlar göze çarpmaktadır: "Öz-yeterlik inancı" (Akgün, 2008; Akkoyunlu, Orhan ve Umay, 2003;), "algılanan öz-yeterlik" (Senemoğlu, 2012), "öz-yeterlik algısı" (Aşkar ve Umay, 2001; Seferoğlu ve Akbıyık, 2005), "öz-yeterlik yargısı" (Celep, 2000), "öz-yeterlik duygusu" (Önen ve Öztuna, 2005). Bu çalışmada ise "öz-yeterlik algısı" ifadesi tercih edilmiştir.

Benzer olarak; alanyazın incelendiğinde programlama kavramının yerine "kodlama" (Kasalak, 2017; Sayın ve Seferoğlu, 2016) ifadesinin kullanıldığı görülmektedir. Bu çalışmada "programlama" ifadesi tercih edilmiştir

1.1 PROBLEM

Bilişim Teknolojileri ve Yazılım dersinde "Algoritma ve Strateji Geliştirme" konusunda blok tabanlı Scratch programlama aracının kullanılması, bu aracın kullanılmadığı ortama göre programlama başarısını, algoritma geliştirmeye ilişkin öz-yeterlik algısını ve güdülenmeyi nasıl etkilemektedir?

1.2 ALT PROBLEMLER

Çalışmada şu sorulara yanıt aranmıştır;

1. Programlama öğretiminde, blok tabanlı Scratch programlama aracının kullanıldığı ortamdaki deney grubu öğrencileri ile programlama aracının kullanılmadığı ortamdaki kontrol grubu öğrencilerin;
 - a. Ders başarısı puanları arasında anlamlı bir fark var mıdır?

- b. Öz-yeterlik algısı puanları arasında anlamlı bir fark var mıdır?
 - c. Derse ilişkin güdülenme puanları arasında anlamlı bir fark var mıdır?
 2. Blok tabanlı Scratch programlama aracının kullanıldığı öğretim ortamındaki deney grubu öğrencilerinin öntest- sontest;
 - a. Ders başarısı puanları arasında anlamlı bir fark var mıdır?
 - b. Öz-yeterlik algısı puanları arasında anlamlı bir fark var mıdır?
 - c. Derse ilişkin güdülenme puanları arasında anlamlı bir fark var mıdır?
 3. Programlama aracının kullanılmadığı geleneksel öğretim ortamındaki kontrol grubu öğrencilerinin öntest-sontest;
 - a. Ders başarısı puanları arasında anlamlı bir fark var mıdır?
 - b. Öz-yeterlik algısı puanları arasında anlamlı bir fark var mıdır?
 - c. Derse ilişkin güdülenme puanları arasında anlamlı bir fark var mıdır?

1.3 ÖNEM

Hayatımızın önemli bir parçası olan eğitim alanı bireyi ilköğretimden üniversiteye kadar olan süreçte akademik bilgi ve deneyimlerin yanı sıra sosyal ve toplumsal birçok alanda geliştirmektedir. Öğrencilerin Matematik eğitiminde problem çözme becerileri, Türkçe eğitiminde hitabet becerileri, Müzik eğitiminde işitsel ve ritm tutma gibi becerileri gelişirken, Bilişim Teknolojileri eğitiminde temel bilgisayar kullanımının yanı sıra dijital vatandaşlık, programlama gibi becerileri de gelişmektedir.

Günümüzde ezberleyen, sorgulamayan, eleştirmeyen bireyler yerine 21. yüzyıl becerileri ile donatılmış üretken ve yaratıcı bir nesil yetiştirmek hedeflenmektedir. Programlama eğitiminin bireysel gelişime olan etkileri düşünüldüğünde erken yaştan itibaren programlama eğitimi verilmesi gerektiği bilim insanları tarafından sıklıkla dile getirilmektedir. Fakat programlama eğitiminin soyut işlem becerileri

gerektirmesi ve geleneksel programlama dillerinin yapılarındaki karmaşıklık özellikle küçük yaştaki öğrenciler için zor ve sıkıcı bir süreç olarak görülebilmektedir. Programlama eğitiminin hedeflenen kazanımlara ulaşmasında alternatif olarak çocukların seviyesine hitap eden eğlenceli ve görselliği ön plana alan programlama platformları oluşturulmuştur: Scratch, Alice, code.org vb. Bu ortamlarda sürükle bırak yöntemi ile kod bloklarının yapboz gibi kenetlenmesiyle algoritma daha basit bir şekilde öğretilmektedir. Bu ortamlarda algoritma yapısı kavratılırken hata ayıklama, işbirlikli çalışma gibi beceriler de geliştirilebilir.

Bu becerilerin yanı sıra duyuşsal becerileri kazandırmanın da önemli olduğu araştırmalarla ortaya çıkarılmıştır. Tutum, inanç gibi duyuşsal faktörlerin arasında önemli bir içsel faktör de öz yeterlik algısıdır. Çünkü öğrencilerin belirlenen kazanımlara ulaşabilmesi ve kendisine yüklenen sorumlulukları yerine getirebilmesi için kendisine olan inancının önemli olduğu vurgulanmaktadır.

Scratch programında programlama dillerinin karmaşıklığının bulunmaması, algoritma ve döngü, koşul gibi temel programlama yapılarının somutlaştırılması, kod bloklarının basit kullanımı bu ortamı eğlendirici ve öğretici hale getirebilmektedir. Programlama eğitiminde bu ortamın kullanılması öğretimin daha etkin olmasına ve kendi oyununu üretebilen öğrencilerin programlamaya yönelik öz yeterlik algılarının artmasına katkı sağlayabilir.

Scratch programı ülkemizde K12 düzeyinde ve üniversitede programlamaya giriş derslerinde yaygın olarak kullanılmaktadır. Bu programın öğrenciler üzerinde problem çözme, motivasyon, öz yeterlik algısı artışı gibi etkileri araştırılmaktadır. Ancak çalışmaların yeterli sayıda olmadığı görülmektedir (Genç ve Karakuş, 2011; Kalelioğlu ve Gülbahar, 2014; Yükseltürk ve Altıok, 2015).

21. yüzyıl becerileri ile donatılmış, teknolojiyi bilinçli olarak kullanabilmesinin yanı sıra ona hâkim olan üretken ve yaratıcı bir nesil yetiştirebilmek için bireylere erken yaştan itibaren programlama eğitimi verilmesinin önemi vurgulanmaktadır (Saygıner ve Tüzün, 2017). Bundan dolayı programlama eğitiminin verimliliğini artırmaya yönelik yapılan çalışmalar önem kazanmaktadır.

Araştırmada Scratch programlama aracının programlama başarısına ve başarıda önemli sayılan duyuşsal özelliklerden öz yeterlik algısı ve güdülenme düzeyine etkisi incelenecektir. Bu bağlamda ortaokul 6. sınıf düzeyine yönelik programlama temel

kavramlarını öğrenme ve algoritma geliřtirmeye iliřkin öz yeterlik algısı ölçeęi geliřtirilmiřtir. Çalıřma, bu özellięinden dolayı alanyazına katkı saęlayacaktır.

Bu arařtırma;

- Ortaokul düzeyine yönelik programlama temel kavramlarını öğrenme ve algoritma geliřtirmeye iliřkin öz yeterlik algısı ölçeęinin geliřtirilmesi aęısından *özgün*;
- Ortaokul öğretim programlarında yer alan biliřim teknolojileri ve yazılım dersi içerięine katkı saęlaması aęısından *iřlevsel*;
- K12 düzeyindeki müfredatlarda bulunan programlama konusu kapsamında öğrencilere programlamayı sevdirmek amacıyla yenilikçi programlama platformlarının kullanımını teřvik etme olasılıęının olması aęısından *güncel ve faydalı*;
- Scratch programlama aracının öğrenci güdülenmesine, öz-yeterlik algısına ve ders başarısına etkisi konusunda alanyazına katkı saęlayacak olması aęısından *gerekli* görölmektedir.

1.4 SINIRLILIKLAR

Bu arařtırma;

1. Ölçeklerin geliřtirilme ařamasında uzman görüşüne bařvurulan BÖTE bölümünde öğretim üyesi olan 5 uzman ile
2. Ölçeklerin pilot uygulamasında ve güvenirlik çalıřmasında ulařılan Sakarya ili, Adapazarı ilçesindeki bir ortaokulda öğrenim gören 171 öğrenci ile
3. Deneysel çalıřmalarda ulařılan Sakarya ile Adapazarı ilçesinde pilot uygulama ve geçerlik güvenirlik çalıřmalarında yer almamıř bir ortaokulda öğrenim gören 32 deney grubu öğrencisi ve 32 kontrol grubu olmak üzere toplam 64 öğrenci ile
4. 2012 yılında yayımlanan Biliřim Teknolojileri ve Yazılım dersi öğretim programı ile
5. Programlama öğretimi için ayrılan 4 haftalık süre ile

6. Çalışmanın deneysel sürecinin gerçekleştirilmesine yönelik geliştirilen 4 haftalık öğretim planı ile sınırlıdır.

1.5 TANIMLAR

Programlama: Bir problemin çözümüne ilişkin adımlar ile çözüm sürecinin bir programlama dili ile bilgisayar ortamına aktarım işi (Eryılmaz, 2003).

Programlama başarısı: Herhangi bir programlama dilini kullanarak verilen bir programlama görevini başarı ile yerine getirme, programı anlama, program hatalarını ayıklayabilme, yeniden bir program yazabilme, programın algoritma şemasını oluşturabilme becerisi (Altun ve Mazman, 2012).

Algoritma: İyi tanımlanmış kuralların ve işlemlerin adım adım uygulanmasıyla bir sorunun giderilmesi veya sonuca en hızlı biçimde ulaşılması işlemi (Özkan, 2003).

Programlama dili: Elektronik cihazları bir görevi yerine getirebilmeleri için hazırlanmış özel kelime ve sembollerden oluşan komutlar bütünü (Ersoy ve diğerleri, 2011).

Akış şeması: Algoritmaların içerdiği işlemlerin geometrik şekiller ile ifade edilmesi (Eker, 2011).

Öz-yeterlik algısı: Bireyin bir konuda performansını gösterebilmesi için kendi kapasitesine olan inancı (Bandura, 1977).

Güdülenme: Bir hedefe yönelik olarak davranışı harekete geçiren, sürdüren ve yönlendiren güç (Dilekmen ve Ada, 2005).

Blok tabanlı programlama aracı: Programlama dillerindeki komutlar yerine günlük konuşma diline uygun olarak geliştirilmiş hazır kod bloklarını kullanan programlama araçlardır.

Scratch: Algoritma ve programlama öğretiminde kullanılan, animasyon ve oyun gibi tasarıma olanak tanıyan grafik tabanlı bir program (Maloney, Resnick, Rusk, Silverman ve Eastmond, 2010).

1.6 KISALTMALAR

ISTE	: International Society for Technology in Education (Uluslararası Eğitimde Teknoloji Topluluğu)
TTKB	: Talim Terbiye Kurulu Başkanlığı
MEB	: Millî Eğitim Bakanlığı
BTE	: Bilişim Teknolojileri Eğitimcileri Derneği
EARGED	: Milli Eğitim Bakanlığı Eğitim Araştırma Dairesi Başkanlığı
K12	: İlkokul 1. sınıftan 12. sınıfa kadar olan eğitim dönemi
MEB	: Milli Eğitim Bakanlığı
%	:Yüzde
ss	:Standart sapma
\bar{X}	:Aritmetik Ortalama
P	: Manidarlık düzeyi
r	: Korelasyon
sd	:Serbestlik derecesi
f	: Frekans
n	: Birey sayısı

BÖLÜM II

ARAŞTIRMANIN KURAMSAL ÇERÇEVESİ VE İLGİLİ ARAŞTIRMALAR

Bu bölümde 21. yüzyıl becerileri, programlama kavramları, programlama öğretimi ile kazandırılan 21. yüzyıl becerileri, programlama öğretimi yöntemleri, ülkemizde programlama öğretiminin nasıl verildiği, güdülenme kavramı, öz yeterlik kavramı, Scratch programlama aracının özellikleri, Scratch aracının kullanımına yönelik yapılan araştırmalar ile ilgili bilgiler yer almaktadır.

2.1 21. YÜZYIL BECERİLERİ

Günümüzde, eğitim dünyasında “21. yüzyıl becerisi” ifadesi sıklıkla kullanılmaya başlanmıştır. Öğrencilerini çağın gerektirdiği şekilde geliştirmeyi hedefleyen öğretmenler 21. yüzyılda yetişen öğrencilerde hangi özelliklerin bulunması gerektiğini bilmeleri, eğitim ortamlarını ve etkinliklerini düzenlemelerine yardımcı olabilir.

Bu bağlamda, araştırmacılar söz konusu becerileri farklı başlıklar altında sınıflandırmışlardır. EARGED (2011), Milli Eğitim Bakanlığı 2011 yılında “21. yüzyıl öğrenci profili ” başlığı altında hazırladığı raporla 21. Yüzyılda yetişen öğrenci profilini açıklamıştır Etkili iletişim kurabilme, takım halinde çalışma (işbirliği), problem çözme, analitik düşünme, yaratıcı düşünme becerilerine sahip, bilimsel ve akılcı, araştırma ve sorgulama becerileri olan, öğrenmeye açık, nasıl öğreneceğini ve verimli ders çalışma tekniklerini bilen, bilgiye ulaşabilen ve bilgiyi yönetebilen, bilimsel okuryazar, zamanı etkili yönetebilen, teknolojiyi etkili ve verimli kullanabilen, teknolojiyi hangi amaçla kullanacağını bilen, kendini yenileyebilen, en az bir sanat dalını benimsemiş, Türkçe’yi iyi kullanabilmenin yanı

sıra yabancı dil bilen, kütüphaneyi etkin bir şekilde kullanabilen, kendini yeniden oluşturabilen, vizyonlu, demokratik tutuma sahip, sınıf ve okul yönetimi ile ilgili kararlara katılabilen, insanlığın ortak değerlerine değer veren, insani ve ahlaki değerlere sahip, yaşam boyu öğrenme becerisine sahip, hoşgörülü, kendisiyle barışık, yurttaşlık bilincine sahip, özgüveni yüksek, öz disiplinli, her türlü fanatizmden uzak kalabilen, dünya sorunlarına duyarlı, dünya barışına katkı sağlayabilen arkadaş seçiminde seçisi bir birey olarak tanımlanabilir (EARGED, 2011).

Uluslararası Eğitimde Teknoloji Topluluğu (ISTE) tarafından 2016 yılında hazırlanan raporda ise, günümüzün dijitalleşen dünyasında öğrencilerin iyi düzeyde gelişmelerinin ve çağa ayak uydurabilmelerinin sağlanması amacıyla öğrencilere kazandırılması gerekli beceri ve yeterlikleri 7 başlık halinde listelenmiştir (ISTE, 2016): Yetkin öğrenen, dijital vatandaş, bilgiyi düzenleyen, yaratıcı tasarımcı, kompütasyonel düşünen, yaratıcı iletişimci, global işbirlikçi.

ISTE'nin (2011) hazırladığı raporda kompütasyonel düşünmenin tanımlanması aşağıdaki yapılmıştır.

- 1. Problemleri bilgisayar veya başka bir araç kullanarak çözmeyi sağlayacak şekilde formüle etme*
- 2. Veriyi mantıksal olarak organize etme ve analiz etme*
- 3. Veriyi model veya benzetimler gibi soyutlaştırmalar yolu ile gösterme*
- 4. Algoritmik düşünme yolu ile otomatik cevaplar üretme*
- 5. Olası çözümleri, olası aşamaları ve kaynakları en etkili ve verimli şekilde birleştirerek tanımlama, analiz etme ve uygulama*
- 6. Bu problem çözme sürecini diğer farklı problem durumlarına transfer etme ve genelleme.*

Kompütasyonel düşünmenin tanımlamasında da yer alan ve programlama aşamalarından biri olan “algoritmik düşünme,” 21. yüzyıl becerisi sayılmaktadır. Günüş, Odabaşı ve Kuzu (2013) ise yaptıkları çalışmada öğretmen adaylarından twitter aracı üzerinden 21. yüzyıl öğrenci özelliklerini tanımlamalarını istemişlerdir. Çalışma sonucunda öğretmen adaylarının tanımlamaları doğrultusunda 21. yüzyıl öğrenci özellikleri; kişisel beceriler, araştırma ve bilgi edinme becerileri, yaratıcılık, yenilik ve kariyer becerileri, teknoloji becerileri olarak 4 ana başlık olarak sınıflandırmışlardır. Teknoloji becerileri olarak teknolojiyi etkin, sağlıklı, verimli

kullanabilen, güncellenen teknolojileri kullanabilen ve üretebilen bireyler olarak tanımlanmıştır.

Programlama, her öğrencinin kazanması gereken bir 21. yüzyıl becerisi olduğu ve erken yaştan itibaren öğrencilere kazandırılması gerektiği düşüncesi ile müfredatlara dâhil edilmiştir. Öğretmenler programlama öğretiminin öğrencileri birçok alanda geliştirebileceğini ifade etmektedirler. Programlama öğretiminde problemlere çözüm üretmek, sonuçlarını görmek, hataları düzeltmek, bir işi tamamlamak, başarılı olunca sevinmek, daha iyisini hedeflemek ve emek vermek gibi daha birçok kazanımı mevcuttur (Yecan, Özçınar ve Tanyeri, 2017).

Programlama yaratıcılığı geliştirir, işbirliği yapmayı, fiziksel ve coğrafi engelleri aşarak takım halinde çalışmayı ve evrensel bir dilde iletişim kurmayı öğretir. Programlama, günümüzün okuryazarlığıdır ve problem çözme, işbirliği yapabilme ve analitik düşünme gibi 21. yüzyıl becerilerini uygulamaya yardımcı olur (European Commission, 2018)

“21. Yüzyıl Çocuklarında Kodlama Dili” başlıklı araştırmaya göre ABD ve Avrupa’da çocuklara programlamayı öğretmek, popülerliği artan bir konu haline gelmektedir. Bilişim teknolojileri alanında çalışmayan kişilerin dahi programlama öğrenmeleri gerektiği söylenmektedir. Apple, Oracle gibi yazılım üreticileri yüksek maaşla çalışan kişilerin genelde programlama alanında çalıştıklarını ve programlamanın her çocuğun öğrenmesi gereken ikinci bir dil olduğunu vurgulamaktadırlar. Ancak burada çocuklardan beklenen karmaşık programlama dilleri ile program yazmanın aksine programlama mantığının kavranmasıdır. Programlama için gerekli üç temel beceri ise, mantıksal düşünme, problem çözme ve yaratıcılıktır (URL1, 2017)

2.2 PROGRAMLAMA

Programlama; insan yaşamındaki bir problemin analizi ve tasarımı yapıldıktan sonra ilgili problemin çözümüne yönelik işlem adımlarının oluşturulmasının ardından program geliştiricilerin programlama dilleri komutlarını kullanarak yapmak istediği işleri bilgisayara aktarması işidir (Eryılmaz, 2003). Programlama, bir problemin çözmek için programlama dili kullanılarak oluşturulan kod satırlarına verilen isimdir

(Arabacıođlu ve diđerleri, 2007). Aslında programlama gerçek yařamdaki olayların bilgisayarın anlayabileceđi modellendirilmesidir. Program geliřtiriciler genellikle soyut yapıda olan programlamayı somutlařtırarak yani gerçek hayata yaklařtırarak öđrencilerin anlamasına yardımcı olurlar. Program geliřtirirken öncelikle problemi analiz etmek, söz konusu olan problemin çözümlüne gerçek hayatta çözümler üretmek ve sonrasında sadece araç olarak düşünülmesi gereken komutları kullanarak programlama yapısını oluřturmak gerekmektedir (Çölkesen, 2002; akt. Demir, 2015). Bilim insanları tarafından belirtilen tanımlara bakıldığında programlama, gerçek yařamdaki bir probleme iliřkin üretilen çözümler yolunun programlama dili kodlarının dizilimi ile bilgisayar programları geliřtirme süreci olarak ifade edilebilir.

Algoritma; bir problemin çözümlenmesi ya da sonuca en hızlı biçimde ulařılması için için dođru yaklařımlar sergileyerek iyi tanımlanmış kuralların adım adım uygulanması işlemidir (İmal ve Eser, 2009; İnce, řenyüzlü ve Uđur, 2007). Yani algoritma, bir problemin çözümlünün adım adım olarak tarif edilmesidir diyebiliriz. Matematik ve bilgisayar bilimleri dışında günlük hayatta da karřılařtıđımız problemlere iliřkin algoritma oluřturabiliriz.

Günlük hayattan bir problem örneđi olarak; lamba yanmıyor. Sorunu çözmek için algoritma geliřtirmeliyiz.

Birinci adım: Lambanın Fiři Takılı mı? Deđilse, fiři tak. Sorun çözümlüřtür.

Takılıysa ikinci adıma geç

İkinci adım: Ampul patlak mı? Cevap hayır ise yeni lamba al, Evet ise ampulü deđiřtir.

Problem çözüldü.

Günlük konuşma dilinde oluřturulan algoritma adımlarının bilgisayarın anlayacađı dilde bilgisayara aktarılması gerekir. Programlama dili, programcının yapmak istediđi şeyleri yani bir algoritmayı bilgisayara anlatmasıdır (Karabak ve Güneř, 2013).

Akış diyagramları, algoritma adımlarının sembolleřtirilmiş halidir. Özellikle programlamaya yeni bařlayanlar açısından problemi anlayıp algoritma oluřturma sürecinde programlamaya iliřkin kavramları kullanmak oldukça zordur. Programlama eđitiminde bu soyut ve zor sürecin ařılması için programlama problemlerinin çözümlünde genellikle akış diyagramları kullanılmaktadır (Eker,

2011). Program ise belirli görevlerin yapılması için bilgisayar komutları kullanılarak yazılıp bilgisayar ortamında çalıştırılan yapıdır (Coşar, 2013).

Kesici ve Kocabaş 'a (2007) göre, bir bilgisayar programının hazırlanması için beş temel aşama açıklamaları ile aşağıda verilmiştir.

Problemin Tanımı: Problemin çok iyi anlaşılması ve net olarak tanımlanması gerekir.

Çözüm Yolunun Belirlenmesi (Algoritmanın Hazırlanması): Problem tanımlandıktan sonra problemin çözümü için algoritma geliştirilir. Yani yapılacak işlemler adım adım maddelendirilir. Algoritma oluşturulduktan sonra maddeler programın genel yapısı şekil ve sembollerle akış şeması üzerinde gösterilir. Çözüm yolu daha önce kullanılmış veya yeni bir yol olabilir.

Programın Kodlanması: Seçilen programlama dilinin komutları ile program yazılır.

Programın Yorumlanması ve Derlenmesi: Yazılan program bir derleyicide makine diline çevrilerek çalıştırılır. Derlenen program kullanılmaya hazırlanmış demektir.

Programdaki Hataların Belirlenmesi ve Giderilmesi: Bu aşamada, geliştirilmiş programın çalışma durumundaki yazım (kodlama) ve mantık hatalarının belirlenir ve giderilir. Daha sonra program tekrar denir.

Feddon ve Charness (1999) ise programlamanın içerdiği alt işlemleri beş başlıkta tanımlamıştır:

Yazılım Tasarımı: Problemin gerekliliklerine karar verme, probleme en uygun çözümü üretme ve mantıksal bir temsil oluşturmaktır.

Anlama: Programın ne yaptığını anlamaktır.

Düzenleme: Tasvir edilen yapıyı program kodlarına çevirmektir.

Hata Ayıklama: Programın yanlış çalışmasının ya da çalışmamasının sebeplerini bulmaktır.

Modifikasyon: Geliştirilen programda değişiklikler yapmaktır.

2.3 PROGRAMLAMA ÖĞRETİMİ

21. yüzyıl becerilerini geliştirmede önemli sayılan programlama öğretimin daha verimli olabilmesi için hangi ortamda ve nasıl gerçekleştirileceği alan yazında tartışılan bir konu olmuştur.

Bayman ve Mayer' e (1988) göre programlama sürecinde öğrenenlerin kazanması gereken üç tür programlama bilgisi vardır. Bunlar kavramsal bilgi, yazımsal bilgi ve problem çözme bilgisidir.

- **Yazımsal (Syntactic) Bilgi:** *Belirli bir programlama dilinin yapılan kurallarının kullanım bilgisi*
- **Kavramsal (Conceptual) Bilgi:** *Programlama kavramlarını ve mantığını anlama bilgisi*
- **Problem Çözme – Stratejik (Strategic) Bilgi:** *Yazımsal ve kavramsal bilgiyi kullanarak problem çözme bilgisi*

Günümüze kadar 500'e yakın programlama dili geliştirilmiştir. Programlama dillerin tamamını öğrenmek zor görünmektedir. Tüm bu dillerin kendine has söz dizimleri ve yazım kuralları olsa da programlama mantığı tümünde benzerdir. Bu nedenle bir programcının programlama dilinden önce programlama mantığını öğrenmesi gerekmektedir (Erol, 2015)

Programlamaya yeni başlamış olan öğrenciler hem öğrendikleri programlama diline ait komutları ve kuralları öğrenirken aynı zamanda programlama mantığını da öğrenmek zorunda kalmaktadırlar. Öğrenciler söz konusu programlama diline ait bütün komutları ve kuralları bilseler bile programlama mantığını kavrayamamakta ve yeni karşılaştıkları farklı bir programlama problemini çözmekte sorun yaşamaktadırlar (Cooper ve diğerleri, 2003; Garner ve diğerleri, 2005; Kinnunen ve Malmi, 2008; akt. Erol, 2015).

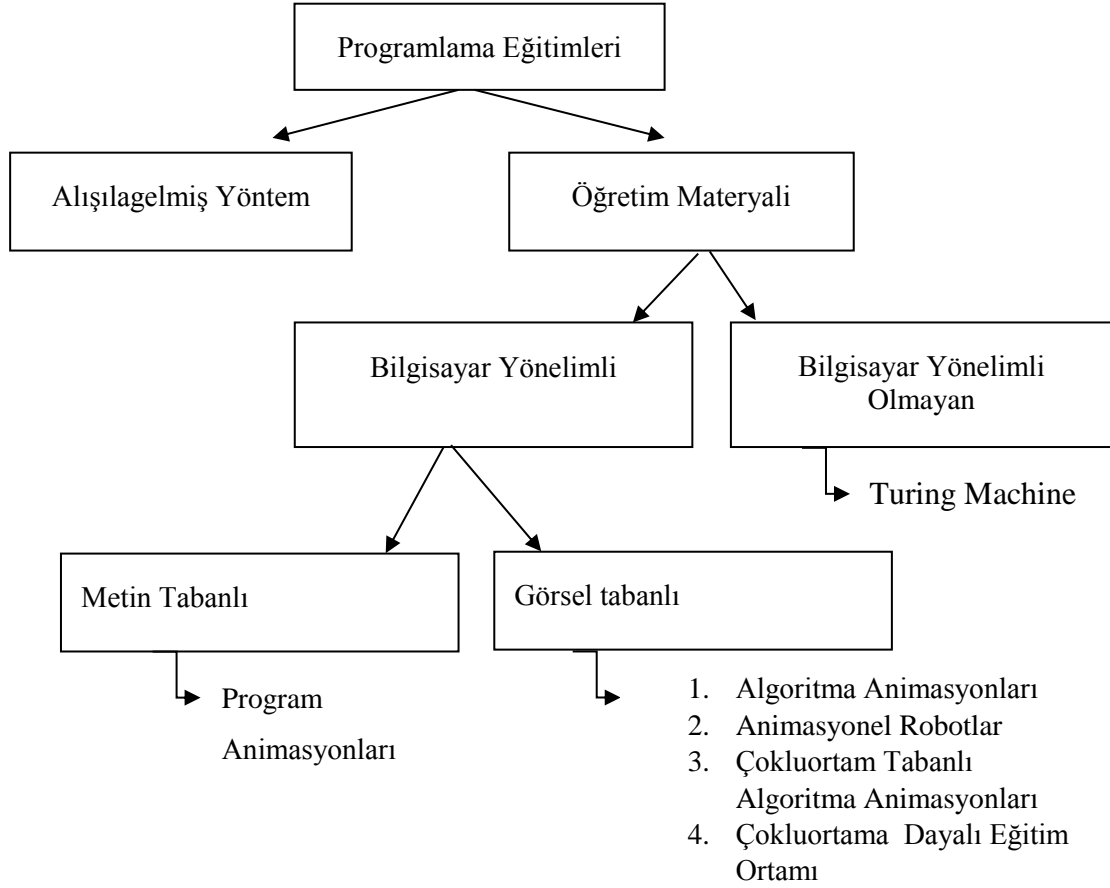
Örnek olarak Basic programlama dilinde “If..end” koşullu ifadesinin yazımı veya komutların arasına koşul yazılması bilgisi programı yazmak için gereken yazımsal bir bilgidir ancak bir probleme çözüm üretmek ya da tasarım yapmak için yeterli değildir. Bundan dolayı programlamaya yeni başlayanlar ve özellikle çocuklar için doğrudan bir programlama dilinin öğretilmesi yerine programlama mantığının ve problem çözme sürecinin kazandırılması daha uygun bir olabilir. Bir programlama

diline geçilmeden önce temel programlama yapıları ve programlama mantığı çerçevesinde öncelikle günlük hayattan algoritma problemleri kullanılabilir.

Temel programlama kavramları değişkenler, karar verme ve kontrol işlemleri, döngüler ve diziler olarak adlandırılabilir. (Eryılmaz, 2003). Programlama dilleri birçok kavramı içerisinde barındıran bir öğrenme alanıdır. Fakat informal programlama eğitimine sahip çoğu kişi “koşul”, “değişken” gibi programlama dili yapılarını kavramadan programlama içerisinde bu yapıları kullanmaktadır. Formal bir eğitim için ön ve öncelikli öğrenmeler temel kavramlar ile başlamalıdır (Gültekin, 2006).

Geçmişte programlama öğretimi mevcut sınıf ortamında sunum yöntemi gibi alışlagelmiş yöntemler kullanılarak yapılmaya çalışılmıştır. Alışlagelmiş öğretim yöntemlerinin ön plana çıkan olumsuzlukları; öğretenden öğrenene doğru bilgi akışı olması ve derslerin anlatım yoluyla iletilmesi, öğrenende hem bir motivasyon düşüklüğü hem de tam bir “anamlı öğrenme” gerçekleştirilememesi olarak belirtilebilir. Bu olumsuzluklardan ve programlama öğretiminin karmaşık yapısından dolayı programlama öğretimi için çeşitli öğretim materyalleri geliştirilmiştir (Gültekin, 2006).

Şekil 1’de programlama öğretiminde kullanılan programlama dili öğretim materyallerinin sınıflandırılması görülmektedir.



Şekil 1. Programlama Dili Öğretim Materyallerinin Sınıflandırılması (Gültekin, 2006).

Programların veya algoritmaların çalıştırıldığında dinamik görsel sunumlarını öğrenciye sunmak amacıyla çoklu ortama dayalı birçok araç geliştirilmiştir. Çoklu ortam araçlarından bazılarının üç boyutlu etkilerle bazılarının ses eklentileriyle algoritmaları görselleştirerek ve somutlaştırarak öğrencide bir anlam oluşturması amaçlanır (Gültekin, 2006).

Algoritma öğretiminin soyut bir süreç olması ileride öğrencilerin ilgisini çekememektedir. Bu nedenle algoritma öğretimini şekil ve sembollerle görselleştirerek daha somut ve anlaşılır hale getirmek için eğitimciler Raptor, Flint, FCPro gibi akış diyagramı oluşturma araçları da kullanmaktadırlar (Baldwin ve Kuljis, 2000; Cooper, Powers, McNally, Goldman, Proulx, Carlisle, 2006; Cooper ve diğerleri, 2003; Maloney, Resnick, Rusk, Silverman ve Eastmond, 2010; akt. Erol, 2015).

Akış diyagramı araçlarına ek olarak programlama öğreniminin zor ve sıkıcı sürecini kolay ve eğlenceli hale getirerek programlama mantığının anlaşılmasını kolaylaştırmak için çeşitli organizasyonlar düzenlenmekte ve programlama araçları geliştirilmektedir. Programlama öğrenmek isteyen herkesin bu imkâna sahip olabilmesi için Code Academy, Code Club, Khan Academy, Coder Dojo ve Code.org gibi organizasyonların yanı sıra Scratch, Microsoft Small Basic, Alice, MIT App Inventor gibi programlama araçları da kullanılmaktadır (Demirer ve Sak, 2016).

Yapılan araştırmalara göre programlama öğretiminde önce konuların genel hatlarıyla verilip daha sonra uygulama etkinliklerinin yapılması yerine teori ve pratik etkinliklerin iç içe uygulanması ile daha başarılı olmaktadır (Crews ve Murphy, 2004, Ziegler ve Crews, 1999; akt. Hu, 2004)

Korhonen (2003) programlama öğretiminde öğrenme ortamının nasıl hazırlanması gerektiği ile ilgili yaptığı çalışmasında, öğrencilerin algoritmayı görselleştirme araçları kullanarak uygulama süreçlerinde aktif katılım sağlamalarının, kendilerine verilen problem örneklerini düzenleyebilmelerinin ve tasarım yapabilmelerinin önemli olduğunu vurgulamaktadır.

Korhonen (2003) programlama öğretimi özetle beş farklı yöntemle yapılabilmektedir:

1. *Öğretici materyaller kullanmak*
2. *Görselleştirme aracı kullanmak*
3. *Sınıf ortamında öğretmen rehberliğinde uygulamalar yapmak*
4. *Öğrencilerin internet ortamında kendi hızlarına göre ilerleyebilmelerine olanak tanımak*
5. *Yukarıdaki maddelerin hep birlikte kullanımı. Yani 1. yöntemden 5. yönteme kadar tüm yöntemler sırası ile kullanılabilir.*

Coşar (2013) programlamada üst düzey becerilerin kazandırılması için öncelikle temel programlama kavramlarının kazandırılması ardından problem çözme stratejileri sunulması gerektiğini vurgulamaktadır.

2.4 ÜLKEMİZDE PROGRAMLAMA EĞİTİMİ

Programlamanın olası katkıları önceki bölümde açıklanmıştır. Bilim insanları yaptıkları çalışmalar sonucunda programlama eğitimi alan öğrencilerin problem çözme, yaratıcılık gibi önemli becerileri geliştirdiğini tespit etmişlerdir. Programlamanın öneminin artmasıyla birlikte ülkeler K12 müfredatlarına programlamayı eklemeye başlamışlardır. Bazı ülkelerde programla eğitimi bağımsız bir ders olarak bazı ülkelerde bilgisayar bilimi dersine entegre edilerek verilebilmektedir.

Ülkemizde programlama eğitimi K12 düzeyinde devlet okulları ve özel okullar tarafından verilmektedir. Bunun yanında özel kuruluşlar insanların ihtiyaçlarına göre çeşitli programlama, robotik kodlama gibi kursları açmakta ve öğrencileri ilgi duydukları konularda geliştirmektedirler.

Günümüzde çağın ihtiyaçları doğrultusunda öğrencilerden beklenen beceriler değiştikçe bilişim teknolojileri dersinin müfredatı, okutulduğu sınıflar ve ders saati de değişime uğramaktadır.

Ülkemizde programlama becerileri eğitimi ilköğretim düzeyinde ilk olarak Milli Eğitim Bakanlığı Talim Terbiye Kurulu Başkanlığı tarafında 28.08.2006 tarihindeki karar ile 2007-2008 eğitim öğretim yılından itibaren İlköğretim Seçmeli Bilgisayar dersi kapsamında verilmeye başlanmıştır. Öğretim programında 8 basamak bulunmaktadır. Programlama eğitimi ilk olarak 6. basamakta “işlemlerin ve problemlerin çözümünü yaparken algoritmanın ve programlamanın genel kavramlarını açıklar”, 7. Basamakta “programlama dillerinin çeşitlerini ve özelliklerini açıklar”, 8. Basamakta “ nesne tabanlı programlama dillerinin sağladığı kolaylıkları açıklar” olmak üzere 3 kazanım içermektedir (İnce ve diğerleri, 2007)

İlköğretimde Seçmeli Bilgisayar dersinin 2012 yılında geliştirilen yeni öğretim programı ile Bilişim Teknolojileri ve Yazılım Dersi olarak adı değiştirilmiştir. Programlama kazanımları ise “Problem Analiz ve Çözme Yaklaşımları”, “Algoritma ve Strateji Geliştirme (algoritma oluşturma mantığı, sözde kod, akış şemaları vb.)”, “Programlama”, “Yazılım Projesi Geliştirme, Uygulama ve Yaygınlaştırma” konu başlıklarında yer almaktadır. Bilişim Teknolojileri ve Yazılım dersi 5 ve 6. Sınıflarda

haftada 2 saat zorunlu, 7 ve 8. Sınıflarda haftada 2 saat seçmeli olarak verilmektedir (TTKB, 2012).

2017 yılında Bilişim Teknolojileri ve Yazılım dersi öğretim programı geliştirilip yeniden yayınlanmıştır. Yenilenen öğretim programının “Problem Çözme ve Programlama” ünitesi altında “problem çözme kavramları ve yaklaşımları” ve “programlama” konu başlıklarını içermektedir. Bilişim teknolojileri ve yazılım dersinin, 5. ve 6. sınıflarda haftada 2 saat zorunlu, 7. ve 8. sınıflarda ise haftada 2 saat seçmeli olarak yürütülmesi planlanmaktadır. Hem 5. Sınıflarda hem de 6. Sınıfta toplam 72 saat süren bilişim teknolojileri ve yazılım dersinin 36 saati, yani dersin toplam süresinin yarısı problem çözme ve programlama ünitesine ayrılmıştır. Öğretim programının kazanım sayısı olarak ise 6. Sınıflarda toplam 77 kazanımın 25’i, 5 sınıflarda ise toplam 75 kazanımın 27 ‘si “problem çözme ve programlama” ünitesinde bulunmaktadır. (TTKB, 2017).

Ortaöğretim düzeyinde 2016 yılında yayınlanan öğretim programında ise Bilgisayar Bilimi Dersi kapsamında programlama becerilerinin kazanımları “Problem Çözme ve Algoritmalar”, “Programlama”, “Web Tabanlı Programlama”, “Mobil Programlama” konu başlıklarında bulunan öğretim programı geliştirilmiştir (TTKB, 2016). Mesleki ve teknik liselerde ise bilişim teknolojileri dersinin yanı sıra bilişim bölümlerinde programlama dersi verilmektedir (Ersoy ve diğerleri, 2011).

Yükseköğretimde ise meslek yüksekokulları bünyesinde olan bilgisayar programcılığı bölümlerinde programlama eğitimi verilmektedir. Öğrenciler bu bölümde yazılım kurulumu ve yönetimi, internet programcılığı, nesne tabanlı programlama, görsel programlama, gibi dersler almaktadırlar. (YÖK, 2010) . Bu bölümün yanı sıra bilgisayar bilimleri üzerine kurulan bölümlerde de (ör. Bilgisayar Mühendisliği, Bilgisayar ve Öğretim Teknolojileri Öğretmenliği vb.) bilgisayar programlama becerileri öğretilmektedir.

Görülmektedir ki ülkemizde programlama eğitimi ortaokul ve lise düzeyinde bağımsız bir ders olarak değil Bilişim Teknolojileri dersinde bir ünite olarak müfredata dâhil edilmiştir. Derste kullanılacak öğretim materyali ders öğretmenin tercihine bırakılmıştır. Mesleki ve teknik liselerde ise programlama dillerinin öğretildiği, bilgisayar biliminden farklı bir ders olarak verilmektedir.

Programlama eğitiminin ortaokul ve ortaöğretim müfredatına dâhil edilmesi önemli bir mihenk taşı olmuştur. Programlama öğrenimini yetişkinler dahi zor bir süreç olarak ifade ederken küçük yaşlardaki öğrenciler tarafından anlaşılabilmesi ve hedefine ulaşabilmesi için eğitsel planlar yapılması gerekir (Mıhçı, Taçgın ve Arslan, 2017).

Ülkemizde programlama becerilerini geliştirmek, insanları programlamayla tanıştırmak amacıyla bilişim teknolojileri ve yazılım dersinin yanı sıra okul, ilçe yada il genelinde, farklı kurumlar tarafından projeler de yapılmaktadır: KodlaManisa, KodlaRize vb. Ülkemizde Bilişim Garaj Akademisi tarafından geliştirilen K12 düzeyine yönelik programlama, mobil programlama web tasarımı, 3D tasarım ve robot, akıllı cihaz üretimi konularında eğitimler ve projeler bulunmaktadır. Bu eğitimler sayesinde öğrenciler programlama ile problem çözme süreci öğrenmektedirler (Bilişim Garaj akademisi, 2019). Erken yaşlardan itibaren çocukların programlama becerilerini desteklemek, birlikte çalışmayı teşvik etmek ve programlama kültürünün yaygınlaştırılması amacıyla için Avrupa çapında yapılan 7-22 Ekim tarihleri arası Kod Haftası (Codeweek) etkinlikleri, 2018 yılından beri Milli Eğitim Bakanlığımızın üyesi olduğu Avrupa Okul Ağı (European Schoolnet) kuruluşu tarafından Avrupa Komisyonu adına yürütülmektedir (EBA, 2019).

2.5 ÖZ YETERLİK

Öz yeterlik, bireyin bir konuda performansını gösterebilmesi için kendi kapasitesine olan inancını ifade eder (Bandura, 1977). Öz yeterlik, insanların karşılaştıkları sorunlar ile mücadele edebilmek için yapılması gerekenleri ne kadar iyi yapabildiklerine dair inançları olarak ifade edilir. Öz yeterlik inancı insanların bir sorun karşısında sorunu çözebilmek için ne kadar ısrarcı olduğunun belirtisidir (Yıldırım Doğru, 2012).

Öz yeterlik, insanların karşılaştıkları olaylara bakış açılarını, düşünce biçimlerini ve duygusal tepkilerini etkiler. Öz yeterlik algısı yüksek olan insanlar zorluklarla karşılaştıklarında kendilerinden emin bir şekilde sabırla mücadele ederler. Öz yeterliği düşük olan insanların ise karşılaştıkları sorunlarda kaygıları yükselir ve bu durumda problemleri çözülmesi daha zor bir duruma getirirler. Sonuç olarak öz

yeterlik algısı ile başarı düzeyi arasında sıkı bir ilişki olduğu görülmektedir (Pajares, 2002). Öz yeterlik inancı güçlü olan bireyler zor bir görevle karşı karşılaştıklarında bu durumdan kaçmak yerine bu görevi üstesinden gelinmesi gereken bir durum olarak görmektedirler (Hazır Bıkmaz, 2004).

2.5.1 Programlamaya İlişkin Öz yeterlik

Öz yeterlik ile ilgili yukarıdaki tanımlardan yola çıkarak, programlamaya yönelik öz yeterlik algısı, kişinin programlama performansı gösterirken programlama görevlerini yerine getirip getiremeyeceği hakkında kendi kapasitesine ilişkin algısı şeklinde ifade edilebilir.

Programlama görevlerinin zor ve karışık görülmesi programlama öğrenmeye engel olabilmektedir (Başer, 2013; Özoran ve diğerleri, 2012). Öğrencilerin genellikle öz yeterlik algılarının düşük olması yani programlamayı baştan zor kabul etmeleri nedeniyle, programlama dersinde başarısız olmaları olasıdır (Altun ve Mazman, 2012). Bandura'ya (1977) göre insanların öz yeterlik inancı; etkinliklerin seçimini, zorluklarla karşılaştığında sabrını, gayretlerinin düzeyini ve performansını etkiler.

Kişinin programlamaya ilişkin öz yeterlik algısının; programlama başarısını ve programlama sürecinde karşılaştığı zorluklarla mücadele edebilme yani gayret düzeyini etkileyeceği söylenebilir. Öyle ise; kişinin programlamaya ilişkin öz yeterlik algısı yüksek olduğunda programlama kavramlarını ve mantığını öğrenirken programlamanın faydaları olarak belirtilen problem çözme, eleştirel düşünme gibi becerilerinin de geliştirebileceği söylenebilir (Akçay, 2015).

2.5.2 Algoritma Geliştirmeye İlişkin Öz yeterlik

Daha önceki bölümlerde algoritma kavramı açıklanmıştır. Algoritmaya kısaca; bir problemin çözümünün adım adım ifade edilmesi diyebiliriz.

Bir problem tanımlandıktan sonra bu problemin çözüm yolu algoritma oluşturularak ifade edilir. Bilim insanları algoritmayı; programlamanın temeli ve mantığı olarak görmektedirler (Arabacıoğlu ve diğerleri, 2007; Gökoğlu ve Yüksel, 2016; Ersoy ve diğerleri, 2011; Futschek, 2006,).

Öyle ise; programlama başarısının algoritmik düşünme ve algoritma geliştirme başarısına paralel olduğu söylenebilir. Bu bağlamda; programlamanın temeli ve mantığı olarak görülen, programlama temel kavramlarını öğrenme ve algoritma geliştirmeye ilişkin öz yeterlik algısı önemlidir. Şöyle ki; bir programlama probleminin nasıl çözülebileceğini ifade edebilme ve çözümde kullanılması gereken döngü, koşul gibi programlama yapılarını doğru olarak kullanabilmede kişinin kendine olan inancı programlama başarısına etki edebilir.

2.6 GÜDÜLENME

İnsan davranışlarını etkileyen çeşitli etkenlerden söz edilebilir. Ancak davranışın yönünü, şiddetini ve kararlılığını belirleyen en önemli güç kaynağı güdülenmedir (Fidan, 2012).

Güdülenme, bir hedefe yönelik olarak kişiyi harekete geçiren, hedefe ulaşmak için yapılan davranışı devam ettiren ve yönlendiren güçtür (Dilts, 1998; Lumsden, 1994; akt. Dilekmen ve Ada, 2005).

Güdülenme soyut bir kavram olup açıklaması oldukça güçtür. Ama insanları belirli amaca yönelik davranışlarını gözlemlediğimizde o kişilerin güdülenmiş olup olmadığı hakkında bir hükmederiz. Güdünün kaynağı çeşitli fizyolojik, psikolojik ve ya toplumsal ihtiyaçlardır (Balaban Salı, 2004; akt. Üçgül, 2013).

Öğrenme ile ilgili bir kavram olan güdülenme (motivasyon), kişinin bir eyleme başlayıp, o eylemi yapmaya devam ederek sonunda başarılı ve verimli olmasının en önemli belirleyicisidir. Genellikle başarısız çocuklarda güdülenme eksikliğinden bahsedilebilir. Çünkü güdülenmiş öğrenci davranışları ile güdülenmemiş öğrenci davranışları arasında farklar vardır. Öğrenmek için her öğrencinin öğrenme-öğretme süreçlerine istekli katılmak gibi bir zorunluluğu vardır. Sınıfta yeteri kadar Güdülenmemiş öğrenciler derslere düzenli devam etmez, ilgileri ve dikkatleri dağınık olup konu üzerine odaklanamaz. Güçlüklerle karşılaştığında mücadele etmek yerine geri çekilme davranışı gösterirler. Yeterince güdülenmiş öğrencilerde ise derse ilgi duyma, derse hazırlıklı gelme, sürekli soru sorma, tartışmalara katılma, davranış için çaba göstermeye ve gerekli zamanı ayırmaya isteklilik, konu üzerinde

odaklaşma, güçlüklerden yılmama, vazgeçmeme, ısrarlı ve kararlı olma gibi davranışlar gözlenir (Dilekmen ve Ada, 2005).

Yapılan tanımlardan güdülenme faktörünün başarı üzerinde olumlu etkisi olduğu görülmektedir. Bu bağlamda, güdülenme, programlama eğitiminde öğrencilerin taşınması gereken önemli özelliklerden biri olduğu görülmektedir ve bu konuda araştırmalar yapılmaktadır. Erol ve Kurt (2017) programlama eğitiminde kod blokları programlama ortamlarının kullanımının öğrenci güdülenmesini artırdığını belirtmişlerdir.

2.7 SCRATCH

Scratch, sözlükte "çizik" ve "tırmalamak" anlamlarına gelmektedir. Ancak Scratch kelimesi, Disk Jokeylerin (DJ) plaklar ile yaptıkları ses efektlerine de denilmektedir. Scratch programını oluşturan geliştiriciler DJ'lerin yaptıkları ses efektlerini birleştirerek yeni şarkılar yaratmasından yaratıcılıktan esinlenerek programlama ortamına "SCRATCH" adını vermişlerdir. Scratch, 8-16 yaşındaki çocuklar için tasarlanmıştır; ScratchJr ise 5-7 yaşındaki çocuklar için daha basitleştirerek tasarlanmıştır. 2 Ocak 2019 yılında Scratch'ın Scratch 2.0 'a göre özellik olarak geliştirilmiş olan yeni sürümü Scratch 3.0 yayınlanmıştır (Scratch, 2019a). Ayrıca sözlükte "kedi tırnağı" anlamına da gelmesinden dolayı programın logosu ve program sahnesinde yer alan karakter bir kedir. Bu çalışmada, deneysel süreçte Scratch 1.4 programı kullanılmıştır.

Scratch, Massachusetts Institute of Technology Üniversitesi'nde Amerikan Ulusal Bilim Kurumu tarafından desteklenen bir proje olarak geliştirilmiştir. Okul dışındaki teknoloji merkezlerinde vakit geçiren gençlerin teknolojiyi kullanmada daha maharetli olmalarını yani teknolojiyi nasıl kullanacağını bilerek ve anlamlı ürünler ortaya çıkarabilmesini amaçlamaktadır. MIT Medya Laboratuvarı'ndan Profesör Mitchel Resnick, UCLA Üniversitesi'nden Doçent Dr. Yasmin Kafai ve yine MIT Medya Laboratuvarı'ndan John Maeda'nın öncülüğünde araştırma grubu oluşturulmuştur. Proje 2003 yılında başlamış 2007 yılında bitirilmiştir (Resnick ve diğerleri, 2003)

Tüm dünyada 40'tan fazla dil seçeneği bulunan görsel bir programlama ortamıdır. Programının internet sitesindeki resmi verilere göre 2019 yılı Nisan ayında, kayıtlı toplam 39.497.483 üye ve 40.755.622 proje paylaşılmıştır. Scratch 'in her yaş grubundan kullanıcısı mevcuttur. Ancak yoğun olarak aktivite gerçekleştiren yaş grubu 10-18 yaş aralığıdır Scratch programının en genel amacı programlamaya yeni başlayanlara programlamayı tanıtmak ve sevdirmektir. Kullanıcılar Scratch' i kendi web sitesinden ücretsiz olarak indirebilmekte, bu ortam ile oyun, animasyon ya da dijital hikâyeler tasarlayabilmekte, ayrıca geliştirdikleri bu projeleri Scratch'in kendi web sitesinde diğer kullanıcılar ile paylaşabilmektedirler (Scratch, 2019b).

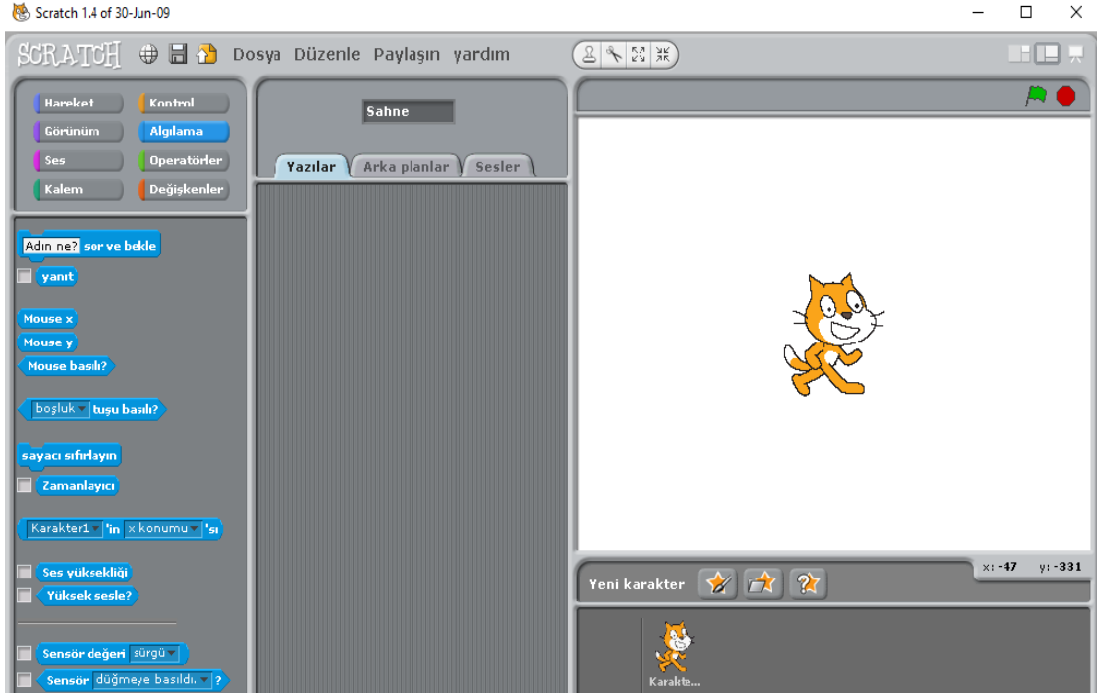
Scratch programlama ortamı Amerika Birleşik Devletleri'nde bulunan okul dışı teknoloji merkezlerindeki öğrencilerin teknolojiyi daha maharetli kullanmalarını amaçlayan bir proje olarak 2003 yılında ortaya çıkmıştır. Proje kapsamında 2005-2006 yılında yapılan araştırmada söz konusu olan merkezde Scratch kullanan öğrencilerin oranının yükseldiği görülmüştür (Kafai, Peppler ve Chin, 2007).

Scratch programlama ortamının genel eğitsel amaçları aşağıdaki gibi sayılabilir:

- Öğrencilerin okul dışında bilgisayarlı teknolojileri kullanarak geçirdikleri zamanı daha verimli ve maharetli kullanmalarını sağlamak
- Çocukların yeni teknolojilerde hayal güçlerini geliştirebilmeleri ve yaratıcı olabilmeleri için imkanları artırmak
- Farklı kültürlerdeki farklı dili konuşan çocukların çevrimiçi platformda işbirliği yapabilmelerine olanak tanımak
- Çocuklara bir işi tamamlayabilme, başarıya duygusunu deneyimlemesine imkan tanımak

Scratch programlama aracının temel özellikleri aşağıda kısaca açıklanmıştır:

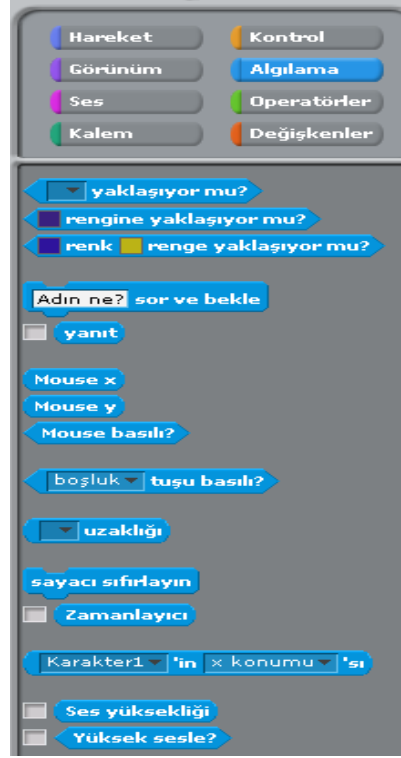
Basit arayüze sahip olma: Scratch programı kolay anlaşılabilir bir arayüze sahiptir. Her yaşta insana hitap etmektedir. Çocukların kullanabileceği basitlikte ancak üniversitede programlama dersinde kullanılabilecek düzeyde tasarlanmıştır. Basit düzeydeki kodlar yüksek düzeyde projeler geliştirilebilir (Resnick ve diğerleri, 2009). Scratch programı çalıştırıldığında ekranda görünen arayüz şekil 1'deki gibidir.



Şekil 2. Scratch Programlama Aracının Arayüzü

Şekil 2’de görülen programın arayüzü; kod blokları bölümü, kod bloklarının birleştirildiği kodlama alanı, oluşturulan projelerin izlendiği sahne alanı, projede kullanılan ve oluşturulan karakterlerin belirtildiği karakterler alanı ve araç çubuğu bölümlerinden oluşur.

Blok tabanlı programlama: Scratch’te hazır bulunan kod blokları işlev özelliklerine göre gruplanmış ve aynı gruptaki kodlar aynı renkte belirtilmiştir. Örneğin “Hareket” bloğu mavi renktedir. Sürükle bırak yöntemi ile bile bloklar kodlama alanında bir yapboz gibi birleştirilirler. Programlama dillerinin karmaşık kod yapısından kıyasla daha çok günlük konuşma dilindeki ifadeler göre tasarlanmıştır. Şekil 3’te Scratch programlama aracında algılama kodlarının bulunduğu kod paneli görülmektedir.



Şekil 2. Scratch'in Kod Panelindeki Algılama Kodlarının Görünümü

Çoklu dil desteği : Scratch'in, 150'den fazla ülkede ve 40'tan fazla dil seçeneğinin olması çocukların dil engeline takılmadan işbirliği yapabilmelerini destekler (Scratch, 2019a).

Çoklu ortam desteği: Scratch'te tasarlanan proje sahnede bölümünde canlandırılmaktadır. Proje çalıştırıldığında hem kodlama bölümündeki kodlar hem de oluşturulan hikâye aynı anda izlenebilir (Maloney ve diğerleri, 2010). Scratch programında bulunan dinamik, görsel ve işitsel öğeler programa çoklu ortam özelliği katmaktadır.

Hata ayıklama kolaylığı: Scratch'te hazır blok kod yapısının olması kod yazma işini ortadan kaldırırken hatalı yada eksik yazım, yanlış noktalama işaretleri gibi hataları da ortadan kaldırmaktadır. Ayrıca kod ezberlemeye de gerek kalmamıştır. Birbiriyle uyumlu olmayan kod bloklarının yapboz gibi kenetlenememeleri, kodları birleştirirken hata yapmanın da önüne geçmektedir. "Adım adım çalışmayı başlat" seçeneği ile olası mantık hataları hemen fark edilmektedir

Akıllı cihaz tasarımı: Scratch programı oyun, animasyon tasarımı ile öğrenmeyi sağlayan bir programlama aracıdır (Kafai ve diğerleri, 2007; Scratch, 2019a). Ancak oyun ve animasyon tasarımının yanı sıra bilgisayara bağlanabilen bir takım donanımların programlanmasıyla robot gibi cihazların kontrolü de gerçekleştirilebilir (Kasalak, 2017).

Kafai (2006) 'ye göre öğrencinin tasarım sürecine aktif katılımı motivasyonu artırmakta ve öğrenmeyi kolaylaştırmaktadır. Scratch programlama aracı ile öğrenci öğrenme sürecine aktif olarak katılabilecektir. Dolayısıyla öğretim sürecinde bir programlama aracı ile kendi oyununu tasarlayabilen öğrencinin güdülenme düzeyinin artması beklenebilir.

Paylaşım ve işbirliğini sağlaması: Scratch prgramında hazırlanan projeler programın kendi web sitesi üzerinden paylaşılabilir. Web sitesi üzerinde çevrimiçi gruplar oluşturulabilmekte, birlikte projeler yapılabilir ve görüş alışverişinde bulunabilmektedir (Scratch, 2019a)

Programlama yapılarına uygunluk: Scratch programının temel amacı programlama mantığını kazandırmaktır. Scratch, programlamanın temel yapıları olan algoritma, değişken, döngü gibi yapılar kod blokları şeklinde bulunmaktadır. Bu blokları kullanarak temel yapıların öğrenilmesi amaçlanmaktadır (Utting ve diğerleri, 2010). Scratch, çocuklara programlamayı sevdirmek ve öğretmek için geliştirilmiş olsa da Harvard Üniversitesi, Berkley Üniversitesi ve California Üniversitesi gibi birçok üniversitede programlamaya giriş derslerinde kullanılmaktadır (Resnick ve diğerleri, 2009). Yapılan araştırmaya göre, Scratch, üniversitelerde temel programlama derslerinde kullanılabilir (Rizvi, Humphries, Major, Jones ve Lauzun, 2011; Westcott, 2008; Yükseltürk ve Altıok; 2015; Erol, 2015). Ayrıca Papert'in yapılandırmacı kuramına dayanarak geliştirilen Scratch; kolay arayüzü, oyun tasarım olanağı sağlaması ve programlama yapılarına uygunluğu gibi özellikleri ile programlama öğretiminde kullanılabilir (Resnick ve diğerleri, 2009; Resnick, 2012).

2.8 İLGİLİ ARAŞTIRMALAR

Bu bölümde Scratch programlama aracının programlama başarısı, 21. yüzyıl becerilerine katkısı, öz yeterlik algısı ve güdülemeye etkisi ile ilgili daha önceden yapılmış ile ilgili araştırmalar bulunmaktadır.

2.8.1 Scratch'in Programlama Başarısı ve Güdülenmeye Etkisi İle İlgili Araştırmalar

Çelik ve Özdenler (2019) bilgisayarlı ve bilgisayarsız programlama etkinliklerinin öğrencilerin güdülenme düzeylerine etkisini araştırmışlardır. Çalışma iki 6. sınıf şubesinde deney ve kontrol grubu olarak belirlenen öğrencilerle 7 haftalık bir uygulama sürecidir. Deney gurubunda bilgisayar sınıfındaki bilgisayarlarda programlama kavramları Scratch programı ile öğretilirken, kontrol grubunda ise programlama kavramları önceden hazırlanmış çalışma kâğıtları ile bilgisayarsız ortamda öğretilmiştir. Çalışmanın sonunda; deney ve kontrol gruplarının güdülenme düzeyleri artmıştır ancak gruplar arasında güdülenme puanları açısından anlamlı bir farklılık bulunmamıştır.

Erol (2015) üniversite öğrencileri ile yaptığı çalışmada Scratch ile programlama öğretiminin öğrencilerin motivasyon ve programlama başarısına etkisini araştırmıştır. Araştırma iki aşamadan oluşmaktadır. Birinci aşamada programlama mantığının kazandırılması ve temel programlama yapılarının öğretilmesi amaçlanmıştır. Bu amaçla deney grubundaki katılımcılar Scratch ile oyun tasarımı etkinlikleri, kontrol grubunda yer alan katılımcılar ise mevcut ders programındaki haliyle akış diyagramları ile problem çözme etkinlikleri yapmışlardır. Uygulamanın ikinci aşamasında ise hem kontrol hem de deney grubunda aynı yöntem kullanılarak C# programlama dili öğretimi gerçekleştirilmiştir. Uygulama sürecinde başarı testi, güdülenme ölçeği ve görüşme formları kullanılmıştır. Araştırma sonunda; öğrencilerin motivasyon puanları öntestte her iki grupta da benzer, sontestlerde ise deney grubu lehine anlamlı farklılık bulunmuştur. Kontrol grubunda katılımcıların motivasyon puanlarının tüm uygulama sonunda azaldığı, deney grubunda ise arttığı görülmüştür. Katılımcıların programlama başarı puanları ele alındığında öntestte her iki grupta da benzer olduğu, sontest ve sontest 2' de ise deney grubu lehine anlamlı bir farklılık olduğu görülmüştür. Araştırmanın nitel verilerine göre; deney grubunda

yer alan katılımcılar Scratch ile oyun tasarımı etkinliklerinin eğlenceli ve kolay olduğunu, ders süresince yapılan etkinliklerin programlama mantığını kazandırmada ve motivasyonu artırmada etkili olduğunu, ancak bazı temel yapılar için yetersiz olduğundan bahsetmişlerdir. Kontrol grubunda yer alan katılımcılar ise akış diyagramları ile problem çözme sürecinin zor ve sıkıcı olduğunu, ders süresince yapılan etkinliklerde ise aktif olamama ve uygulamanın olmaması gibi sınırlılıkların motivasyonlarını düşürdüğünü dile getirmişlerdir.

Yükseltürk ve Altıok (2015) araştırmaları kapsamında 2013-2014 öğretim yılında Kırıkkale Üniversitesi Bilgisayar ve Öğretim Teknolojileri Eğitimi Bölümünde bir seçmeli ders kapsamında 169 Bilişim Teknolojileri Öğretmen adayı ile çalışmışlardır. Katılımcıların oyun konularını özgürce seçmeleri ve bireysel olarak projelerini Scratch ortamında geliştirmeleri beklenmiştir. Projeler hedef kitlesi, konusu ve oyunun sahip olması gereken özelliklere göre belirlenen değerlendirme puanına göre değerlendirildiğinde oyunların ilköğretim seviyesinde hazırlandığı ve kalitesinin genel olarak “Orta veya İyi” olduğu görülmüştür. Öğretmen adayları; programlama becerilerinin zor ve karmaşık olmasından dolayı Scratch programının yeni başlayanlar için daha ilgi çekici, kolay, heyecan verici ve eğlenceli ortam sunan bir görsel programlama dili olarak kolaylaştırıcı etkisinin olduğu belirtilmiştir. Kendilerini bu projede başarılı gören öğretmen adayları ile yapılan görüşmelerde, öğretmenler kendilerini başarılı görmelerinin sebebini “Tamamlayabilme ve kendine ait bir ürün ortaya koymanın çok güzel bir duygu olması” şeklinde ifade etmişlerdir.

Çatlak, Tekdal ve Baz (2015) Scratch programlama ortamının programlama öğretiminde kullanımına yönelik yapılan mevcut çalışmaları araştırmak üzere doküman inceleme yöntemi kullanılarak bir alanyazın taraması yapmışlardır. Bu kapsamda ulaşılan 53 makalenin 21 tanesi çeşitli nedenlerden dolayı çalışma kapsamından çıkarılmış, 32 makale ile çalışma yapılmıştır. Araştırmanın bulgularına göre, Scratch yazılımının programlama öğretiminde etkili olduğu ve programlama öğrenimini zevkli ve daha anlaşılır hale getirdiği belirtilmiştir. Ayrıca programlama derslerine, özünde oyun teması barındıran Scratch yazılımı ile başlamanın katılımcıların derse olan ilgi ve motivasyon değişkenleri üzerinde olumlu etkilerine dair bulgulara ulaşıldığı da belirtilmiştir.

Koorsse, Ciliers ve Calitz (2015) ortaöğretim öğrencileri ile yaptıkları çalışmada programlamaya yardımcı araçların öğrencilerin programlama kavramlarını

öğrenmelerine etkisini araştırmışlardır. Delphi programlama dili öğretim yapılan üç deney grubu oluşturulmuştur. Delphi'nin yanında Scratch, B# ve Robomind programları bir yıl boyunca üç farklı deney grubunda öğretim materyali olarak kullanılmıştır. Kontrol grubu ise yalnızca Delphi Programlama dili ile öğretime devam etmiştir. Araştırma sonunda Scratch, B# ve Robomind programlama ortamlarının öğrenci başarısını artırdığını ancak kontrol grubu ile arasında anlamlı bir fark olmadığı görülmüştür. Ancak temel programlama yapılarından olan döngü kavramını deney gruplarının daha iyi kavradığı görülmüştür. Öğrenciler üç programlama ortamını da kullanışlı olarak gördüklerini ayrıca Scratch programının temel programlama yapılarını öğrenmede daha kolaylaştırıcı etkisi olduğunu ifade etmişlerdir.

Calder (2010) yaptığı çalışmada matematiksel kavramların öğretiminde 6. Sınıfta öğrenim gören öğrencilere Scratch ile oyun tasarlama etkinlikleri yaptırmıştır. Etkinlik sürecini ve bu süreçte öğrencileri gözlemleyerek görüşmeler yapmıştır. Çalışmanın sonunda Scratch'in tasarlama etkinlikleri ile matematiksel kavramların öğrenilmesinde güdüleyici etkisi olduğunu, ayrıca matematiksel, mantıksal düşüncenin geliştirilmesinde Scratch programının kullanılması gerektiğini önermiştir.

Nikou ve Economides (2014) lise öğrencileri ile yaptıkları çalışmada öğrencilerin bilişim teknolojileri alanında öğrenimini teşvik etmek amacıyla yenilikçi yöntemlerden Scratch ve API Inventor Android görsel programlama araçlarının öğrencilerin programlamaya ilişkin motivasyonlarına etkisini incelemiştir. Çalışma sonucunda her iki programlama ortamının da programlamaya ilişkin ilgiyi artırdığı görülmüştür. Hem Scratch hem de API Inventor araçlarının çalışma sonunda öğrencilerin programlama öğretiminde içsel hedef yönelimi, görev değeri, öğrenme kontrol inancı ve öz yeterlik algısı motivasyonlarını artırdığı ancak dışsal motivasyonu değiştirmediği sonucuna ulaşılmıştır.

Adleberg (2013) 98 ortaokul öğrencisi ile yaptığı çalışmada Scratch ortamında oyun tasarlama etkinliklerinin kız öğrencilerin bilişim konularına yönelik olumsuz tutumlarına etkisini araştırmıştır. Scratch ile öğretim etkinliklerinden sonra öğrencilerin bilgisayar konularına yönelik tutum açısından cinsiyet değişkenine göre anlamlı bir farklılık olmadığı sonucuna ulaşmıştır. Araştırmacı etkinlik sürecindeki gözlemlerinde; kız öğrencilerin akran onayı almaları, öğretmen ve ebeveynlerin

olumlu söylemleri ve teşvikleri ile etkinliklerde başarılı olmalarının güdülenme düzeylerini etkilemiş olabileceği sonucuna ulaşmıştır.

2.8.2. Scratch'in 21. Yüzyıl Becerilerine Etkisi İle İlgili Araştırmalar

Yecan ve diğerleri (2017) yaptıkları çalışmada bilişim teknolojileri dersinde hali hazırda görsel programlama öğretimi yapan öğretmenlerin gözünden ilköğretimde görsel programlama öğretimini incelemişlerdir. Araştırmacılar öğretmenlerle yüz yüze görüşmeler sonrasında bir anket geliştirilerek uygulanmıştır. Araştırma sonucunda; programlama eğitiminin başlangıcında tüm öğretmenlerin görsel programlama aracı olarak Scratch'i tercih ettiği görülmüştür. Öğretmenler programlama öğretilerinin nedeni olarak; uluslararası eğilimler, devlet büyüklerinin programlamaya yönelik söylemleri, yıllar içinde farklılaşan öğrenci profili gibi etkenlerden bahsetmişlerdir. Görsel programlamayı tercih etmelerinin nedeni olarak; öğretmenlerin %54'ü "Öğrencileri üretime ya da yaratıcılığa teşvik edeceğini", %46'sı ise "Programlama öğrencilere gündelik hayatlarında kullanabilecekleri bir düşünme biçimi sağlıyor" şeklinde görüşlerini belirtmişlerdir.

Doğan ve Kert (2016) bilgisayar oyunu geliştirme sürecinin eleştirel düşünme becerilerine ve algoritma başarılarına etkisini araştırdığı çalışmasında ortaokul 6. sınıfta öğrenim gören 54 öğrenci ile çalışmıştır. Araştırma sonucunda, bilgisayar oyunu geliştirme süreci ile algoritma ve programlama derslerini yapmanın, geleneksel yöntem ile yapılan derslere göre eleştirel düşünme becerilerini ve algoritma başarılarını anlamlı düzeyde olumlu etkilediği görülmüştür.

Genç ve Karakuş (2011) çalışmalarında öğrencilerin Eğitimde Bilgisayar Oyunları tasarımı dersinde matematiksel ve kompütasyonel becerilerini geliştirmek ve tasarım sürecinde aktif katılımlarını sağlamak amacıyla Scratch programı kullanarak öğrenciler ile oyun tasarım etkinlikleri yapmıştır. Çalışmaya Fırat Üniversitesi Bilgisayar ve Öğretim Teknolojileri Eğitimi (BÖTE) bölümü 2. sınıfta eğitim gören 109 öğrenci katılmıştır. Veriler öğrencilerin uygulama sürecinde oluşturdukları blog yazılarından toplanmıştır. Çalışma sonunda öğrenciler Scratch'in keyifli bir ortam olduğunu, temel programlama yapılarının öğrenilmesinin metin tabanlı programlama dillerine göre Scratch'te daha kolay olduğunu, temel programlama yapıları için uygun fakat ileri düzey programlamacılar için uygun olmadığı şeklinde görüşte

bulunmuşlardır. Ayrıca öğrenciler tasarım etkinliklerinde etkin olmaktan mutlu olduklarını ve programlamaya yönelik motivasyonlarının yüksek olduğunu ifade etmişlerdir.

Tekerek ve Altan (2014) ortaokul 6. sınıfta eğitim gören 60 öğrenci ile yaptığı çalışmalarında Scratch'in algoritma (temel programlama) öğretimine etkisini incelemiştir. Araştırmada öğrenciler deney ve kontrol grubu olarak ayrılmış ve deney grubunda algoritma öğretimi, Scratch programlama ortamı ile kontrol grubunda ise düz anlatım yöntemiyle gerçekleştirilmiştir. Çalışma sonunda her iki grupta da öğrenci başarısının arttığı görülmüştür ancak son test puanları karşılaştırıldığında deney ve kontrol grubu arasında bir farklılık bulunmamıştır. Aynı şekilde cinsiyet değişkeni açısından kontrol ve deney gruplarının son test puanları karşılaştırıldığında anlamlı bir farklılık bulunmamıştır.

Kalelioğlu ve Gülbahar (2014) ilköğretim 5. sınıfta öğrenim gören 49 öğrenci ile yaptıkları çalışmada Scratch ile programlama öğretiminin öğrencilerin problem çözme becerilerine olan etkisini incelemiştir. Beş hafta süren uygulamada öğrenciler basit programlar ve oyunlar tasarlamışlardır. Çalışma sonunda öğrencilerle Scratch ile programlama öğretimi hakkında odak grup görüşmeleri yapılmıştır. Uygulamanın sonunda öğrencilerin ön test ve son test problem çözme puanları arasında; öz güven, öz denetim ve kaçınma alt boyutlarında anlamlı bir farklılık olmadığı görülmüştür. Nitel veriler incelendiğinde öğrencilerin Scratch programını kolay bulduklarını, Scratch ile programlama sürecini sevdiklerini ve programlamayı devam ettirme ve geliştirmede istekli oldukları belirlenmiştir.

Kim, Chung ve Yu (2013) ortaokul öğrencileri ile yaptıkları çalışmada programlama eğitimi kapsamında Scratch ile programlamanın öğrencilerin yaratıcı problem çözme becerilerine olan etkisini araştırmışlardır. Çalışma 119 normal ve 30 üstün zekalı öğrenci ile yürütülmüştür. Bu bağlamda hem normal çocuklarda hem de üstün yetenekli çocuklar için ayrı ayrı Scratch ile tasarım etkinlikleri yapılmıştır. Mantıksal yeteneği, bilimsel problem çözme yeteneği ve farklı düşünme yeteneği arasında yüksek korelasyona sahip iki yapay yaratıcı problem çözme testleri uygulanmıştır. Çalışma sonunda yaratıcılığın her grup için yükseldiği görülmüştür. normal çocuklar açısından yaratıcı problem çözme becerileri incelendiğinde deney ve kontrol grubu arasında anlamlı farklılık görülmezken; üstün yetenekli çocuklar açısından deney grubu lehine anlamlı farklılık ortaya çıkmıştır.

Westcott (2008) üniversite öğrencileri ile yaptığı çalışmasında Scratch ile programlama öğretiminin öğrencilerin C++ programlama dilleri başarısına olan etkisini araştırılmıştır. C++ programlama dili öğretiminde destek için deney grubunda ise C++ kodlarının Scratch’de karşılığını öğrenmişler ve Scratch’te tasarım etkinlikleri yapılırken, kontrol grubunda problem çözümü etkinlikleri yapılmıştır. Uygulama sırasında her iki gruba da başarı testleri uygulanmıştır. Bu testlerde C++ kodlarının yanı sıra algoritma problemleri sorulmuştur. Çalışma sonunda programlamanın temel yapıları ile ilgili testte gruplar arasında fark bulunmazken; programlamanın döngü gibi daha karmaşık yapılarının yer aldığı testlerde deney grubu lehine anlamlı farklılık bulunmuştur. Çalışma sonunda Scratch ile yapılan etkinliklerin öğrencilerin algoritmik düşünme becerilerini geliştirdiği ayrıca bunu başka bir programlama diline aktarabildikleri vurgulanmıştır.

2.8.3. Scratch’in Öz-yeterlik Algısına Etkisi İle İlgili araştırmalar

Kasalak (2017) robotik kodlama etkinliklerinin ortaokul öğrencilerinin programlamaya ilişkin öz yeterlik algılarına etkisini araştırdığı çalışmasında “blok temelli programlamaya ilişkin öz-yeterlik algısı ölçeği” geliştirmiştir. Ölçek basit blok temelli programlamaya ve karmaşık blok temelli programlamaya ilişkin olmak üzere iki faktörlüdür. 5 haftalık robotik kodlama etkinlikleri planlamış ve bu etkinlikleri bir devlet okulundaki 58 öğrenciyle uygulanmıştır. Araştırma sonucunda basit ve karmaşık blok temelli programlamaya ilişkin öz-yeterlik algısı puanlarına bakıldığında cinsiyet değişkeni açısından anlamlı fark görülmemiştir. Daha önce Scratch’le programlama dersi alma ve daha önce Scratch’le program yazma durumlarına göre öğrencilerin basit blok temelli programlamaya ilişkin öz yeterlik algılarında anlamlı bir fark görülmemiştir. Daha önce Scratch’le programlama dersi alan öğrencilerin karmaşık blok temelli programlamaya ilişkin öz yeterlik algısı puanları daha önce Scratch’le programlama dersi almayan öğrencilere göre daha önce Scratch’le program yazan öğrencilerin karmaşık blok temelli programlamaya ilişkin öz yeterlik algısı puanları daha önce Scratch’le program yazmayan öğrencilere göre anlamlı derecede yüksek olarak bulunmuştur. Halen Scratch’le programlama dersi alan öğrencilerin halen Scratch’le programlama dersi almayan öğrencilere göre, halen Scratch’le program yazan öğrencilerin halen Scratch’le program yazmayan öğrencilere göre hem basit hem de karmaşık blok temelli programlamaya ilişkin öz-

yeterlik algısı puanları anlamlı derecede yüksek olduğu sonucuna ulaşılmıştır. Çalışmada uygulanan etkinliklerin öğrenci yaşantılarına etkisi incelendiğinde hem kız hem erkek öğrenciler etkinlikleri oldukça yüksek oranda eğlenceli, ilgi çekici bulmuşlar ve etkinliklere katılmak konusunda istekli olmuşlardır.

Chiu (2014) yaptığı çalışmasında Scratch ile programlama öğretiminin lise öğretmenlerinin programlama öz yeterliklerine etkisini araştırmıştır. Programlama bilgisi ve deneyimi olmayan öğretmenler Scratch ile oyun ve hikâye tasarımı geliştirmişlerdir. Çalışma sonunda öğretmenlerin programlamaya ilişkin öz yeterlik öntest ve sontest puanları arasında bir farklılık bulunmamıştır. Ancak öğretmenlerin uygulamaya süreci ile ilgili görüşleri incelendiğinde Scratch'in programlama için uygun bir araç olduğunu, ilgi çekici olduğunu ve kodların sürükle bırak özelliğine sahip olmasının öğrenmeyi kolaylaştırıcı etkisi olduğunu, programlama sürecine ilişkin pozitif tutuma sahip olduklarını belirtmişlerdir.

Rizvi ve diğerleri (2011) üniversite birinci sınıf öğrencileri ile yaptıkları çalışmalarında CS0 dersinde Scratch programının öğrencilerin programlamaya ilişkin öz yeterliklerini, programlamaya yönelik tutumlarını ve ikinci sınıftaki CS1 dersinde programlama başarılarına etkisini araştırmıştır. Araştırma iki yıl sürmüştür. Birinci yıl Bilgisayar Bilimi dersinin başlangıcı olan algoritma öğretiminin yapıldığı CS0 dersinde öğrencilere Scratch ile oyun, animasyon, hikâye tasarım etkinlikleri yaptırılmıştır. İkinci yıl CS0 dersini alan Scratch grubu ve CS0 dersini almayan grup programlama eğitimine devam etmişlerdir. Çalışmanın sonunda Scratch öğrenimi gören grubun CS0 dersinde programlamaya yönelik öz yeterlikleri ve tutumu anlamlı şekilde artmıştır. CS1 dersindeki programlama başarılarının ise Scratch öğrenimi gören grubun Scratch öğrenimi görmeyen gruba göre daha çok arttığı görülmüştür. Araştırmacılar temel programlama yapılarının öğretildiği bir derste Scratch 'in kullanılmasının gerektiğini vurgulamışlardır.

2.9 ALANYAZIN TARAMASININ SONUCU

Alanyazın incelendiğinde Scratch programlama aracı ile ilgili birçok çalışma bulunmaktadır. Yapılan çalışmaların genellikle deneysel çalışmalar olduğu, bunun yanında durum çalışmalarının da yer aldığı görülmektedir. Bu araştırmalarda

programlama öğretiminde genellikle programlamayı sevdirmek amacıyla Scratch aracının kullanıldığı görülmektedir. Programlama öğretiminde Scratch kullanımına ilişkin çalışmaların genellikle K-12 düzeyindeki çocuklara yönelik yapıldığı bunun yanı sıra üniversite düzeyindeki yetişkinlerle yapılan çalışmaların da yer aldığı görülmektedir. Çalışmalarda genellikle Scratch ile oyun tasarımını etkinliklerinin programlama başarısına etkisi incelenmiş, bazı çalışmalarda ise başarının yanında motivasyona etkisi de incelenmiştir (Gülmez, 2009). Bununla birlikte Scratch'in eleştirel düşünme becerileri (Doğan ve Kert, 2016), kompütasyonel becerileri (Genç ve Karakuş, 2011), yaratıcılık, problem çözme becerileri (Kalelioğlu ve Gülbahar, 2014; Kim ve diğerleri, 2013) gibi 21. yüzyıl becerilerine ilişkin etkileri incelenmiştir. Scratch ile yapılan programlama öğretiminin bu becerilere genellikle olumlu etkisi olduğu görülmektedir.

Ancak alan yazında Scratch ile programlama etkinliklerinde elde edilen deneyim ve bilgilerin yeni bir programlama dili öğretiminde devam edip etmediği ya da transfer edilip edilmediğiyle ilgili yeterli çalışma olmadığı görülmektedir (Ozoran ve diğerleri, 2012; Rivzi ve diğerleri, 2011; Wescott, 2008).

BÖLÜM III

YÖNTEM

Bu bölümde araştırmanın modeli, çalışma grubu, veri toplama araçları, verilerin analizine ilişkin bilgiler yer almaktadır.

3.1 ARAŞTIRMANIN MODELİ

Bu çalışmada Scratch programlama aracıyla yapılan programlama öğretiminin, öğrencilerin programlama başarısı, derse yönelik güdülenme düzeyi ve algoritma geliştirmeye yönelik algıladıkları öz yeterlik düzeylerine etkisi araştırılmıştır. Araştırmada öntest - sontest kontrol gruplu yarı deneysel model kullanılmıştır.

Deneysel desenli çalışmalar değişkenler arasındaki nedenselliği test etmeyi amaçlayan, araştırmacı tarafından değişken veya değişkenlerin kontrol altına alınabildiği ve sonuçların gözlemlendiği çalışma türleridir (Büyüköztürk, Çakmak, Akgün, Karadeniz ve Demirel, 2011).

Öntest-sontest kontrol gruplu modele göre oluşturulan çalışmanın bağımlı değişkenlerini programlama başarısı, algoritma geliştirmeye ilişkin öz yeterlik algısı ve güdülenme oluştururken; bağımsız değişkenini Scratch ile programlama öğretimi veya önceden uygulanan öğretim yöntemiyle dersin işlenmesi oluşturmaktadır.

Araştırmada kullanılacak ön test- son test kontrol gruplu yarı deneysel desen tablo 1’de verilmiştir.

Tablo 1. Araştırmada Kullanılacak Ön Test- Son Test Kontrol Gruplu Yarı Deneysel
Desen

Gruplar	Öntest	Uygulama	Sontest
Deney	BT _{ÖT}	SPÖ	BT _{ST}
	GÖS _{ÖT}		GÖS _{ST}
	AGÖ _{ÖT}		AGÖ _{ST}
Kontrol	BT _{ÖT}	GAPÖ	BT _{ST}
	GÖS _{ÖT}		GÖS _{ST}
	AGÖ _{ÖT}		AGÖ _{ST}

SPÖ : SCRATCH Aracı ile Programlama Öğretimi

GAPÖ: Geleneksel Anlatım Yöntemi ile Programlama Öğretimi

BT_{ÖT} : Başarı Testi – Öntest

GÖS_{ÖT} : Güdülenme ve Öğrenme Stratejileri Ölçeği– Öntest

AGÖ_{ÖT} : Algoritma Geliştirmeye İlişkin Özyeterlik Algısı Ölçeği Öntest

BT_{ST} : Başarı Testi – Sontest

GÖS_{ST} : Güdülenme ve Öğrenme Stratejileri Ölçeği – Sontest

AGÖ_{ST} : Algoritma Geliştirmeye İlişkin Özyeterlik Algısı Ölçeği Sontest

3.2 ÇALIŞMA GRUBU

Araştırmanın katılımcılarını Sakarya ili merkez ilçedeki bir ortaokulda, 2016-2017 eğitim öğretim yılı 2. döneminde 6. sınıfta okuyan 64 öğrenci oluşturmaktadır. Okulda bulunan 7 tane 6. sınıf şubesinden akademik başarı, sınıf mevcudu ve cinsiyet dağılımı özellikleri bakımından birbirine en yakın olan iki şube çalışma grubu olarak seçilmiştir. Katılımcı olarak; 15 kız ve 17 erkek öğrenci deney grubunda, 15 kız ve 17 öğrenci kontrol grubunda bulunmaktadır.

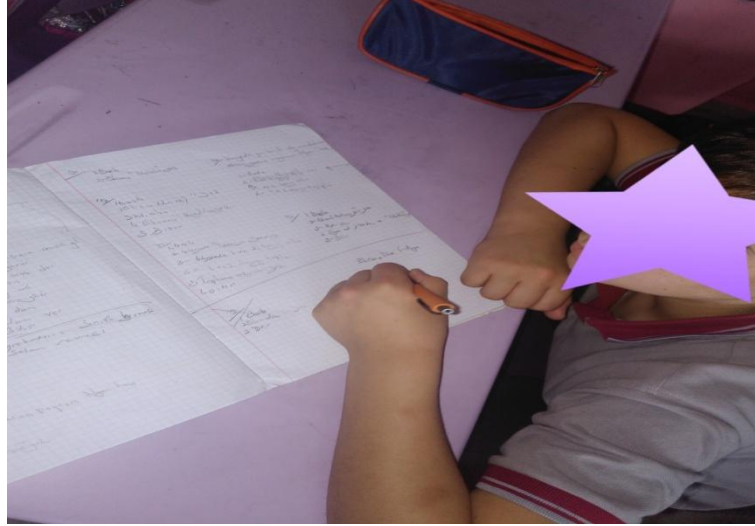
3.3 DENEYSEL İŞLEMLER

Deneysel işlemler, bilişim teknolojileri ve yazılım dersinde 4 haftalık süreçte gerçekleştirilmiştir. “Problem Çözme, Programlama ve Özgün Ürün Geliştirme” ünitesinde “Problem Analiz ve Çözme Yaklaşımları Programlama” ve “Algoritma ve Strateji Geliştirme” öğrenme alanlarının temel amacı temel programlama mantığının öğretilmesidir. Ders içeriğinde ise algoritma ve akış diyagramları başta olmak üzere, değişken, sözde kod, sabit, girdi ve çıktı eylemleri, döngüler, karar verme yapıları, gibi temel programlama yapıları yer almaktadır (MEB, 2012).

Çalışmanın amacı doğrultusunda 6. sınıfta öğrenim gören, ilk kez programlama eğitimi alacak 64 öğrenciyle uygulama gerçekleştirilmiştir. Programlamaya ilişkin temel kavramlar ve programlama mantığının öğretimi; kontrol grubunda yer alan 32 öğrenciye geleneksel anlatım yöntemi ve akış diyagramlarıyla; deney grubunda yer alan 32 öğrenciye ise Scratch programlama ortamında yapılan etkinliklerle gerçekleştirilmiştir.

3.3.1 Kontrol Grubu Ders İçeriğinin Hazırlanması ve Uygulanması

Kontrol grubunda öğrencilerin programlama mantığını ve temel programlama yapılarını (problem, algoritma, değişken, karar verme ve kontrol, döngü gibi) mevcut ders içeriğinde olduğu şekliyle yani akış diyagramları kullanılarak öğrenmeleri amaçlanmaktadır. Bu amaç doğrultusunda; ders içeriği MEB’in (2012) Bilişim Teknolojileri ve Yazılım dersi öğretim programındaki ders içeriğine göre belirlenmiş ve bu kazanımlara uygun olarak planlanan ders etkinlikleri uzman görüşü doğrultusunda hazırlanmıştır. Hazırlanan ders etkinlikleri temel programlama yapılarını kullanmayı gerektiren yapılandırılmış programlama problemlerini içermektedir. Öğrenciler günlük problemlerin çözümünde öncelikle canlandırma etkinliklerini, sonra da akış diyagramlarını kullanmışlardır. Hazırlanan ders içeriği uzman görüşü alındıktan sonra günlük planlar şeklinde düzenlenmiştir (EK 1). Kontrol grubunda etkinlik yapan öğrenci görseli şekil 4’te verilmiştir.



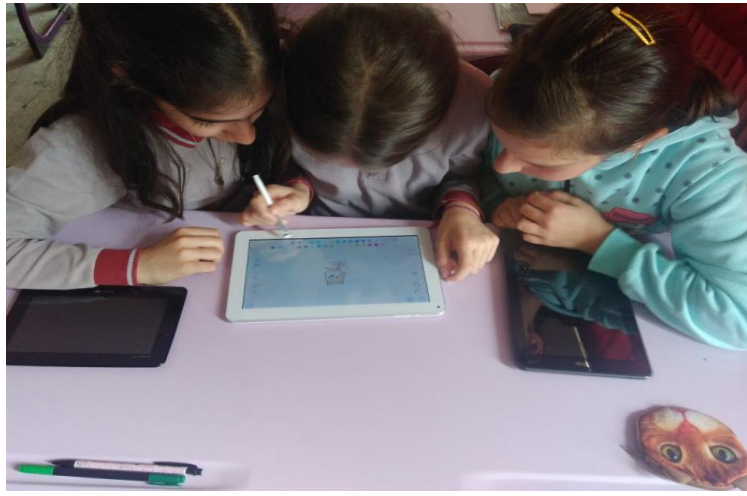
Şekil 3. Kontrol Grubu Öğrenci Görsele

Şekil 4'te görüldüğü gibi kontrol grubunda programlama problemini çözen öğrenciler günlük dilde algoritma adımlarını yazdıktan sonra akış şemaları çizerek etkinlik görevlerini tamamlamaktadırlar. Her dersin sonunda öğrenciler geliştirdikleri algoritmalar üzerinden rubrik formları ile değerlendirilmektedirler.

3.3.2 Deney Grubu Ders İçeriğinin Hazırlanması ve Uygulanması

Deney grubunda öğrencilerin programlama mantığını ve temel programlama yapılarını (problem, algoritma, değişken, karar verme ve kontrol, döngü gibi) Scratch programı kullanılarak öğrenmeleri amaçlanmaktadır. Bu amaç doğrultusunda; ders içeriği MEB'in (2012) bilişim teknolojileri ve yazılım dersi öğretim programındaki ders içeriğine göre belirlenmiş ve bu kazanımlara uygun olarak planlanan ders etkinlikleri uzman görüşü doğrultusunda hazırlanmıştır. Deney grubunda uygulanan ders kazanımları kontrol grubuna paralel olarak hazırlanmış ancak etkinlikler deney grubuna yönelik düzenlenmiştir. Öğrenciler bu problemleri çözmek için kendi tablet bilgisayarlarını uygulama ortamında kullanmışlardır. Öğrenciler algoritma geliştirmede Scratch programını kullanmışlardır. Bu nedenle programlama problemleri Scratch programında öğrencilere bir oyun senaryosu şeklinde sunulmuştur. Ayrıca her ders sonunda öğrencilerin tüm süreçlerine (senaryo, karakter, sahne, tasarım) kendilerinin karar verdiği ve dört hafta sonunda bitirmeleri

gereken bir oyun tasarımlamaları ve Scratch 'in kendi web sitesinde paylaşması için serbest etkinlik süresi verilmiştir. Bunun nedeni Scratch' in tasarım odaklı yapısı ile katılımcıların öğrenme sürecine aktif katılımını sağlayarak, oyun, öğrenme ve işbirliğini bir araya getirme olanağı sağlanmasıdır. Hazırlanan ders içeriği son olarak uzman görüşü doğrultusunda yeniden düzenlenmiş ve deneysel süreç öncesinde son şekli verilmiştir (EK 1). Hazırlanan ders etkinlikleri temel programlama yapılarını kullanmayı gerektiren yapılandırılmış programlama problemlerini içermektedir. Deney grubunda etkinlik yapan öğrenci görseli Şekil 5'te verilmiştir.



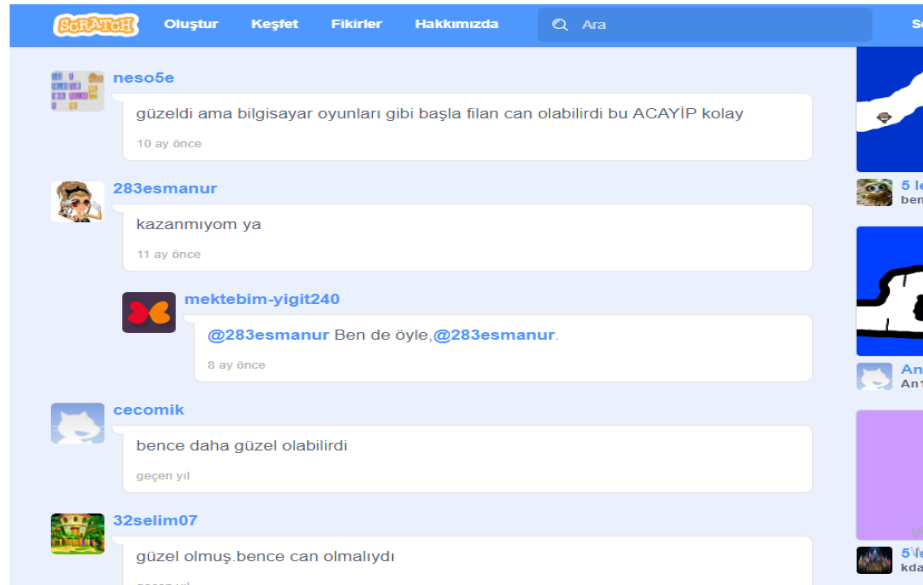
Şekil 4. Deney Grubu Öğrenci Görseli

Şekil 5'te görüldüğü gibi deney grubunda öğrenciler bir araya gelerek tablet bilgisayarlarındaki Scratch aracında programlama etkinliklerini yapmaktadırlar. Her dersin sonunda öğrenciler geliştirdikleri algoritmalar üzerinden rubrik formları ile değerlendirilmektedirler. Ayrıca deney grubundaki öğrencilerden işbirliği becerilerini geliştirmeleri amacıyla yaptıkları projeleri diğer çocuklarla paylaşmaları istenmiştir. Şekil 6'da deney grubundan bir öğrencinin Scratch'te tasarlayıp Scratch'in kendi web sitesinde diğer çocuklarla paylaştığı örnek bir oyun projesi bulunmaktadır.



Şekil 5. Deney Grubunda Hazırlanan Proje Örneği

Şekil 6 'da görülen oyun projesi beş aşamalı bir oyundan oluşmaktadır. Proje, 2043 kez diğer kullanıcılar tarafından görüntülenmiştir. Ayrıca deney grubunda öğrencilerden birbirlerinin projelerine dönütlerde bulunarak projelerini geliştirmeleri beklenmiştir. Şekil 7'de bir projenin altında yazılan öğrenci yorumları görülmektedir.



Şekil 6. Deney Grubunda Hazırlanmış Bir Projeye İlişkin Akran Dönütleri

Şekil 7’de öğrencilerin paylaşılan projeye ilişkin olumlu ve olumsuz görüşleri görülmektedir. Ayrıca öğrenciler projenin geliştirilmesine ilişkin önerilerde bulunmaktadırlar.

Deney ve kontrol gruplarının haftalık konu, etkinlik ve kazanımları Tablo 1’deki gibidir:

Tablo 2. Deneysel Sürecin Haftalık Konu, Kazanım ve Etkinlikler Tablosu

Tarih	Kontrol Grubu	Deney Grubu
27-31 Mart 2017	Öntestlerin uygulanması a. Başarı Testi b. Güdülenme ve Öğrenme Stratejileri Ölçeği c. Algoritma ve Strateji Geliştirme Özyeterlik Algısı Ölçeği	Öntestlerin uygulanması a. Başarı Testi b. Güdülenme ve Öğrenme Stratejileri Ölçeği c. Algoritma ve Strateji Geliştirme Özyeterlik Algısı Ölçeği Konu: Scratch programlama ortamı ile tanışma Etkinlik: Program tanıtılır ve scratch ile yapılmış bazı ilgi çekici projeler öğrencilere gösterilir. Kazanım: Scratch programının özelliklerini ve bu program ile yapabileceklerini sıralar. Basit bazı komutlar öğrenir.
1. Hafta 03-07 Nisan 2017	Konu: Bilgisayarda algoritma ve programlama kavramlarının açıklanması Etkinlik: Bir arkadaşlarını komut vererek yürütmeleri istenir. Günlük hayattan bir probleme yönelik algoritma oluşturmaları istenir. Kazanım: Öğrenciler algoritma ve yazılım kavramlarını açıklayabilir ve örnekler verebilir. Farklı algoritmaları inceleyerek en hızlı ve doğru çözümü seçer	Konu: Bilgisayarda algoritma ve programlama kavramlarının açıklanması Etkinlik: Başlangıç olarak kedi karakterinin yürümesi, koşması, bir arabanın hareket etmesi için gereken komutlarla uygulama yaptırılır. Serbest etkinlik için zaman verilir. Kazanım: Öğrenciler algoritma ve yazılım kavramlarını açıklayabilir ve örnekler verebilir. Farklı algoritmaları inceleyerek en hızlı ve doğru çözümü seçer.
2. Hafta 10-14 Nisan 2017	Konu: Akış şeması ve değişkenler kavramının açıklanması Etkinlik: Sabah kalkıp okula gelene kadar yaptığın işlerin algoritma ve akış şemasını oluşturması istenir. Verilen iki sayının toplamını bulup sonucu	Konu: Akış şeması ve değişkenler kavramının açıklanması Etkinlik: Sabah kalkıp okula gelene kadar yaptığın işlerin algoritma ve akış şemasını oluşturması istenir. Scratch programında, verilen iki sayının toplamını bulup sonucu gösteren bir akış

	gösteren bir akış şeması oluşturması istenir.	şeması oluşturması istenir.
	Kazanım: Akış şeması ve değişken kavramlarını açıklar. Değişken kullanarak basit matematik işlemlerine yönelik akış şeması oluşturur	Kazanım: Akış şeması ve değişken kavramlarını açıklar. Değişken kullanarak basit matematik işlemlerine yönelik akış şeması oluşturur.
3.Hafta 17-21 Nisan 2017	Konu: “Eğer ise” koşullu yapı mantığının açıklanması Etkinlikler: Klavyeden girilen bir sayının 50’den büyükse ekrana “büyük”, 50’den küçükse ekrana “küçük” yazan programın akış şeması oluşturması istenir. Klavyeden girilen sayının pozitif, negatif, sıfır olduğunu ekrana yazdıran programın algoritmasını yazması istenir. Kazanım: “Eğer ise” koşullu yapı mantığını örnek vererek ve akış şeması oluşturarak açıklar.	Konu: “Eğer ise” koşullu yapı mantığının açıklanması Etkinlikler: Klavyeden girilen bir sayının 50’den büyükse ekrana “büyük”, 50’den küçükse ekrana “küçük” yazan bir program yapması istenir. Scratch’te algılama bloğunu kullanarak örnek uygulamalarla (Labirent oyunu) koşul yapıları kullanılır. Klavyeden girilen sayının pozitif, negatif, sıfır olduğunu ekrana yazdıran programı Scratch’te yapması istenir. Kazanım: “Eğer ise” koşullu yapı mantığını örnek vererek ve akış şeması oluşturarak açıklar. Algılama bloğunu kullanarak örnek uygulamalar yapabilir.
4.Hafta 24-28 Nisan 2017	Konu: Döngüsel algoritmanın açıklanması Etkinlikler: Sayı tahmin oyununun algoritmasını yazması istenir. Kazanım: Döngü kavramını açıklar ve örnek uygulamalar gerçekleştirebilir. Farklı programların algoritmalarını inceleyerek en hızlı ve doğru çözümü seçer. Son testlerin uygulanması	Konu: Döngüsel algoritmanın açıklanması Etkinlikler: Scratch programında sayı tahmin oyununu yapması istenir. Kazanım: Döngü kavramını açıklar ve örnek uygulamalar gerçekleştirebilir. Farklı programların algoritmalarını inceleyerek en hızlı ve doğru çözümü seçer. Son testlerin uygulanması

Tablo 2’de deneysel süreçte deney ve kontrol gruplarının haftalık olarak kazanım ve etkinlikleri görülmektedir.

3.4 VERİ TOPLAMA ARAÇLARI

Araştırmada elde edilen verilerin toplanması için dört bölümden oluşan anket formu kullanılmıştır. Araştırmada kullanılan anket formunun birinci bölümünde

katılımcıların demografik özelliklerinin belirlenmesi için 3 adet soru formu, anketin ikinci bölümünde katılımcıların öz-yeterlik algısı düzeylerini belirlemek için beşli likert tipli ölçek, araştırmanın üçüncü bölümünde katılımcıların güdülenme düzeylerini ölçmek için yedili likert tipli ölçek sunulmuş olup araştırmanın son bölümünde katılımcılara başarı testi sunulmuştur. Veri toplama bölümlerine ait detaylı bilgiler aşağıda sunulmuştur.

3.4.1 Kişisel Bilgi Formu

Veri toplama aracı olan anket formunun ilk bölümünde öğrencilerin demografik düzeylerini belirlemek için sırası ile cinsiyet, sınıf ve yaş soruları sorulmuştur. Araştırmaya katılan öğrencilerin demografik özelliklerine ait dağılımları tablo 3’de verilmiştir.

Tablo 3. Öğrencilerin Demografik Özelliklerine Ait Dağılımları

Cinsiyet	f	%
Kız	30	46,9
Erkek	34	53,1
Yaş	f	%
11 veya altı	4	6,3
12 yaş	60	93,8
Sınıf	f	%
6.Sınıf	64	100,0
Toplam	64	100,0

Araştırmaya katılan öğrencilerin demografik özellikleri incelendiğinde, öğrencilerin %46,9’unun kız, %53,1’inin erkek olduğu, %6,3’ünün 11 yaş veya altı, %93,82inin 12 yaşında olduğu ve öğrencilerin tamamının 6. Sınıfta okudukları belirlenmiştir.

Tablo 4. Öğrencilerin Deney ve Kontrol Grubuna Ait Dağılımları

Grup	f	%
Deney	32	50,0
Kontrol	32	50,0
Toplam	64	100,0

Deney ve kontrol grubu öğrencilerin eşit dağıldığı, %50'sinin deney, %50'sinin kontrol grubundan oluşmaktadır.

3.4.2 Öz-yeterlik Algısı Ölçeği

Veri toplama aracı olan anket formunun ikinci bölümünde araştırmacı tarafından geliştirilen 16 maddelik beşli likert (1= Kesinlikle Katılmıyorum, 2= Katılmıyorum, 3=Ne katılıyorum ne katılmıyorum, 4=Katılıyorum, 5=Kesinlikle Katılıyorum) öz-yeterlik ölçeği güvenirlik ve geçerlilik analizleri için 171 öğrenciye uygulanmıştır. Ölçeğe ait geçerlilik güvenirlik analizi sonuçları Tablo 5'te verilmiştir.

Tablo 5. Öz-yeterlik Algısı Ölçeği Maddelerine İlişkin İstatistikler

Madde No	Madde Ortalaması	Madde Standart Sapması	Madde Toplam Korelasyonu	Madde Silme Güvenirlik Katsayısı
1	48,415	243,785	0,906	0,994
2	48,649	240,559	0,957	0,994
3	48,766	238,768	0,980	0,993
4	48,795	238,740	0,982	0,993
5	49,006	240,512	0,937	0,994
6	48,749	238,660	0,985	0,993
7	48,731	238,186	0,972	0,994
8	48,690	238,497	0,961	0,994
9	48,444	241,307	0,916	0,994
10	48,842	239,475	0,971	0,994
11	48,977	240,529	0,940	0,994
12	48,772	238,707	0,978	0,993
13	48,901	240,667	0,955	0,994
14	48,924	240,671	0,953	0,994
15	49,129	242,477	0,916	0,994
16	48,737	240,666	0,935	0,994

Cronbach's Alpha= 0,994

Tablo 5 incelendiğinde, 16 maddeden oluşan öz yeterlik ölçeğinde 0,45'in altında bulunan madde olmadığı belirlenmiştir. Madde-toplam korelasyon değeri 0,45'in altında madde olmamasından dolayı bu aşamada ölçekten madde çıkarımına gerek duyulmamıştır. Ölçeğin iç tutarlılığını belirlemek için Cronbach's Alpha analizi kullanılmış olup ölçeğin güvenirlilik seviyesinin yüksek düzeyde olduğu tespit edilmiştir ($\alpha=0,994$).

Bu aşamadan sonra değişkenlerin toplam ölçekle arasındaki ilişki incelenmiştir. Bu bağlamda $r > ,30$ düzeyindeki ilişkiler veri setinin faktör analizine uygunluğuna işaret etmektedir. Tablo 5 incelendiğinde, ölçek maddeleri ile toplam ölçek arasındaki

ilişkinin tamamının söz konusu ölçütü karşıladığı görülmektedir. Tablo 6'ya göre maddelerle toplam ölçek arasındaki ilişkiler ,917-,987 arasında olduğu ve tüm maddeler için ilişkilerin istatistiksel olarak anlamlı olduğu belirlenmiştir ($p<,01$). Bu bulgular ölçekteki maddelerin toplam puan ile ilişkisinin yeterli olduğunu ve maddelerde tutarlılık açısından problem olmadığını göstermektedir.

Tablo 6. Öz-yeterlik Algısı Ölçeğinin Madde ve Toplam Ölçek Korelasyonu Değerleri

<u>Madde No</u>	<u>R</u>	<u>p</u>
1	0,917	0,000**
2	0,962	0,000**
3	0,983	0,000**
4	0,984	0,000**
5	0,945	0,000**
6	0,987	0,000**
7	0,975	0,000**
8	0,966	0,000**
9	0,926	0,000**
10	0,974	0,000**
11	0,948	0,000**
12	0,981	0,000**
13	0,961	0,000**
14	0,958	0,000**
15	0,926	0,000**
16	0,943	0,000**

** $p<,01$

Öz-yeterlik algısı ölçeğine ait toplam 16 sorudan oluşan ölçeğin, faktör analizinin ön şartları olan değişkenler arasında belli oranda korelasyon bulunmasının sonucunda veri setinin faktör analizine uygunluğuna karar vermek amacıyla KMO değeri,

Barlett Küresellik testi ve deęişkenler arasındaki ilişkiler esas alınmıştır (Tabachnick ve Fidel, 2014). KMO deęerinin ,60'tan büyük olması veriler üzerinden faktör analizi yapılabileceğini göstermektedir (Büyüköztürk, 2007). Tablo 7'de özyeterlik ölçeğinin KMO ve Barlett testinin sonucu görülmektedir.

Tablo 7 : Öz-yeterlik Ölçeği KMO ve Barlett Testi Sonucu

İstatistik	Deęer	
KMO Örneklem Yeterlilięi	0,968	
Barlett Küresellik Testi	Ki-kare Deęeri (χ^2)	6932,485
	Serbestlik Derecesi (df)	170
	Anlamlılık Deęeri (p)	0,000

Tablo 7'de görüldüğü üzere ,968 (>,60) ve Barlett küresellik testi $p<0,01$ önem düzeyinde anlamlı bulunmuştur. Bu bulgular örneklem büyüklüğünün faktör analizi için uygun olduđu ve verilerin çok deęişkenli normal dağılımdan elde edildiğini göstermektedir (Kan ve Akbaş, 2005). Sonraki süreç olan AFA'da faktör çıkarımı için Temel Bileşenler Analizi kullanılmış, faktörlerin nasıl döndürüleceğinin belirlenmesi için de dikey döndürme yöntemlerinden olan varimax dik döndürme yöntemi tercih edilmiştir. Ölçekteki maddelerin kalması ya da kalmaması durumuna karar vermede faktör yük deęerlerinin ,45 veya daha üzeri bir deęer olması ölçüt olarak alınmıştır (Büyüköztürk, 2007). Bununla birlikte maddelerin tek bir faktör altında yük deęeri taşıma özellięi de dikkate alınmıştır. 16 maddelik ölçekte faktör analizi sonucu toplam varyansın %91,929'unu açıklayan tek faktörlü bir yapı ortaya çıktığı görülmüştür. Tablo 8'de yapılan analize ilişkin bulgular gösterilmiştir.

Tablo 8. Öz-yeterlik Ölçeğinin Öz Değerleri ve Açıkladıkları Varyans Düzeyi

Bileşenler	Başlangıç Öz değerleri			Döndürme Sonrası Yüklerin Kareler Toplamı		
	Toplam	Varyans%	Kümülatif %	Toplam	Varyans %	Kümülatif %
1	14,709	91,929	91,929	14,709	91,929	91,929

Tablo 8’da görüldüğü üzere öz yeterlik ölçeği öz değeri 1,00’den büyük tekfaktörlü bir yapı sergilemektedir. Birinci faktör tek başına toplam varyansın %91,929’unu açıkladığı belirlenmiştir. Söz konusu faktör yapısını doğrulamak amacıyla değerlendirilen diğer bir nokta ise ölçeğin çizgi yamaç (Scree plot test grafiği) grafiğidir. Aşağıdaki grafikte kırılmanın birinci boyuttan sonra gerçekleştiği ve tüm maddelerin faktör yapıları bakımından mantıksal bütünlük sağladığı açık bir şekilde görülmektedir.



Şekil 7. Öz-yeterlik Ölçeğine Ait Scree Plot Grafiği

Öz-yeterlik algısı ölçeğine ait scree plot grafiği incelendiğinde 16 maddelik ölçeğin öz değeri 1'in üzerinde olan tek faktörden oluşmaktadır.

Tablo 9. Öz-yeterlik Algısı Ölçeği Maddeleri Faktör Yük Değerleri

Faktör		Faktör	
Madde No	1	Madde No	1
Madde 1	0,917	Madde 10	0,974
Madde 2	0,963	Madde 11	0,947
Madde 3	0,983	Madde 12	0,981
Madde 4	0,984	Madde 13	0,961
Madde 5	0,945	Madde 14	0,959
Madde 6	0,987	Madde 15	0,926
Madde 7	0,975	Madde 16	0,943
Madde 8	0,966		
Madde 9	0,925		

Tablo 9’de oluşan faktör yapısı ve faktör yük değerleri sunulmaktadır. Tablodaki bulgular değerlendirilirken faktör yük değerinin $>,45$ durumları dikkate alınmıştır (Çokluk, Şekercioğlu ve Büyüköztürk, 2016). Tabloda görüldüğü üzere faktör yük değerleri ise 0,917- 0,987 arasında değişmektedir. Ölçek tek faktörlü yapıda çıktığı için binişiklik durumuna bakılmamış olup herhangi bir döndürme işlemi yapılamamıştır. Özetle, değerlendirilen ölçütler açısından sorunlu maddeye rastlanmamış faktör yüklerinin düzeyine göre madde çıkarımı yapılmasına gerek duyulmamıştır.

Ölçeğin faktör yapısı incelendiğinde ise 1. Faktör altında toplanan maddeler göz önüne alınarak faktör 1’in adına ölçeğin adı olan “*Algoritma Geliştirmeye İlişkin Öz-yeterlik Algısı*” (EK 2) adı verilmiştir.

Algoritma geliştirmeye ilişkin öz yeterlik algısı ölçeğinin, ölçtüğü özellik açısından kişileri ayırt etmede ne kadar yeterli olduğunu belirlemek amacıyla madde toplam korelasyonları ile özgün ölçekte toplam puana göre belirlenmiş Alt-Üst % 27’lik grup ortalamaları farkına dayalı madde analizi yapılmıştır (Büyüköztürk, 2011). Ölçek aracılığıyla hesaplanan puanlarla, puanlara göre belirlenen Alt-Üst % 27’lik gruplar arasında anlamlı bir fark olup olmadığını belirlemek için bağımsız örneklem t-testi yöntemine başvurulmuştur. Tablo 10’da algoritma geliştirmeye ilişkin öz-yeterlik algısı ölçeğinin Alt-Üst % 27’lik gruplar için t-testi madde analizi verilmiştir.

Tablo 10. Algoritma Geliştirmeye İlişkin Öz -yeterlik Algısı Ölçeği Alt-Üst % 27'lik Gruplar İçin t-Testi Sonuçları

Madde No	Grup	n	\bar{x}	s	T	Sd	p
Madde 1	Alt	46	2,43	0,69	-17,653	90	0,000**
	Üst	46	4,63	0,49			
Madde 2	Alt	46	2,11	0,74	-18,314	90	0,000**
	Üst	46	4,52	0,51			
Madde 3	Alt	46	1,91	0,69	-20,277	90	0,000**
	Üst	46	4,48	0,51			
Madde 4	Alt	46	1,89	0,71	-20,054	90	0,000**
	Üst	46	4,46	0,50			
Madde 5	Alt	46	1,63	0,49	-26,953	90	0,000**
	Üst	46	4,28	0,46			
Madde 6	Alt	46	1,93	0,71	-19,764	90	0,000**
	Üst	46	4,48	0,51			
Madde 7	Alt	46	1,93	0,77	-18,858	90	0,000**
	Üst	46	4,50	0,51			
Madde 8	Alt	46	1,93	0,77	-18,703	90	0,000**
	Üst	46	4,48	0,51			
Madde 9	Alt	46	2,22	0,70	-20,379	90	0,000**
	Üst	46	4,72	0,46			
Madde 10	Alt	46	1,83	0,68	-20,676	90	0,000**
	Üst	46	4,37	0,49			
Madde 11	Alt	46	1,70	0,55	-23,568	90	0,000**
	Üst	46	4,24	0,48			
Madde 12	Alt	46	1,89	0,71	-20,054	90	0,000**
	Üst	46	4,46	0,50			
Madde 13	Alt	46	1,80	0,62	-21,129	90	0,000**
	Üst	46	4,30	0,51			
Madde 14	Alt	46	1,76	0,60	-21,801	90	0,000**
	Üst	46	4,28	0,50			
Madde 15	Alt	46	1,61	0,54	-25,318	90	0,000**
	Üst	46	4,15	0,42			
Madde 16	Alt	46	2,02	0,68	-18,669	90	0,000**
	Üst	46	4,50	0,59			

**p<0.01

Tablo 10 incelendiğinde, öz yeterlik ölçeği maddelerinin Alt-Üst % 27'lik gruplar arasında puan ortalamaları açısından farkların istatistiksel açıdan anlamlı olduğu görülmüştür ($p<,01$). Bu bulgular, ölçek maddelerinin iç tutarlılığının yüksek ve bireyleri ayırt etmede yeterli olduğunu gösterir niteliktedir.

Tablo 11'de ölçme aracında yer alan aralıkların eşit olduğu varsayılarak seçeneklere ilişkin alt ve üst sınırlar ve düzeyleri belirlenmiştir.

Tablo 11. Ölçeklere İlişkin Düzey Sınırları

Seçenekler	Sınırlar	Düzeyler
Kesinlikle katılmıyorum	1,00-1,80	Çok Düşük
Katılmıyorum	1,81-2,60	Düşük
Ne Katılıyorum Ne Katılmıyorum	2,61-3,40	Orta
Katılıyorum	3,41-4,20	Yüksek
Kesinlikle katılıyorum	4,21-5,00	Çok Yüksek

3.4.3 Güdülenme ve Öğrenme Stratejileri Ölçeği

Öğrencilerin derse ilişkin güdülenme düzeylerini belirlemek için Pintrich, Smith, Garcia ve McKeachie (1993) tarafından geliştirilen; Büyüköztürk, Akgün, Özkahveci ve Demirel (2007) tarafından Türkçeye uyarlanan "Güdülenme ve Öğrenme Stratejileri Ölçeği" (EK 3) gerekli izinler alınarak kullanılmıştır. Ölçek güdülenme ve öğrenme stratejileri olmak üzere iki farklı alt ölçekten oluşmaktadır. Güdülenme bölümünde 31 madde ve altı alt boyut bulunmaktadır. Bu çalışmada ölçeğin güdülenme bölümü kullanılmıştır. 7'li Likert tipi ölçekte her bir madde "1- Benim için kesinlikle yanlış" ile "7- Benim için kesinlikle doğru" arasında değişen değerler almaktadır. Ölçek "içsel hedef düzenleme", "dışsal hedef düzenleme", "görev değeri", "öğrenmeye ilişkin kontrol inancı", "öğrenme ve performansla ilgili öz yeterlik" ve "sınav kaygısı" olmak üzere altı alt boyutu bulunmaktadır. Türkçe uyarlama çalışmasında alt faktörlerin iç tutarlılık kat sayıları 0.52 ile 0.86 arasında değişen değerler almıştır (Büyüköztürk ve diğerleri, 2007). Bu çalışmada ölçeğin alt boyutlarında inceleme yapılmamıştır.

3.4.4 Başarı Testi

Öğrencilerin programlama ile ilgili akademik başarılarını ölçmek için araştırmacı tarafından başarı testi geliştirilmiştir (EK 4). Bu başarı testinde temel programlama kavramlarını kapsayan (Algoritma, değişken, koşul ve kontrol, döngü) 25 çoktan seçmeli soru yer almaktadır. Başarı testindeki sorular, Milli eğitim Bakanlığı'nın 2012 yılında yayınladığı, çalışmanın yapıldığı dönemde de geçerli olan bilişim teknolojileri ve yazılım dersi öğretim programındaki kazanımlara uygun olarak yazılmıştır.

Başarı testinin oluşturulması için bilme, kavrama, uygulama ve değerlendirme düzeyindeki sorulardan oluşan bir madde havuzu oluşturulmuştur. Testin kapsam ve görünüş geçerliliği için BÖTE alanından beş uzmanın görüşlerine başvurulmuştur. Uzman görüşü doğrultusunda madde havuzunda yer alan bazı sorular testten çıkarılmış, yeniden düzenlenmiş, seçeneklerde ya da madde köklerinde değişiklikler yapılmıştır. Başarı testinin ilk taslak formunda 27 soru yer almıştır. Bu 27 soruluk ilk taslak form pilot uygulama yapılmadan önce, Sakarya ili, merkez ilçedeki bir ortaokulda eğitim gören 10 öğrenciye uygulanmıştır. Öğrencilerden test sırasında anlaşılmayan soruları araştırmacıya belirtmeleri istenmiştir. Test yaklaşık 40 dakika sürmüştür. Testi ilk bitiren öğrenci 30 dakikada son bitiren öğrenci ise 40 dakikada bitirmiştir. Öğrencilerin testi bitirme süreleri göz önüne alındığında pilot uygulamada test için 40 dakika süre öngörülmüştür. Pilot uygulamada ayrıca öğrencilerden gelen dönütlere göre anlaşılmayan sorularda yeniden düzenleme yapılmıştır.

Pilot uygulama sonrası düzenlenerek son hali verilmiş olan 27 soruluk test formu güvenilirlik çalışması için Sakarya ili, merkez ilçedeki bir ortaokulda eğitim gören 171 öğrenciye uygulanmıştır. Uygulama sonrası madde analizi yapılarak her bir maddenin ayırt edicilik (r) ve güçlük indisleri (p) hesaplanmış ve elde edilen değerler Tablo 12'de başarı testinin madde analizi görülmektedir.

Tablo 12. Başarı Testine İlişkin Madde Analizi Sonuçları

Madde No	Madde Ayırt edicilik indisi (r)	Ayırt Edicilik	Madde Güçlük İndisi (p)	Güçlük	Durum
1	.48	Çok iyi	.40	Orta	-
2	.29	Düzenlenebilir	.69	Orta	Yeniden düzenleme
3	.42	Çok iyi	.34	Zor	-
4	.52	Çok iyi	.35	Zor	-
5	.48	Çok iyi	.73	Kolay	-
6	.48	Çok iyi	.40	Orta	-
7	.29	Düzenlenebilir	.79	Kolay	Yeniden düzenleme
8	.48	Çok iyi	.73	Kolay	-
9	.42	Çok iyi	.63	Orta	-
10	.48	Çok iyi	.73	Kolay	-
11	.48	Çok iyi	.73	Kolay	-
12	.55	Çok iyi	.47	Orta	-
13	.52	Çok iyi	.65	Orta	-
14	.48	Çok iyi	.73	Kolay	-
15	.74	Çok iyi	.56	Orta	-
16	.65	Çok iyi	.55	Orta	-
17	.21	Düzenlenebilir	.23	Zor	Yeniden düzenleme
18	.32	İyi	.39	Zor	-
19	.48	Çok iyi	.56	Orta	-
20	.71	Çok iyi	.48	Orta	-
21	.58	Çok iyi	.32	Zor	-
22	.42	Çok iyi	.53	Zor	-
23	.32	İyi	.39	Zor	-
24					
25	.19	Testten Atılmalı	.16	Zor	Testten çıkartıldı
26	.01	Testten Atılmalı	.26	Zor	Testten çıkartıldı
27	.74	Çok iyi	.56	Orta	-

Madde güçlük indisi, her bir maddenin doğru yanıtlanma oranını göstermektedir ve “0” ile “1” arasında değerler almaktadır. Bulunan değer sıfıra yaklaştıkça maddenin zor olduğu, 1’e yaklaştıkça maddenin kolay olduğu söylenebilir. Testin ortalama madde güçlük indisininin 0,50 civarında olması ise arzu edilen bir durumdur (Çepni ve diğerleri, 2008). Madde ayırt edicilik indisi ise, testin ölçmeyi amaçladığı özelliğe yüksek düzeyde sahip bireylerle, düşük düzeyde sahip bireyleri ayırt etme derecesidir. Madde ayırt edicilik indisi “-1” ile “+1” arasında değerler

alabilmektedir. Madde ayırt edicilik indisinin sıfıra yaklaşması, maddenin ayırt ediciliğinin düşük, +1'a yaklaşması ayırt ediciliğinin yüksek olması demektir. Madde ayırt edicilik indisinin negatif çıkması ise bireyleri ters ayırt ettiğini ve testten çıkartılması gerektiğini göstermektedir (Özçelik, 2010). Madde güçlük indisi düşük çıkan ve madde ayırt edicilik indisi negatif olan test maddeleri testten çıkarılmaktadır. Analiz sonucunda iki soru maddesinin ayırt edicilik indisi .20'nin altında olduğu için testten çıkartılmıştır. Madde ayırt edicilik indisi .20 ve .30 arasında olan üç madde uzman görüşü doğrultusunda yeniden düzenlenmiştir. Ayırt edicilik indisi düşük maddeler testten çıkartıldıktan sonra 25 maddelik testin ortalama ayırt edicilik indisi $r=.42$ olarak bulunmuştur.

Testte yer alan maddelerin güçlük indisi incelendiğinde 25 maddenin güçlük indisi .23 ile .79 arasında değişmektedir. Ortalama güçlük indisi ise $p=.51$ olarak hesaplanmıştır. Bu durumda testin orta güçlükte bir test olduğu söylenebilir. Bununla birlikte testin güvenilirliğini ölçmek için ise KR20 iç tutarlılık kat sayısı hesaplanmıştır. KR-20 testi elde edilen test puanları arasındaki iç tutarlılığı ölçen ve özellikle başarı testlerinde kullanılan güvenilirlik testidir. Geliştirilen başarı testinin KR20 güvenilirlik katsayısı .710 olarak hesaplanmıştır. Bu durumda testin güvenilir ölçüm yaptığı söylenebilir (Büyüköztürk, 2009).

Kapsam geçerliliğini sağlamak için her bir kazanım en az birer soru ile sorgulanmıştır. Belirtke tablosunda bulunan kazanımlar ve soru dağılımları (25 çoktan seçmeli soru) Tablo 13'de gösterilmiştir.

Tablo 13. Belirtke Tablosunda Bulunan Kazanımlar ve Soru Dağılımları

Kazanımlar	Sorular
Algoritma ve programlama, akış şeması, sabit ve değişkenin tanımını bilir.	1, 2, 3, 4, 6
Günlük hayattaki problemlerin çözümüne yönelik algoritma oluşturur.	7, 8, 9,11
Sabit ve değişken yapıları kullanarak basit matematik işlemlerine yönelik algoritma oluşturur.	10, 13, 19, 23
“Eğer ise” koşullu yapısını kullanarak algoritma oluşturabilir.	5, 11
Farklı programların algoritmalarını inceleyerek en hızlı ve doğru çözümü seçer.	25
Döngü kavramını açıklar, sayaç kullanarak algoritma oluşturabilir.	18,25
Algoritmanın farklı değişken değerlerine göre çalışmasını inceler.	12, 14, 15, 16
Verilen algoritmanın amacını açıklar.	21, 22
Algoritmanın düzgün çalışıp çalışmadığını inceler, hataları düzeltir.	17, 20, 24,

Programlama başarı testinin son şeklinin belirtke tablosu Tablo 14’te verilmiştir.

Tablo 14. Programlama Başarı Testinin Belirtke Tablosu

Öğrenme Alanı	Bilgi	Kavrama	Uygulama	Analiz	Sentez	Değerlendirme
Algoritma Kavramları	1, 2, 3, 4,5	6	7,8,9			
Akış Şeması,			21,24			
Sabit, Değişkenler		10,12,13	16,18			
Koşullu İfadeler		11	14,17			20,23
Döngü Yapıları			15			22,25

3.5 VERİLERİN ANALİZİ

Araştırmada elde edilen veriler SPSS 25.0 ile analiz edilmiştir. Araştırmaya katılanların demografik özelliklerine ait dağılımları belirlemek için frekans ve yüzde analizinin yanı sıra ölçeklere ait düzeylerin belirlenmesi için betimsel analizlerden ortalama ve standart sapma kullanılmıştır. Araştırmada kullanılan ölçeklere ait geçerlilik güvenirlik analizlerinde önce madde analizi, ardından madde toplam korelasyon analizi yapılmış olup daha sonra ölçeklerin geçerliliklerinin belirlenmesi için açıklayıcı (AFA) faktör analizleri yapılarak maddelerin ayırt edicilik düzeylerini belirlemek için %27 alt-üst madde ayırt edicilik analizi yapılmıştır. Elde edilen ölçeklerin güvenirlik düzeylerini belirlemek için Cronbach's Alpha iç tutarlılık analizi yapılmıştır. Araştırma sonucunda elde edilen verilerin dağılımlarını belirlemek için normal dağılım analizlerinden Shapiro-Wilk analizi yapılarak dağılımın normal olmaması sonucu 2'li gruplar arasındaki farkın anlamlılığına bakarken Mann Whitney U testi, ön test-son test düzeylerine göre farklılıklar bakılırken Wilcoxon Sıralı İşaretler analizleri yapılmıştır.

BÖLÜM IV

BULGULAR

Araştırmanın bu bölümünde geliştirilen öz-yeterlik algısı ölçeği ve araştırmada kullanılan güdüleme ölçeği ile başarı testlerinin dağılımı, betimsel analizlerinin yanı sıra araştırmanın problemlerine yönelik geliştirilen hipotezlere ait analiz sonuçlarına yer verilmiştir.

4.1 ARAŞTIRMADA KULLANILAN ÖLÇEKLERE İLİŞKİN BETİMSSEL BULGULAR

Tablo 15’te, araştırmada kullanılan ölçeklere ilişkin betimsel bulgular görülmektedir.

Tablo 15. Araştırmada kullanılan Ölçeklere İlişkin Betimsel Bulgular

Ölçekler	\bar{X}	s.s
Öz-yeterlik	2,70	1,75
Güdüleme	5,52	0,70
Başarı Testi	44,31	37,0

Araştırmada kullanılan ölçeklere ait betimsel istatistikler incelendiğinde, öz-yeterlik algısı düzeylerinin orta seviyede ($\bar{X}=2,70$), güdüleme düzeylerinin orta düzeyde olduğu ($\bar{X}=5,52$) ve araştırmada kullanılan başarı testi düzeylerinin de orta seviyede olduğu ($\bar{X}=44,31$) belirlenmiştir.

Tablo 16’da araştırmada kullanılan ölçeklerde verilerin dağılımına ait normallik testi sonuçları görülmektedir.

Tablo 16. Verilerin Dağılımına Ait Normallik Testi Sonuçları

Grup	Ölçek	Statistic	p	\bar{x}	Medyan	Çarpıklık	Basıklık
Deney	Başarı Testi	0,797	0,000	46,13	40,00	0,27	-3,18
	Öz Yeterlilik	0,674	0,000	2,89	2,66	0,05	-3,47
	Güdüleme	0,911	0,000	5,62	5,82	-1,71	-2,76
Kontrol	Başarı Testi	0,829	0,000	42,50	36,00	0,49	-2,98
	Öz Yeterlilik	0,767	0,000	2,52	2,09	0,66	-3,02
	Güdüleme	0,923	0,001	5,42	5,54	-0,85	-2,24

Verilerin hangi dağılımdan geldiğini belirlemek için; Saphiro-Wilk normal dağılım analizi yapılmış olup bu analize göre verilerin dağılımının normallikten gelmediği ($p < 0.05$) belirlenmiştir. Verilerin dağılımına daha net karar verebilmek için sadece normal dağılım analizindeki p değerine değil aritmetik ortalama, medyan, çarpıklık ve basıklık katsayıları incelenmesi gerekliliği ve bu değerlerden, aritmetik ortalama ve medyanın eşit ya da yakın olması, çarpıklık ve basıklık katsayılarının ± 2 sınırları içinde bulunmasından verilerin dağılımının normallikten geldiğine işaret (Tabachnick ve Fidell, 2014) olduğunu göstermektedir. Ölçeklerin ortalama ve medyan değerlerinin yakınlığı olmadığı ve aynı zamanda basıklık ile çarpıklık değerlerinin ± 2 sınırları içerisine düşmediğinden dağılımın normal olmadığı varsayımı daha da kuvvetlenmiştir.

4.2 BAŞARI TESTİ, ÖZ-YETERLİK ALGISI VE GÜDÜLENME PUANLARININ DENEY VE KONTROL GRUPLARININ KARŞILAŞTIRILMASINA YÖNELİK BULGULARI

Araştırmanın birinci alt problemine ait bulgular:

Araştırmanın birinci problemi “Programlama öğretiminde, blok tabanlı Scratch programlama aracının kullanıldığı ortamdaki deney grubu öğrencileri ile programlama aracının kullanılmadığı ortamdaki kontrol grubu öğrencilerinin ders

başarısı, öz-yeterlik algısı ve güdülenme puanları arasında anlamlı bir fark var mıdır?” şeklinde olup yanıtın belirlenmesinde Mann Whitney U testi yapılmıştır.

Öğrencilerin başarı testi düzeylerinin deney-kontrol gruplarına ait istatistikleri Tablo 20’de verilmiştir.

Tablo 17. Öğrencilerin Başarı Testi Düzeyleri Deney-Kontrol Grubuna Göre Farklılıklarına Ait Mann Whitney U Testi Sonucu

Değişken	Grup	n	Mean Rank	Z	p
Başarı Testi	Deney	64	67,13	-0,804	0,422
	Kontrol	64	61,88		

Öğrencilerin başarı testi düzeyleri deney-kontrol gruplarına göre farklılık gösterip göstermediğini belirlemek için yapılan Mann Whitney U testi sonucunda, öğrencilerin başarı testlerinden aldıkları puanların deney-kontrol gruplarına göre farklılığı istatistiksel olarak anlamlı olmadığı belirlenmiştir ($p>0.05$).

Öğrencilerin öz-yeterlik algısı düzeylerinin deney-kontrol gruplarına göre farklılık gösterip göstermediğinin test edilmesinde Mann Whitney U testi yapılmıştır.

Öğrencilerin öz-yeterlik algısı düzeyleri deney-kontrol gruplarına ait istatistikleri Tablo 18’de verilmiştir.

Tablo 18. Öğrencilerin Öz-yeterlik Algısı Düzeylerinin Deney-Kontrol Grubuna Göre Farklılıklarına Ait Mann Whitney U Testi Sonucu

Değişken	Grup	n	Mean Rank	Z	p
Öz Yeterlik Algısı	Deney	64	70,04	-1,795	0,073
	Kontrol	64	58,96		

Öğrencilerin öz-yeterlik algısı düzeyleri deney-kontrol gruplarına göre farklılık gösterip göstermediğini belirlemek için yapılan Mann Whitney U testi sonucunda,

öğrencilerin öz yeterlik düzeylerinin deney-kontrol gruplarına göre farklılığı istatistiksel olarak anlamlı olmadığı belirlenmiştir ($p>0.05$).

Öğrencilerin güdülenme düzeylerinin deney-kontrol gruplarına göre farklılık gösterip göstermediğinin test edilmesinde Mann Whitney U testi yapılmıştır.

Öğrencilerin güdülenme düzeylerinin deney-kontrol gruplarına ait istatistikleri Tablo 19’de verilmiştir.

Tablo 19. Öğrencilerin Güdüleme Düzeyleri Deney-Kontrol Grubuna Göre Farklılıklarına Ait Mann Whitney U Testi Sonucu

Değişken	Grup	n	Mean Rank	Z	p
Güdüleme	Deney	64	70,31	-1,773	0,076
	Kontrol	64	58,69		

Öğrencilerin güdüleme düzeyleri deney-kontrol göre farklılık gösterip göstermediğini belirlemek için yapılan Mann Whitney U testi sonucunda, öğrencilerin güdüleme düzeylerinin deney-kontrol gruplarına göre farklılığı istatistiksel olarak anlamlı olmadığı belirlenmiştir ($p>0.05$).

4.4 BAŞARI TESTİ, ÖZ-YETERLİK ALGISI VE GÜDÜLENME PUANLARININ DENEY GRUBUNA YÖNELİK BULGULARI

Araştırmanın ikinci alt problemine ait bulgular:

Araştırmanın ikinci alt problemi “Blok tabanlı Scratch programlama aracının kullanıldığı öğretim ortamındaki deney grubu öğrencilerinin öntest-sontest;

- Ders başarısı puanları arasında anlamlı bir fark var mıdır?
- Öz-yeterlik algısı puanları arasında anlamlı bir fark var mıdır?
- Derse ilişkin güdülenme puanları arasında anlamlı bir fark var mıdır?

şeklinde olup yanıtların belirlenmesinde Wilcoxon Sıralı İşaretler testi yapılmıştır.

Deney grubu öğrencilerinin öğretim etkinliklerine göre öntest-sontest başarı testi puanlarının anlamlı bir fark gösterip göstermediğinin test edilmesinde yapılan Wilcoxon Sıralı İşaretler testine ait sonuçlar Tablo 20’de verilmiştir.

Tablo 20. Deney Grubu Öğrencilerin Öntest-Sontest Başarı Testi Puanlarının Farklılıklarına Ait Wilcoxon Sıralı İşaretler Testi Sonucu

	Gruplar	N	Sıra Ort.	Z	p
Son Test Başarı Testi - Ön Test Başarı Testi	Negatif Sıralar	0a	0,00	-4,958	0,000**
	Pozitif Sıralar	32b	16,50		
	Eşit	0c			

a. Son Test Başarı Testi < Ön Test Başarı Testi

b. Son Test Başarı Testi > Ön Test Başarı Testi

c. Son Test Başarı Testi = Ön Test Başarı Testi

**p<0.01

Deney grubu öğrencilerin öntest-sontest başarı düzeyleri Scratch programlama etkinliklerine göre farklılık gösterip göstermediğini belirlemek için yapılan Wilcoxon Sıralı İşaretler testi sonucunda, deney grubu öğrencilerinin öntest-sontest ders başarı düzeyleri Scratch programlama etkinliklerinden sonra istatistiksel olarak anlamlı olduğu belirlenmiştir (Z= -4,958; p= 0,000; p<0.01). Gruplar incelendiğinde, deney grubu öğrencileri için başarı test puanlarının son test başarı testlerinin ön test başarı testlerine göre pozitif anlamda yükseldiği görülmektedir (Sıra Ort.= 16,50). Bu da ders başarısı puanlarının deney grubundaki öğrenciler için Scratch programlama ortamında öğretim etkinlikleri yapıldıktan sonra daha yüksek olduğunu göstermektedir.

Deney grubu öğrencilerinin öğretim etkinliklerine göre öntest-sontest öz-yeterlik algısı puanlarının anlamlı bir fark gösterip göstermediğinin test edilmesinde yapılan Wilcoxon Sıralı İşaretler testine ait sonuçlar Tablo 21’de verilmiştir.

Tablo 21. Deney Grubu Öğrencilerin Öntest-Sontest Öz-yeterlik Algısı Puanlarının Farklılıklarına Ait Wilcoxon Sıralı İşaretler Testi Sonucu

	Gruplar	n	Sıra Ort.	Z	P
Son Test Öz yeterlik - Ön Test Öz yeterlik	Negatif Sıralar	0a	0,00	-4,947	0,000**
	Pozitif Sıralar	32b	16,50		
	Eşit	0c			

a. Son Test Öz yeterlik < Ön Test Öz yeterlik
b. Son Test Öz yeterlik > Ön Test Öz yeterlik
c. Son Test Öz yeterlik = Ön Test Öz yeterlik
**p<0.01

Deney grubu öğrencilerinin öntest-sontest öz-yeterlik algısı düzeyleri Scratch ile yapılan programlama etkinliklerine göre farklılık gösterip göstermediğini belirlemek için yapılan Wilcoxon Sıralı İşaretler testi sonucunda, deney grubu öğrencilerinin öntest-sontest öz-yeterlik algısı düzeyleri Scratch programlama öğretimi etkinliklerinden sonraki farklılığı istatistiksel olarak anlamlı olduğu belirlenmiştir (Z= -4,947; p= 0,000; p<0.01). Gruplar incelendiğinde, deney grubundaki öğrenciler için öz-yeterlik algısı düzeylerinin son test öz yeterlik düzeylerinin ön test öz yeterlik düzeylerine göre pozitif anlamda yükseldiği görülmektedir (Sıra Ort.= 16,50). Bu da öz-yeterlik algısı düzeylerinin, deney grubu öğrencileri için Scratch programlama ortamında yapılan etkinliklerden sonra daha yüksek olduğunu göstermektedir.

Deney grubu öğrencilerinin öğretim etkinliklerine göre öntest-sontest güdülenme puanlarının anlamlı bir fark gösterip göstermediğinin test edilmesinde yapılan Wilcoxon Sıralı İşaretler testine ait sonuçlar Tablo 22’de verilmiştir.

Tablo 22. Deney Grubu Öğrencilerinin Öntest-Sontest GÜdüleme Puanlarının Farklılıklarına Ait Wilcoxon Sıralı İşaretler Testi Sonucu

	Gruplar	n	Sıra Ort.	Z	p
Son Test Güdüleme - Ön Test Güdüleme	Negatif Sıralar	5a	6,80	-3,850	0,000**
	Pozitif Sıralar	23b	16,17		
	Eşit	4c			

a. Son Test Güdüleme < Ön Test Güdüleme
b. Son Test Güdüleme > Ön Test Güdüleme
c. Son Test Güdüleme = Ön Test Güdüleme
**p<0.01

Deney grubu öğrencilerinin öntest-sontest güdüleme düzeyleri Scratch programlama ortamında öğretim etkinliklerine göre farklılık gösterip göstermediğini belirlemek için yapılan Wilcoxon Sıralı İşaretler testi sonucunda, deney grubu öğrencilerinin öntest-sontest güdülenme düzeylerinin Scratch öğretim etkinliklerinin yapılmasına ait farklılığı istatistiksel olarak anlamlı olduğu belirlenmiştir ($Z = -3,850$; $p = 0,000$; $p < 0,01$). Gruplar incelendiğinde, deney grubundaki öğrenciler için güdüleme düzeylerinin son test güdüleme düzeylerinin ön test güdüleme düzeylerine göre pozitif anlamda yükseldiği görülmektedir (Sıra Ort= 16,17). Bu da güdüleme düzeylerinin deney grubu öğrenciler için Scratch programlama ortamında yapılan etkinliklerden sonra daha yüksek olduğunu göstermektedir.

4.5. BAŞARI TESTİ, ÖZ-YETERLİK ALGISI VE GÜDÜLENME PUANLARININ KONTROL GRUBUNA YÖNELİK BULGULARI

Araştırmanın üçüncü alt problemine ait bulgular:

Araştırmanın üçüncü alt problemi “Programlama aracının kullanılmadığı geleneksel öğretim ortamındaki kontrol grubu öğrencilerinin öntest-sontest,

- Ders başarısı puanları arasında anlamlı bir fark var mıdır?
- Öz-yeterlik algısı puanları arasında anlamlı bir fark var mıdır?

c. Derse ilişkin güdülenme puanları arasında anlamlı bir fark var mıdır?

şeklinde olup yanıtların belirlenmesinde Wilcoxon Sıralı İşaretler testi yapılmıştır.

Kontrol grubu öğrencilerinin öğretim etkinliklerine göre öntest-sontest başarı testi puanlarının anlamlı bir fark gösterip göstermediğinin test edilmesinde yapılan Wilcoxon Sıralı İşaretler testine ait sonuçlar Tablo 23’da verilmiştir.

Tablo 23. Kontrol Grubu Öğrencilerinin Öntest-Sontest Ders Başarısı Puanlarının Farklılıklarına Ait Wilcoxon Sıralı İşaretler Testi Sonucu

	Gruplar	n	Sıra Ort.	Z	p
Son Test Başarı Testi - Ön Test Başarı Testi	Negatif Sıralar	0a	0,00	-4,944	0,000**
	Pozitif Sıralar	32b	16,50		
	Eşit	0c			

a. Son Test Başarı Testi < Ön Test Başarı Testi

b. Son Test Başarı Testi > Ön Test Başarı Testi

c. Son Test Başarı Testi = Ön Test Başarı Testi

**p<0.01

Kontrol grubu öğrencilerin öntest-sonteste göre ders başarı düzeylerini yani geleneksel programlama etkinliklerinden sonra farklılık gösterip göstermediğini belirlemek için yapılan Wilcoxon Sıralı İşaretler testi sonucunda, kontrol grubu öğrencilerinin öntest-sontest ders başarı düzeyleri öğretim etkinliklerine ait farklılığı istatistiksel olarak anlamlı olduğu belirlenmiştir (Z= -4,944; p= 0,000; p<0.01). Gruplar incelendiğinde, kontrol grubu öğrencileri için başarı test puanlarının son test başarı testlerinin ön test başarı testlerine göre pozitif anlamda yükseldiği görülmektedir (Sıra Ort = 16,50). Buda ders başarısı puanlarının kontrol grubundaki öğrenciler için geleneksel programlama öğretiminden sonraki başarı düzeylerinin öğretim etkinlikleri yapılmadan önceki puanlara göre daha yüksek olduğunu göstermektedir.

Kontrol grubu öğrencilerinin öğretim etkinliklerine göre öntest-sontest öz-yeterlik algısı puanlarının anlamlı bir fark gösterip göstermediğinin test edilmesinde yapılan Wilcoxon Sıralı İşaretler testine ait sonuçlar Tablo 24’de verilmiştir.

Tablo 24. Kontrol Grubu Öğrencilerin Öntest-Sontest Öz-yeterlik Algısı Puanlarının Farklılıklarına Ait Wilcoxon Sıralı İşaretler Testi Sonucu

	Gruplar	n	Sıra Ort.	Z	p
Son Test Öz yeterlik - Ön Test Öz yeterlik	Negatif Sıralar	0a	0,00	-4,937	0,000**
	Pozitif Sıralar	32b	16,50		
	Eşit	0c			

a. Son Test Öz yeterlik < Ön Test Öz yeterlik
b. Son Test Öz yeterlik > Ön Test Öz yeterlik
c. Son Test Öz yeterlik = Ön Test Öz yeterlik
**p<0.01

Kontrol grubu öğrencilerinin öntest-sontest öz-yeterlik algısı düzeylerinin geleneksel programlama öğretim etkinliklerine göre farklılık gösterip göstermediğini belirlemek amacıyla yapılan Wilcoxon Sıralı İşaretler testi sonucunda, kontrol grubu öğrencilerinin öntest-sontest öz yeterlik düzeyleri öğretim etkinliklerinden sonraki farklılığı istatistiksel olarak anlamlı olduğu belirlenmiştir (Z= -4,937; p= 0,000; p<0.01). Gruplar incelendiğinde, kontrol grubundaki öğrenciler için öz yeterlik algısı düzeylerinin son test öz yeterlik düzeylerinin ön test öz yeterlik düzeylerine göre pozitif anlamda yükseldiği görülmektedir (Sıra Ort= 16,50). Bu da öz-yeterlik algısı düzeylerinin kontrol grubu öğrenciler için geleneksel öğretim etkinlikleri yapıldıktan sonra daha yüksek olduğunu göstermektedir.

Kontrol grubu öğrencilerinin öğretim etkinliklerine göre öntest-sontest güdülenme puanlarının anlamlı bir fark gösterip göstermediğinin test edilmesinde yapılan Wilcoxon Sıralı İşaretler testine ait sonuçlar Tablo 25’de verilmiştir.

Tablo 25. Kontrol Grubu Öğrencilerinin Öntest-Sontest GÜdüleme Puanlarının Farklılıklarına Ait Wilcoxon Sıralı İşaretler Testi Sonucu

	Gruplar	n	Sıra Ort	Z	P
Son Test Güdüleme - Ön Test Güdüleme	Negatif Sıralar	30a	17,05	-4,631	0,000**
	Pozitif Sıralar	2b	8,25		
	Eşit	0c			

a. Son Test Güdüleme < Ön Test Güdüleme
b. Son Test Güdüleme > Ön Test Güdüleme
c. Son Test Güdüleme = Ön Test Güdüleme
**p<0.01

Kontrol grubu öğrencilerinin öntest-sontest güdüleme düzeyleri geleneksel öğretim etkinliklerine göre farklılık gösterip göstermediğini belirlemek için yapılan Wilcoxon Sıralı İşaretler testi sonucunda, kontrol grubu öğrencilerinin öntest-sontest güdülenme düzeylerinin geleneksel programlama öğretimi etkinliklerinden sonraki farklılığın istatistiksel olarak anlamlı olduğu belirlenmiştir (Z= -4,631; p= 0,000; p<0.01). Gruplar incelendiğinde, kontrol grubundaki öğrenciler için güdüleme düzeylerinin son test güdüleme düzeylerinin ön test güdüleme düzeylerine göre negatif anlamda yükseldiği görülmektedir (Sıra Ort= 17,05). Bu da güdüleme düzeylerinin kontrol grubu öğrenciler için akış şemalarıyla yapılan geleneksel programlama etkinliklerinden sonra daha düşük olduğunu göstermektedir. Yani kontrol grubu öğrencilerinin geleneksel programlama etkinliklerinden sonra güdülenme düzeylerinin düştüğü görülmektedir.

BÖLÜM V

SONUÇ, TARTIŞMA VE ÖNERİLER

Bu bölümde, araştırmanın amacı, yöntem ve bulguları özetlenerek sonuçlara dayalı önerilere yer verilmiştir. Araştırma konusu ile ilgili yapılan tez, makale, bildiri gibi akademik yayınlardan elde edilen sonuçların benzerlik gösterip göstermediği açıklanmaktadır.

5.1 SONUÇ VE TARTIŞMA

Dünya çapında programlama eğitiminin öneminin artmasıyla birlikte bilişim teknolojileri dersi öğretim programına yönelik düzenlemeler yapılmıştır. 2006-2012 yılları arasında seçmeli bilgisayar dersi olarak verilen ve içinde üç programlama kazanımı olan dersin adı “Bilişim Teknolojileri ve Yazılım” dersi olarak değiştirilmiş ve yeni öğretim programında programlama konusunun kazanımları artırılmıştır (TTKB, 2012). Ortaöğretim müfredatlarında yapılan çalışmalar sonucunda yenilenen “Bilgisayar Bilimi Dersi Öğretim Programı”nda programlama konusuna ve programlama dillerinin öğretimine yer verilmektedir (TTKB, 2016).

Milli Eğitim Bakanlığı farklı ülkelerin güncellenen öğretim programlarını, yapılan akademik çalışmaları, illerden gelen zümre raporları gibi çalışmaları inceleyerek 2017 yılında ilköğretim bilişim teknolojileri ve yazılım dersi öğretim programını güncellemiş ve yeniden yayınlanmıştır. Yeni öğretim programında ilkokul 1. Sınıftan 4. Sınıfa kadar olan dönemde dersin serbest etkinlik saatlerinde verilip verilmemesi sınıf öğretmenin tercihine bırakılmıştır.

5. ve 6. sınıflarda haftada 2 saat zorunlu, 7. ve 8. sınıflarda ise haftada 2 saat seçmeli olarak yürütülmesi planlanan dersin toplam süresinin yarısının “problem çözme ve programlama” ünitesine ayrıldığı görülmektedir (TTKB, 2017).

Milli Eğitim Bakanlığı'nın 2017 yılında yayınladığı öğretim programında bilişim teknolojileri ve yazılım dersinde “Problem Çözme ve Programlama” ünitesinin dersin tüm kazanımların yaklaşık yarısına sahip olması bu konunun nasıl daha verimli işlenmesi gerektiğini düşündürmektedir. Ancak bu konuda yapılan çalışmaların ve örnek uygulamaların sınırlı sayıda olduğu görülmektedir.

Bu araştırmada; öğrencilere programlama eğitimi verilerek öğrencilerin eleştirel, mantıksal ve algoritmik düşünme, problem çözme becerisi gibi 21. yüzyıl becerilerinin geliştirildiğine; programlamanın da bir 21. yüzyıl becerisi olarak görüldüğüne ilişkin çalışmalara yer verilmiştir. Ayrıca 21. yüzyıl becerileri ve programlama mantığının erken yaşlardan itibaren öğrencilere kazandırılmasının önemine değinilmiş; bu konuda yapılan araştırmalar, hazırlanan raporlar sunulmuştur. Bilim insanları programlama eğitiminin başlangıcında; programlamanın temeli ve mantığı olarak temel kavramları öğretmenin ve algoritmik düşünmeyi geliştirmenin önemli olduğunu vurgulamaktadırlar.

Çalışma kapsamında; Scratch programlama aracının öğrencilerin programlama başarısı, algoritma geliştirmeye ilişkin öz-yeterlik algısı ve güdülenme düzeylerine etkisi araştırılmıştır. Öğrencilerin başarı puanlarını ve öz-yeterlik algısı düzeylerini belirleyebilmek için araştırmacı tarafından programlama başarı testi ve algoritma geliştirmeye ilişkin öz-yeterlik algısı ölçeği geliştirilmiştir. Programlama performansının bir göstergesi olarak “ Algoritma geliştirmeye ilişkin öz-yeterlik algısı ölçeği” bu alanda yapılacak çalışmalarda kullanılabilir.

Elde edilen verilere göre sonuçlar aşağıdadır.

5.1.1 Deney ve Kontrol Gruplarının Başarı Puanı, Öz-yeterlik Algısı ve Güdülenme Düzeylerinin Karşılaştırılmasına Yönelik Ortaya Çıkan Sonuçlar

- Yapılan programlama öğretim etkinliklerinden sonra deney ve kontrol grubu arasında başarı puanı açısından anlamlı farklılık bulunamamıştır.

Programlama öğretiminin yapıldığı her iki grupta da başarı puanı yükselmekte fakat grupların başarı puanları arasında anlamlı farklılık görülmemektedir.

Benzer bir arařtırmada ıkan sonular arařtırmanın bu bulgusunu desteklemektedir. Koorsse ve diđerleri (2015) yaptıkları alıřmada Delphi programlama dilinin ğretildiđi u deney grubunda Delphi' ye ilaveten; Scratch, B# ve Robomind programlama ortamlarından biri ğretim materyali olarak kullanılmıřtır. Kontrol grubu ise yalnızca Delphi Programlama dili ile ğretime devam etmiřtir. ğretimin sonunda Scratch, B# ve Robomind programlama ortamlarının đrenci bařarısını artırdıđını ancak kontrol grubu ile arasında anlamlı bir fark olamadıđı grlmřtr. Ancak temel programlama yapılarından olan dng kavramını deney gruplarının daha iyi kavradıđı grlmřtr.

Erol'un (2015) yaptıđı arařtırmada ise farklı sonular grlmektedir: Scratch ile oyun tasarımı yapan đrenci grubunun geleneksel yntemlerle programlama đrenen gruba gre programlama bařarılarının daha yksek olduđunu grlmřtr.

- Yapılan programlama ğretim etkinliklerinden sonra deney ve kontrol grubu arasında z-yeterlik algısı puanları aısından anlamlı farklılık bulunamamıřtır.
- Yapılan programlama ğretim etkinliklerinden sonra deney ve kontrol grubu arasında gdlenme puanları aısından anlamlı farklılık bulunamamıřtır.

Benzer bir alıřmada benzer sonular elde edilmiřtir. elik ve zdener (2019) bilgisayarlı (Scratch ortamı) ve bilgisayarsız (alıřma kađıtları, akıř řemaları) programlama etkinliklerinin đrencilerin gdlenme dzeyine etkisini arařtırdıklarında farklı bulgulara ulařmıřlardır. Hem deney hem de kontrol grubunun đrenme etkinleri srecinin sonunda gdlenme dzeylerinin arttıđı ancak gruplar arasında gdlenme puanları aısından anlamlı bir farklılık bulunmadıđı grlmřtr. Geleneksel yntemin yeniliki Scratch ortamı kadar gdleyici olmasının sebebi olarak; iyi tasarlanmış ders etkinliklerinin đrencinin ilgisini ekmiř olması gsterilebilir.

5.1.2 Scratch Programlama Aracı İle đrenen đrencilerin Bařarı Puanı, z-yeterlik Algısı ve Gdlenme Dzeylerine Gre Ortaya ıkan Sonular

- Scratch ortamında programlama đrenen deney grubu đrencilerinin bařarı puanları n test ve son testte aldıkları puanlara gre karřılařtırıldıđında son

test puanlarının ön test puanlarından anlamlı derecede yüksek olduğu sonucuna ulaşılmıştır.

- Scratch ortamında programlama öğrenen deney grubu öğrencilerinin öz yeterlik algısı düzeylerinin öğretim etkinliklerinden sonra anlamlı derecede yükseldiği sonucuna ulaşılmıştır.

Benzer bir çalışmada farklı sonuçlar elde edilmiştir; Chiu (2014) Scratch ile programlama öğretim etkinliklerinin öğretmenlerinin programlama öz yeterliklerine etkisini araştırdığı çalışmada; programlama bilgisi ve deneyimi olmayan öğretmenler Scratch ile oyun tasarlama etkinlikleri yaptıkları çalışmada programlamaya ilişkin öz- yeterlik öntest ve sontest puanları arasında bir farklılık bulunmamıştır. Ancak öğretmenler, Scratch programlama ortamının ilgi çekici olduğunu, programın sahip olduğu özelliklerin öğrenmeyi kolaylaştırdığını, Scratch programına yönelik pozitif tutuma sahip olduklarını ve bu programın programlama öğretimi için uygun bir araç olduğunu ifade etmişlerdir.

- Scratch ortamında programlama öğrenen deney grubu öğrencilerinin güdülenme düzeylerinin öğretim etkinliklerinden sonra anlamlı derecede yüksek olduğu sonucuna ulaşılmıştır.

Rizvi ve diğerlerinin (2011) yaptıkları çalışmada da benzer sonuçlar elde edilmiştir. Üniversite öğrencilerinin CS0 dersinde, programlamanın temeli olan algoritma konusu Scratch ile oyun, animasyon, hikâye tasarım etkinlikleri ile öğretilmiştir. Etkinlikler sonucunda Scratch programının öğrencilerin programlamaya ilişkin öz yeterlik algılarının ve tutumunu anlamlı şekilde artırdığı görülmüştür. Kafai'ye (2006) göre programlama sürecinde aktif olan öğrencinin motivasyonu yükselir ve öğrenmesi kolaylaşır.

5.1.3 Programlama Aracı Kullanılmayan Ortamda Öğrenen Öğrencilerin Başarı Puanı, Öz-yeterlik Algısı ve Güdülenme Düzeylerine Göre Ortaya Çıkan Sonuçlar

- Geleneksel öğrenme ortamında programlama öğrenen kontrol grubu öğrencilerinin başarı puanlarının öğrenme etkinliklerinden sonra arttığı görülmüştür.

- Geleneksel öğrenme ortamında programlama öğrenen kontrol grubu öğrencilerinin öz-yeterlik algısı puanlarının öğrenme etkinliklerinden sonra arttığı görülmüştür.
- Geleneksel öğrenme ortamında programlama öğrenen kontrol grubu öğrencilerinin güdülenme düzeyi puanlarının öğrenme etkinliklerinden sonra azaldığı görülmüştür. Yani yapılan öğrenme etkinliklerinin kontrol grubu öğrencilerinin güdülenme düzeyini düşürdüğü görülmektedir. 32 kontrol grubu öğrencisinden 30 kişinin güdülenme düzeyinin öğrenme etkinliklerinden sonra düştüğü, 2 öğrencinin ise öğrenme etkinliklerinden sonra güdülenme düzeyinin yükseldiği görülmüştür.

Benzer bir çalışmada ortaya çıkan sonuçlar araştırmanın bu bulgularını desteklemektedir. Erol'un (2015) yaptığı araştırma sonunda; öğrencilerin motivasyon puanları ön testte her iki grupta da benzer, son testlerde ise deney grubu lehine anlamlı farklılık bulunmuştur. Geleneksel yöntemlerle programlama öğrenen kontrol grubundaki katılımcıların motivasyon puanlarının tüm uygulama sonunda azaldığı, Scratch programlama ortamında programlama öğrenen deney grubu öğrencilerinin motivasyon puanlarının arttığı görülmüştür.

Ortaokulda algoritma öğretimine yönelik olarak yapılan benzer bir çalışmada başarı değişkeni açısından paralel sonuçlar görülürken cinsiyet değişkeni açısından farklı sonuçlar elde edilmiştir. Tekerek ve Altan'ın (2014) yaptıkları çalışmada ortaokul 6. sınıfta Scratch'in algoritma (temel programlama) öğretimine etkisini incelenmiştir. Araştırmanın deney grubunda algoritma öğretimi, Scratch programlama ortamı ile kontrol grubunda ise düz anlatım yöntemiyle gerçekleştirilmiştir. Çalışma sonunda her iki grupta da öğrenci başarısının arttığı görülmüştür ancak son test puanları karşılaştırıldığında deney ve kontrol grubu arasında bir farklılık bulunmamıştır.

Tüm bu bulgular incelendiğinde hem Scratch ile yapılan programlama öğretimi etkinliklerinin hem de programlama aracı kullanılmayan ortamda geleneksel yöntemlerle yapılan programlama öğretim etkinliklerinin başarı ve öz-yeterlik algısını artırmada etkili olduğu yani programlama mantığını kazandırmada etkili olduğu ve öğrencilerin algoritma geliştirmeye ilişkin öz-yeterlik algısı düzeyini artırdığı görülmektedir. Programlama deneyimine sahip olmayan öğrencilerin programlama kavramları ile daha önce karşılaşmamış olmaları öz-yeterlik algılarının başlangıçta düşük olmasına sebep olabilir. Bu bağlamda; daha önce programlama ve

algoritma kavramlarına ait hiçbir fikri olmayan öğrencilerin programlama eğitimi aldıktan sonra öz-yeterlik algılarının yükselmesi beklenebilir.

Scratch'in kullanıldığı ya da kullanılmadığı öğretim ortamlarında iyi tasarlanmış programlama etkinliklerinin 21. yüzyıl becerilerinden biri olan “algoritmik düşünme” becerisini geliştirdiği söylenebilir.

Ancak Scratch ile öğrenen öğrencilerin güdülenme düzeylerinin etkinlikler sonunda yükseldiği, programlama aracı kullanılmayan ortamda öğrenen öğrencilerin güdülenme düzeylerinin etkinlikler sonunda azaldığı görülmektedir. Programlama sürecine başlamadan önce deney grubuna Scratch programının tanıtılması, ilgi çekici örnek projelerin gösterilmesi öğrencilerin öğrenme etkinliklerine başlarken konuya güdülenmiş bir şekilde başlamalarına yardımcı olmuş olabilir. Buradan yola çıkarak; uygulama sürecine deney gurubu öğrencilerinin kendi tasarımlarını yapabileceklerine ilişkin algılarının yüksek olması ve güdülenmiş olarak başlaması öğrencilerinin programlama mantığını da daha kolay kavramalarına yardımcı olmuş olabilir.

Kontrol grubunda yapılan etkinliklerin güdülenmeyi azaltmasının sebebi olarak; deneysel süreçler başlamadan önce bilişim teknolojileri ve yazılım dersinde bilgisayarlı teknolojiler kullanılırken; deneysel süreçte geleneksel anlatım yöntemi ve akış şemaları ile yapılan öğretimde algoritma adımlarının kâğıt üzerinde yazılması, renksiz, şekil ve sembollerin kullanılması ve bunların ilgi çekici olmaması gösterilebilir.

Scratch programlama aracı ile öğrenim gören öğrencilerin kendi tasarımlarını yapabilmesi öz-yeterlik algısını yükseltirken; yaptıkları tasarımın görsel ve işitsel olarak canlandırılması güdülenmeleri artırmış olabilir. Ayrıca deney grubunda her ders yeni bir etkinliğin olması, öğrencilerin derse meraklı ve heyecanlı gelmelerini sağlamış olabilir. Bunlara ilaveten; Scratch programlama aracında tasarım yapan öğrencilerin projelerini Scratch'in web sayfasında yayınlayabilmesi, akranlarından olumlu dönütler alması güdülenme düzeylerinin yükselmesini sağlamış olabilir.

5.2 ÖNERİLER

5.2.1 Uygulayıcılara Yönelik Öneriler

1. Programlama öğretiminde, öğrenenlerin başarı, güdülenme ve öz-yeterlik algılarını artırmak için öğrenen merkezli öğrenme ortamları tasarlanabilir.
2. Programlama öğretiminde, öğrenenlerin başarı, güdülenme ve öz-yeterlik algılarını artırmak için öğrenenlerin birlikte öğrendiği ve öğrenenleri araştırmaya ve ürünlerini sergilemeye teşvik eden öğrenme ortamları tasarlanabilir.
3. Öğretmenlerin programlama etkinliklerini planlarken öğrenenlerin konuya yönelik öz-yeterlik algılarını göz önünde bulundurması yapılan etkinliklerin daha verimli olmasını sağlayabilir.
4. Öğretmenler öğrenenlerin öz-yeterlik algılarını yükseltecek etkinlik seçimi, dönüt ve söylemlerde bulunmaları öğrenenlerin verilen göreve başlama, görevi yapmaya devam etme ve tamamlama gibi davranışları geliştirilebilir.
5. Programlama mantığını kazandırmak için programlama eğitiminin başlangıcında akış diyagramları yerine Scratch, Alice gibi blok tabanlı görselleştirici araçlar kullanılarak oyun ve hikâye tasarımı etkinlikleri gerçekleştirilebilir.
6. Programlama öğretime başlamadan önce programlama aracının ayrıntılı olarak tanıtılması, programlama sürecine başlandığında arayüzden kaynaklanabilecek zorlukları azaltacaktır. Programın özellikleri, butonların işlevleri ve nasıl kullanacağı tam olarak öğrenilmeden öğretime başlanırsa, öğrenci geliştireceği algoritmaya odaklanmadan diğer ayrıntılarla uğraşabilir ve derse yönelik motivasyonunu kaybedebilir.
7. Scratch aracı ile programlama öğretime başlamadan önce özellikle küçük yaşlardaki öğrencilerin Scratch'in web sayfasında paylaştığı projelerin öğrencilere gösterilmesi konuya yönelik ilginin çekilmesini sağlayabilir. Bu sayfaya nasıl kayıt olunacağı, burada nelerin yapılabileceği, hali hazırda yayınlanmış projelere yorum yazılması gibi web sayfasının olanakları

öğrencilere tanıtılırsa öğrenciler de bu topluluğun içinde bulunmak isteyebilirler ve derse yönelik güdülenmeleri artırılabilir.

8. Özel eğitime gereksinimi olan öğrencilere algoritmik düşünme becerisinin kazandırılması için Scratchjr aracının kullanımı bir alternatif olabilir. Çünkü Scratchjr 5-7 yaş arası çocuklar için geliştirilmiş, kullanımı daha kolay ve daha eğlenceli tasarıma sahip bir programlama aracıdır. Böylece öğrencinin verilen görevleri adım adım planlayabilme becerisinin geliştirilmesine katkı sağlanabilir.
9. Scratch programının işlevlerinden birisi de öğrencilerin işbirliği becerilerini geliştirmesidir. Şöyle ki; Scratch aracında çevrimdışı tasarlanan projeler Scratch'in kendi web sitesinde, çevrimiçi bir platformda diğer kullanıcılarla paylaşılabilir. Öğretmenler programlama konusunun başlarında Scratch'in web sitesinde bir sınıf grubu oluşturup burada öğrencilerin birlikte etkileşimli olarak proje geliştirebilmeleri için rehberlik ve teşvikte bulunabilirler.
10. E-twinning projelerinde Scratch programlama aracı kullanılarak farklı temalarda oyun ve hikâyeler tasarlanabilir. Böylece öğrenciler ve okullar arasında işbirliği geliştirilebilir.
11. Algoritmik düşünce becerisinin geliştirmek için okul öncesi dönemden itibaren tüm eğitim kademelerinde öğrenci seviyesine uygun olan programlama araçları ile programlama eğitimi verilebilir.
12. Scratch ile programlama etkinlikleri yalnızca programlama becerilerini kazandırmak için değil, aynı zamanda yaratıcılık ve problem çözme gibi 21. yüzyıl becerilerinin de kazandırılması için kullanılabilir.

5.2.2 Araştırmacılara Yönelik Olarak Öneriler

1. Bu çalışma 4 haftalık bir uygulama süreci ile sınırlı kalmıştır. 2017-2018 öğretim yılında güncellenen bilişim teknolojileri ve yazılım dersi öğretim programında programlama ünitesine ayrılan sürenin toplam ders süresinin yarısından daha az olduğu görülmektedir. Daha uzun bir uygulama sürecinde, öğretim programında sunulan örnek etkinliklerle planlanan derste

öğrencilerin öz-yeterlik algıları ve güdülenmelerindeki farklılaşma incelenebilir.

2. Bu çalışmada deneysel süreçte algoritma geliştirmeyi öğrenen öğrencilerin başarı, öz-yeterlik algısı ve güdülenme düzeylerindeki farklılaşmalar incelenmiştir. Benzer bir deneysel sürecin sonunda deney ve kontrol gruplarına ikinci bir deneysel süreç olarak programlama dili öğretildikten sonra başarı, öz-yeterlik algısı ve güdülenme düzeylerindeki farklılaşmalar incelenebilir.
3. Code.org gibi çevrimiçi olarak kullanılan, farklı yaş gruplarına uygun programlama etkinlikleri sunan programlama araçlarının programlama öğretiminde kullanımının öğrencilerin öz yeterlik algısı ve güdülenmelerine etkisi araştırılabilir.
4. Scratch ve Scratchjr programlarının okulların özel alt sınıflarında kullanımının öğrencilerin bilişsel, mantıksal, tasarımsal ve algoritmik düşünme becerilerine etkisi araştırılabilir.
5. Okuldan kaçma davranışı olan, derse ilgisi olmayan öğrencilere okulda Scratch gibi görsel blok tabanlı bir programla aracı ile programlama öğretiminin öğrencilerin okula yönelik güdülenmelerine ve okul devamlılığına etkisi araştırılabilir.

KAYNAKÇA

- Adleberg, B. M. (2013). *Scratch Programming and Remix Culture: Gender Differences in Interaction and Motivation for Pre-Adolescents*. Unpublished post graduate thesis, Georgetown University. Washington, D.C.
- Akçay, A. (2015). *Programlama Becerisi Öz Yeterliğinin Problem Çözme ve Sorgulama Becerileri Bağlamında İncelenmesi*. Yüksek lisans tezi. Necmettin Erbakan Üniversitesi, Eğitim Bilimleri Enstitüsü.
- Akgün, Ö. E. (2008). Bilgisayar Öz-Yeterlik İnançları. *Bilişim Teknolojileri Öğretiminde Sosyo-Psikolojik Değişkenler*. D. Deryakulu (Editör). (1. Baskı). s. 11. Ankara: Maya Akademi.
- Akkoyunlu, B., Orhan, F. ve Umay, A. (2005). Bilgisayar Öğretmenleri İçin "Bilgisayar Öğretmenliği Öz Yeterlik Ölçeği" Geliştirme Çalışması. *Hacettepe Üniversitesi Eğitim Fakültesi Dergisi*, 29 (29), 1-8. http://www.efdergi.hacettepe.edu.tr/shw_artcl-756.html adresinden erişilmiştir.
- Akpınar, Y. ve Altun, Y. (2014). Bilgi Toplumu Okullarında Programlama Eğitimi Gereksinimi. *İlköğretim Online*, 13(1), 1-4. <http://ilkogretim-online.org.tr/index.php/io/article/view/2099> adresinden erişildi.
- Altun, A. ve Mazman, S. G. (2012). Programlamaya İlişkin Öz Yeterlik Algısı Ölçeğinin Türkçe Formunun Geçerlilik ve Güvenirlik Çalışması. *Eğitimde ve Psikolojide Ölçme ve Değerlendirme Dergisi*, 3(2), 297-308. <http://dergipark.ulakbim.gov.tr/epod/article/view/5000045510> adresinden erişildi.
- Arabacıoğlu, C., Bülbül, H. ve Filiz, A. (21 Ocak-2 Şubat 2007). Bilgisayar Programlama Öğretiminde Yeni Bir Yaklaşım. *Akademik Bilişim'07 - IX. Akademik Bilişim Konferansı Bildirileri Kitapçığı*, Dumlupınar Üniversitesi, Kütahya.
- Askar P. ve Umay, A. (2001). İlköğretim Matematik Öğretmenliği Öğrencilerinin Bilgisayarla İlgili Öz-yeterlik Algısı. *Hacettepe Üniversitesi Eğitim*

Fakultesi Dergisi, 21, 1-8.

<http://www.efdergi.hacettepe.edu.tr/yonetim/icerik/makaleler/1020-published.pdf> adresinden erişilmiştir.

Atmatzidou, S. ve Demetriadis, S. (2016). Advancing Students' Computational Thinking Skills Through Educational Robotics: A Study on Age and Gender Relevant Differences. *Robotics and Autonomous Systems*, 75, 661-670. DOI: 10.1016/j.robot.2015.10.008

Bandura, A. (1977). Self-Efficacy: Toward a Unifying Theory of Behavioral Change. *Psychological Review*, 84(2), 191-215. DOI: 10.1016/0146-6402(78)90002-4

Başer, M. (2013). Attitude , Gender and Achievement in Computer Programming. *Middle-East Journal of Scientific Research*, 14(2), 248–255. DOI:10.5829/idosi.mejsr.2013.14.2.2007

Baştemur Kaya, C. ve Çakır, H. (2018). Programlama Dili Öğreniminde Alice Programının Kullanım Sürecinin İncelenmesi. *Eğitimde Nitel Araştırmalar Dergisi- Journal of Qualitative Research in Education*, 6(2), 187-206. DOI:10.14689/issn.2148 - 2624.1.6c2s9m

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

Bilişim Garaj Akademisi (2019). 02.04.2019 tarihinde <https://bilisimgaraji.com/> adresinden edinilmiştir.

BTE (Bilişim Teknolojileri Eğitimcileri Derneği). (2013). Ne oldu, Ne oluyor, Ne olacak?. 29.12.2017 tarihinde http://www.bte.org.tr/belge/ne_oldu_ne_oluyor_ne_olacak_BTE_derneği.pdf adresinden edinilmiştir.

BTE (Bilişim Teknolojileri Eğitimcileri Derneği). (2012). Bilişim Teknolojileri Dersinin ve Öğretmenlerinin Durumu. 29.12.2017 tarihinde http://www.bte.org.tr/belge/BTE_derneği_BT_egitimi_durum_raporu.pdf adresinden edinilmiştir.

- Büyüköztürk, Ş., Akgün, Ö. E., Kahveci, Ö. ve Demirel, F. (2007). Güdülenme ve Öğrenme Stratejileri Ölçeğinin Türkçe Formunun Geçerlik ve Güvenirlik Çalışması. *Kuram ve Uygulamada Eğitim Bilimleri*, 4(2), 207-239.
- Büyüköztürk, Ş. (2011). *Sosyal Bilimler İçin Veri Analizi El Kitabı*. (14. Baskı). Ankara: Pegem Yayıncılık.
- Büyüköztürk, Ş., Çakmak K., E., Akgün, Ö. E., Karadeniz, Ş. ve Demirel F. (2011). *Bilimsel Araştırma Yöntemleri*. (Genişletilmiş 12. Baskı). Ankara: Pegem Akademi
- Büyüköztürk, Ş. (2009). *Sosyal Bilimler İçin Veri Analizi El Kitabı: İstatistik, Araştırma Deseni, Spss Uygulamaları ve Yorum*. (9. Baskı). Ankara: Pegem Yayınları.
- Büyüköztürk, Ş. (2007). *Sosyal Bilimler İçin Veri Analizi El Kitabı*. (7. Baskı). Ankara: Pegem A Yayıncılık.
- Calder, N. (2010). Using Scratch: An Integrated Problem-Solving Approach to Mathematical Thinking. *Australian Primary Mathematics Classroom*, 15(4), 9-14. <https://files.eric.ed.gov/fulltext/EJ906680.pdf> adresinden edinilmiştir.
- Celep, C. (2000). *Eğitimde Örgütsel Adanma ve Öğretmenler*. (1. Baskı). Ankara: Anı Yayıncılık.
- Chiu, C.F. (2014). Teaching Programming Concepts to K-12 Teachers with Scratch. *Journalism and Mass Communication*, 4(2), 125-132.
- Clements, D. H. ve Fullo, D. F. (1984). Effects of Computer Programming on Young Children's Cognition. *Journal of Educational Psychology*. 76(6). 1051-1058. DOI:10.1037/0022-0663.76.6.1051
- Coşar, M. (2013). *Problem Temelli Öğrenme Ortamında Bilgisayar Programlama Çalışmalarının Akademik Başarı, Eleştirel Düşünme Eğilimi ve Bilgisayara Yönelik Tutuma Etkileri*. Doktora tezi. Gazi Üniversitesi, Eğitim Bilimleri Enstitüsü.
- Çakıroğlu, Ü., Sarı, E. ve Akkan, Y. (22-24 Eylül 2011). The View of the Teachers about the Contribution of Teaching Programming to the Gifted Students in the Problem Solving. *Paper Presented at 5th International Computer & Instructional Technologies Symposium*, Fırat Üniversitesi, Elazığ.

- Çatlak, Ş., Tekdal, M. ve Baz, F. (2015). Scratch Yazılımı İle Programlama Öğretiminin Durumu: Bir Doküman İnceleme Çalışması. *Journal of Instructional Technologies & Teacher Education*, 4 (3). <http://dergipark.org.tr/jitte/issue/25088/264774> adresinden edinilmiştir.
- Çelik, A. ve Özdener, N. (2019). Bilgisayarlı ve Bilgisayarsız Programlama Etkinliklerinin Güdülenme Üzerindeki Etkisi. *Akademik Sosyal Araştırmalar Dergisi*, 88, 651-669. DOI: 10.16992/ASOS.14692
- Çepni, S., Bayrakçeken, S., Yılmaz, A., Yücel, C., Semerci, Ç., Köse, E., Sezgin, F., Demircioğlu, G. ve Gündoğdu, K. (2008). *Ölçme ve Değerlendirme*. (4. Baskı). Ankara: Pegem Akademi.
- Çokluk, Ö., Şekercioğlu, G. ve Büyüköztürk, Ş. (2016). *Sosyal Bilimler İçin Çok Değişkenli İstatistik: Spss ve Lisrel Uygulamaları*. (4. Baskı). Ankara: Pegem Akademi.
- Demir, F. (2015). *Programlama Öğretiminde Eğitsel Programlama Dilinin Farklı Kullanımlarının Programlama Başarısı ve Kaygısına Etkisi*. Doktora tezi. Atatürk Üniversitesi, Eğitim Bilimleri Enstitüsü.
- Demirer, V. ve Sak, N. (2016). Programming Education and New Approaches Around The World and in Turkey / Dünyada ve Türkiye'de Programlama Eğitimi ve Yeni Yaklaşımlar. *Eğitimde Kuram ve Uygulama*, 12 (3), 521-546.
- Dilekmen, M. ve Ada, Ş. (2005). Öğrenmede Güdülenme. *Kazım Karabekir Eğitim Fakültesi Dergisi*, 11, 113-123. <http://e-dergi.atauni.edu.tr/ataunikkefd/article/view/1021003972> adresinden edinilmiştir.
- Doğan, U. ve Kert, S. (2016). Bilgisayar Oyunu Geliştirme Sürecinin, Ortaokul Öğrencilerinin Eleştirel Düşünme Becerilerine ve Algoritma Başarılarına Etkisi. *Boğaziçi Üniversitesi Eğitim Dergisi*, 33 (2), 21-42. <https://dergipark.org.tr/buje/issue/29693/319507> adresinden edinilmiştir.
- EARGED. (2011). *MEB 21. Yüzyıl Öğrenci Profili*. 02.04.2019 tarihinde https://www.meb.gov.tr/earged/earged/21.%20yy_og_pro.pdf adresinden edinilmiştir.

- Eker, M. (2011). *Algoritmayı Anlamak*. (5. Baskı). Ankara: Nirvana Yayınları.
- Ekici, G. (2006). Meslek Lisesi Öğretmenlerinin Öğretmen Öz Yeterlik İnançları Üzerine Bir Araştırma. *Eurasian Journal of Educational Research*, 24, 87-96.
https://www.researchgate.net/publication/284264508_Meslek_lisesi_Ogretmenlerinin_Ogretmen_Ozyeterlik_Inanclari_Uzerine_bir_arastirma adresinden edinilmiştir.
- Eğitim Bilişim Ağı (2019). <http://codeweekturkiye.eba.gov.tr/> 10.04.2019 tarihinde adresinden edinilmiştir.
- European Commission. (2018). Coding-The 21st Century Skill. 01.04.2019 tarihinde <https://ec.europa.eu/digital-single-market/en/coding-21st-century-skill> adresinden edinilmiştir.
- Erdoğan, B. (2005). *Programlama Başarısı ile Akademik Başarı, Genel Yetenek, Bilgisayara Karşı Tutum, Cinsiyet ve Lise Türü Arasındaki İlişkilerinin İncelenmesi*. Yüksek lisans tezi. Marmara Üniversitesi, Eğitim Bilimleri Enstitüsü.
- Erol, O. (2015). *Scratch İle Programlama Öğretiminin Bilişim Teknolojileri Öğretmen Adaylarının Motivasyon ve Başarılarına Etkisi*. Yüksek lisans tezi. Anadolu Üniversitesi, Eğitim Bilimleri Enstitüsü.
- Erol, O. ve Kurt, A. A. (2017). The Effects of Teaching Programming with Scratch on Pre-Service Information Technology Teachers' Motivation and Achievement. *Computers in Human Behavior*, 77, 11-18.
DOI:10.1016/j.chb.2017.08.017
- Ersoy, H., Madran, O. ve Gülbahar, Y. (2- 4 Şubat 2011). Programlama Dilleri Öğretimine Bir Model Önerisi: Robot Programlama. *Akademik Bilişim Konferansı*. İnönü Üniversitesi, Malatya.
- Eryılmaz, S. (2003). *Algoritma Tasarlama ve Programlamaya Giriş*. (1. Baskı). Ankara: Detay Yayıncılık.
- Feddon, J. S. ve Charness, N. (5-7 Ocak 1999). Component Relationships Depend on Skill in Programming. *11th Annual PPIG Workshop*, University of Leeds, UK.

- Fidan, N. (2012). *Okulda Öğrenme ve Öğretme*. (3. Baskı). Ankara: Pegem Akademi.
- Futschek, G. (2006). Algorithmic Thinking: The Key for Understanding Computer Science. *Informatics Education–The Bridge between Using and Understanding Computers Lecture Notes in Computer Science*, 4226, 159-168. DOI:10.1007/11915355_15
- Genç, Z. ve Karakuş, S. (22-24 Eylül 2011). Tasarımla Öğrenme: Eğitsel Bilgisayar Oyunları Tasarımında Scratch Kullanımı. *International Computer and Instructional Technologies Symposium ICITS*. Fırat Üniversitesi, Elazığ.
- Gökoğlu, S. ve Yüksel, D. (16-18 Mayıs 2016). Bilgisayar Programcılığı Öğrencilerinin Algoritma ile İlgili Metaforları. *10th International Computer and Instructional Technologies Symposium (ICITS)*, Recep Tayyip Erdoğan Üniversitesi, Rize.
- Gülmez, I. (2009). *Programlama Öğretiminde Görselleştirme Araçlarının Kullanımının Öğrenci Başarı ve Motivasyonuna Etkisi*. Yüksek lisans tezi. Marmara Üniversitesi, Eğitim Bilimleri Enstitüsü.
- Gültekin, K. (2006). *Çokluortamın Bilgisayar Programlama Başarısı Üzerine Etkisi*. Yüksek lisans tezi. Hacettepe Üniversitesi, Fen Bilimleri Enstitüsü.
- Günüç, S., Odabaşı, H.F. ve Kuzu, A. (2013). 21. Yüzyıl Öğrenci Özelliklerinin Öğretmen Adayları Tarafından Tanımlanması: Bir Twitter Uygulaması, *Eğitimde Kuram ve Uygulama*, 9(4), 436-455.
- Hawi, N. (2010). Causal attributions of success and failure made by undergraduate students in an introductory-level computer programming course. *Computers & Education*, 54, 1127–1136. DOI: 10.1016/j.compedu.2009.10.020
- Hazır Bıkmaz, F. (2004). “Sınıf Öğretmenlerinin Fen Öğretiminde Öz Yeterlilik İnancı” Ölçeğinin Geçerlik ve Güvenirlik Çalışması. *Milli Eğitim Dergisi*, 161.
http://dhgm.meb.gov.tr/yayimlar/dergiler/Milli_Egitim_Dergisi/161/bikmaz.htm adresinden edinilmiştir.

- Hu, M. (2004). Teaching Novices Programming with Core Language and Dynamic Visualization. *Papers from the Proceedings of the 17th NACCQ (National Advisory Committee on Computing Qualifications)* s. 94-103.
- İnce, İ., Şenyüzlü, B. ve Uğur, B. (2007). *İlköğretim Bilişim Teknolojileri 6, 7 Ve 8. Basamak Öğretmen Kılavuz Kitabı*. (1.Baskı). Ankara: MEB.
- İmal, N. ve Eser, M. (22-24 Ekim 2009). Programlama Dili Öğrenmedeki Zorluklar ve Çözüm Yaklaşımları. *Elektrik Elektronik Bilgisayar Biyomedikal Mühendislikleri Eğitimi IV. Ulusal Sempozyumu*, Osmangazi Üniversitesi, Eskişehir.
- ISTE (International Society for Technology in Education). (2016). *ISTE Standarts for Students 2016*. 10.04.2019 tarihinde <https://www.iste.org/standards/for-students> adresinden edinilmiştir.
- ISTE (International Society for Technology in Education). (2011). *Operational Definition of Computational Thinking for K-12 Education*. 10.04.2019 tarihinde <https://id.iste.org/docs/ct-documents/computational-thinking-operational-definition-flyer.pdf> adresinden edinilmiştir.
- Kafai, Y. B. (2006). Playing and Making Games for Learning Instructionist and Constructionist Perspectives for Game Studies. *Games and Culture*, 1(1), 36-40. DOI: 10.1177/1555412005281767
- Kafai, Y.B., Peppler, K.A., ve Chin, G. (29-30 Haziran 2007). High Tech Programmers in Low Income Communities: Creating a Computer Culture in a Community Technology Center. In C. Steinfeld, B. Pentland, M. Ackermann, & N. Contractor (Editörler.), *Proceedings of the Third International Conference on Communities and Technology*, Michigan State University, s. 545-562. New York: Springer.
- Kalelioğlu, F. ve Gülbahar, 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. https://www.mii.lt/informatics_in_education/pdf/INFE232.pdf adresinden edinilmiştir.

- Kan, A. ve Akbař, A. (2005). Lise Öğrencilerinin Kimya Dersine Yönelik Tutum Ölçeđi Geliřtirme Çalıřması. *Mersin Üniversitesi Eğitim Fakültesi Dergisi*, 1(2), 227-237.
<http://dergipark.ulakbim.gov.tr/mersinefd/article/view/5000003016> adresinden edinilmiřtir.
- Karabak, D. ve Güneř, A. (2013). Ortaokul Birinci Sınıf Öğrencileri için Yazılım Geliřtirme Alanında Müfredat Önerisi. *Eđitim ve Öğretim Arařtırmaları Dergisi*, 2(3), 175-181.
<http://www.jret.org/FileUpload/ks281142/File/21b.karabak.pdf> adresinden edinilmiřtir.
- Kasalak, İ. (2017). *Robotik Kodlama Etkinliklerinin Ortaokul Öğrencilerinin Kodlamaya İliřkin Öz-Yeterlik Algularına Etkisi Ve Etkinliklere İliřkin Öğrenci Yařantıları*. Yüksek lisans tezi. Hacettepe Üniversitesi, Eğitim Bilimleri Enstitüsü.
- Kert, S. B. ve Uđrař, T. (1-3 Mayıs 2009). Programlama Eğitiminde Sadelik ve Eğlence: Scratch Örneđi. *I. Uluslararası Eğitim Arařtırmaları Kongresi*, Onsekiz Mart Üniversitesi, Çanakkale.
- Kesici, T. ve Kocabař, Z. (2007). *Bilgisayar-1 Liseler için*. (1. Baskı). Ankara: MEB.
- Kim, S., Chung, K. ve Yu, H. (2013). Enhancing Digital Fluency Through a Training Program for Creative Problem Solving Using Computer Programming. *The Journal of Creative Behavior*, 47(3), 171-199. DOI: 10.1002/jocb.30
- Koorsse, M., Cilliers, C. ve Calitz, A. (2015). Programming Assistance Tools to Support the Learning of IT Programming in South African Secondary Schools. *Computers & Education*, 82, 162-178.
DOI: 10.1016/j.compedu.2014.11.020
- Korhonen, A. (2003). *Visual Algorithm Simulation*. Doctoral dissertation. Helsinki University of Technology Department of Computer Science and Engineering. Espoo, Finland.
- Korkmaz, O. (2016). The Effect Of Lego Mindstorms Ev3 Based Design Activities On Students' Attitudes Towards Learning Computer Programming, Self-

- Efficacy Beliefs And Levels Of Academic Achievement. *Baltic Journal of Modern Computing*, 4(4), 994–1007. DOI:10.22364/bjmc.2016.4.4.24
- Kukul, V. ve Gökçearsan, Ş. (18-20 Eylül 2014). Scratch ile Programlama Eğitimi Alan Öğrencilerin Problem Çözme Becerilerinin İncelenmesi. 8. *Uluslararası Bilgisayar ve Öğretim Teknolojileri Sempozyumu*, s.58-63. Trakya Üniversitesi, Edirne.
- Lewis, C. M. (10-13 Mart 2010). How Programming Environment Shapes Perception, Learning and Goals: Logo vs. Scratch. *Proceedings of the 41st ACM Technical Symposium on Computer Science Education*. s. 346-350. ACM, New York.
- Maloney, J. H., Peppler, K., Kafai, Y., Resnick, M. ve Rusk, N. (2008). Programming by Choice: Urban Youth Learning Programming with Scratch. *ACM SIGCSE Bulletin*, 40(1), 367-371. DOI: 10.1145/1352322.1352260
- Maloney, J., Resnick, M., Rusk, N., Silverman, B. ve Eastmond, E. (2010). The Scratch Programming Language and Environment. *ACM Transactions on Computing Education (TOCE)*,10(4). DOI: 10.1145/1868358.1868363
- Milli Eğitim Bakanlığı. (2018). 2023 Eğitim Vizyonu. 10.03.2019 tarihinde http://2023vizyonu.meb.gov.tr/doc/2023_EGITIM_VIZYONU.pdf adresinden edinilmiştir.
- Milli Eğitim Bakanlığı. (2012). *Bilişim Teknolojileri ve Yazılım Dersi (5, 6, 7 ve 8. Sınıflar) Öğretim Programı*. 04.12.2017 tarihinde <http://ttkb.meb.gov.tr/program2.aspx> adresinden erişilmiştir.
- Milli Eğitim Bakanlığı. (2010). *İlköğretim Seçmeli Bilgisayar(1-8) Dersi Öğretim Programı*. 04.12.2017 tarihinde <http://ttkb.meb.gov.tr/program2.aspx> adresinden erişilmiştir.
- Mihçı, C., Taçgın, Z., ve Arslan, A. (2017). Türkiye’deki Böte Programlarında Sunulan Programlama Derslerine Yönelik Bir Durum Çalışması. *The Journal of Academic Social Science (ASOS)*, 5(45), 504-529. http://www.asosjournal.com/Makaleler/2129582253_12257%20Zeynep%20TA%c3%87GIN.pdf adresinden edinilmiştir.

- Nikou, S. A. ve Economides, A. A. (3-5 Nisan 2014). Transition in Student Motivation During a Scratch and an App Inventor Course. *Global Engineering Education Conference (EDUCON) IEEE* (1042-1045), İstanbul.
- Önen, F. ve Öztuna, A. (5 Mart 2005). Fen Bilgisi ve Matematik Öğretmenlerinin Öz Yeterlik Duygusunun Belirlenmesi. *İstek Vakfı Okulları I. Fen ve Matematik Öğretmenleri Sempozyumu*. Yeditepe Üniversitesi, İstanbul.
- Özkan, Y. (2003). *Programlama Dilleri: C ile Programlama*. (1. Baskı). İstanbul: Alfa Yayınları.
- Özmen, B. ve Altun, A. (2014). Undergraduate Students' Experiences in Programming: Difficulties and Obstacles. *Turkish Online Journal of Qualitative Inquiry*, 5(3), 9–27. DOI:10.17569/tojqi.20328
- Özmen, B. ve Varol, F. (22-24 Eylül 2011). Uzman, Aile ve Öğretmen Gözü ile Eğitim Yazılımları: Eyades. *5 th International Computer & Instructional Technologies Symposium*, Fırat Üniversitesi, Elazığ.
- Özoran, D., Cagiltay, N. E. ve Topallı, D. (31 Ekim- 3 Kasım 2012). Using Scratch in Introduction to Programming Course for Engineering Students. *Proceedings of 2nd International Engineering Education Conference (IEEC2012)*, Antalya.
- Pajares, F. (2002). Overview of Social Cognitive Theory and Self-Efficacy. <http://www.emory.edu/EDUCATION/mfp/eff.html> adresinden edinilmiştir.
- Resnick, M. (2012). Reviving Papert's Dream. *Educational Technology*, 52(4), 42-46. <https://www.media.mit.edu/publications/reviving-paperts-dream/> adresinden edinilmiştir.
- Resnick, M. (2007). Sowing The Seeds for a More Creative Society. *Learning and Leading with Technology*, 35(4), 18-22. <https://web.media.mit.edu/~mres/papers/Learning-Leading.pdf> adresinden edinilmiştir.
- Resnick, M., Kafai, Y., Maloney, J., Rusk, N., Burd, L., & Silverman, B. (2003). A Networked, Media-Rich Programming Environment to Enhance Technological Fluency at After-School Centers in Economically-

Disadvantaged Communities. Proposal to National Science Foundation.
Information Technology Research.

DOI:10.1145/1592761.1592779

Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B. ve Kafai, Y. (2009). Scratch: Programming for All. *Communications of The ACM*, 52(11), 60-67.
DOI:10.1145/1592761.1592779

Rizvi, M., Humphries, T., Major, D., Jones, M. ve Lauzun, H. (2011). A CS0 Course Using Scratch. *Journal of Computing Sciences in Colleges*, 26(3), 19-27.
<https://dl.acm.org/citation.cfm?id=1859166&dl=ACM&coll=DL> adresinden edinilmiştir.

Scratch. (2019a). Scratch Hakkında. 01.04.2019 tarihinde
<https://scratch.mit.edu/about/> adresinden edinilmiştir.

Scratch. (2019b). Bir Bakışta Topluluk İstatistikleri. 01.04.2019 tarihinde
<https://scratch.mit.edu/statistics/> adresinden edinilmiştir.

ScratchEd Team. (2011). Creative Computing - a Design-based Introduction to Computational Thinking. MIT Media Lab. 28.03.2019 tarihinde
<http://scratched.gse.harvard.edu/sites/default/files/curriculumguide-v20110923.pdf> adresinden edinilmiştir.

Tekerek, M. ve Altan, T. (2014). The Effect of Scratch Environment on Student's Achievement in Teaching Algorithm. *World Journal on Educational Technology*, 6(2), 132-138.
http://archives.sproc.org/index.php/wjet/article/view/2791/pdf_288 adresinden edinilmiştir.

Sayın, Z. ve Seferoğlu, S. S. (30 Ocak- 5 Şubat 2016). Yeni Bir 21. Yüzyıl Becerisi Olarak Kodlama Eğitimi ve Kodlamının Eğitim Politikalarına Etkisi. *XVIII. Akademik Bilişim Konferansı*, Adnan Menderes Üniversitesi, Aydın.

Saygıner, Ş. ve Tüzün, H. (24-26 Mayıs 2017). Programlama Eğitiminde Yaşanan Zorluklar Ve Çözüm Önerileri. *11. Uluslararası Bilgisayar ve Öğretim Teknolojileri Sempozyumu*, İnönü Üniversitesi, Malatya.

- Senemođlu, N. (2012). *Gelişim, Öğrenme ve Öğretim - Kuramdan Uygulamaya* (21. Baskı). Ankara: Pegem Akademi.
- Seferođlu, S. S. ve Akbıyık, C. (2005). İlköğretim Öğretmenlerinin Bilgisayara Yönelik Öz-Yeterlik Algıları Üzerine Bir Çalışma. *Eğitim Araştırmaları Dergisi*. 19, 89-101. DOI: 10.7822/egt206+
- Tabachnick, B. G. ve Fidel, L. S. (2014). *Using Multivariate Statistics*. USA: Pearson Education Limited
- TTKB. (2017). *Bilişim Teknolojileri Ve Yazılım Dersi Öğretim Programı*. 04.12.2017 tarihinde <http://mufredat.meb.gov.tr/Dosyalar/> adresinden edinilmiştir.
- TTKB. (2016). *Ortaöğretim Bilgisayar Bilimi Dersi (Kur 1, Kur 2) Öğretim Programı*. 10.03.2019 tarihinde <http://mufredat.meb.gov.tr/Dosyalar/> adresinden edinilmiştir.
- TTKB. (2012). *Bilişim Teknolojileri Ve Yazılım Dersi (5, 6, 7 Ve 8. Sınıflar) Öğretim Programı*. 04.12.2017 tarihinde <http://ttkb.meb.gov.tr/program2.aspx> adresinden erişilmiştir.
- URL1: <https://indigodergisi.com/2017/02/cocuklarin-dili-kodlama/> 2 Nisan 2019 tarihinde adresinden edinilmiştir.
- Utting, I., Cooper, S., Kölling, M., Maloney, J. ve Resnick, M. (2010). Alice, Greenfoot and Scratch –A Discussion. *ACM Transactions on Computing Education*, 10(4). DOI:10.1145/1868358.1868364
- Üçgül, M. (2013). Bilgisayar Oyunlarının Öğrenci Gütülenmesine Etkisi. *Amasya Üniversitesi Eğitim Fakültesi Dergisi* 2(1), 71-86. <http://dergi.amasya.edu.tr/article/view/1031000062> adresinden erişilmiştir.
- Westcott, S. (2008). *Effectiveness of Using Digital Game Playing in a First-Level Programming Course*. Doctoral dissertation. PaceUniversity, ABD.
- Yaşar, M. (2014). İstatistiğe Yönelik Tutum Ölçeği: Geçerlilik ve Güvenirlik Çalışması. *Pamukkale Üniversitesi Eğitim Fakültesi Dergisi*, 36(3), 59-75.

- Yecan, E., Özçınar, H. ve Tanyeri, T. (2017). Bilişim Teknolojileri Öğretmenlerinin Görsel Programlama Öğretimi Deneyimleri. *Elementary Education Online*, 16(1), 377-393. DOI: 10.17051/io.2017.80833
- Yıldırım Doğru, E. (2012). *Matematik Öğretiminde Kullanılan Ayrılıp Birleşme Tekniğinin Öğrencilerin Özyeterlik, Kaygı ve Kalıcılık Düzeylerine Etkisi*. Yüksek lisans tezi, Akdeniz Üniversitesi, Sosyal Bilimler Enstitüsü.
- YÖK (Yükseköğretim Kurulu). (2010). 2010-2011 Akademik Yılında Uygulanacak olan MYO Öğretim Programları. Web: <http://ikmep.yok.gov.tr/> 10.05.2016 tarihinde adresinden edinilmiştir.
- Yükseltürk E. ve Altıok, S. (2015). Bilişim Teknolojileri Öğretmen Adaylarının Bilgisayar Programlama Öğretimine Yönelik Görüşleri. *Amasya Üniversitesi Eğitim Fakültesi Dergisi*, 4(1), 50-65. <http://dergi.amasya.edu.tr/article/view/5000087734> adresinden erişilmiştir.

EKLER

EK 1. ÖĞRENME ETKİNLİKLERİNİN GÜNLÜK PLANLARI

ÇALIŞMA PLANI

1.HAFTA

BÖLÜM 1

Dersin Adı:	Bilişim Teknolojileri ve Yazılım
Sınıf:	6. Sınıf
Ünitenin Adı/No:	Problem Çözme, Programlama
Konu	Algoritma ve Strateji Geliştirme
Önerilen Süre	40 dk + 40 dk

BÖLÜM 2

Öğrenci Kazanımları /Hedef ve Davranışlar	<ul style="list-style-type: none">• Problemin tanımını bilir.• Bir problemin çözümü için yapılması gereken işlem adımlarını bilir.• Algoritmanın ve programlamanın tanımını bilir.• Belirlenen problemin çözümü için algoritma geliştirilmesinin önemini ifade eder.
Ünite Kavramları ve Sembolleri/Davranış Örüntüsü	Problem, algoritma, programlama.
Öğretme-Öğrenme-Yöntem ve Teknikleri	Anlatım, Soru-Yanıt, Gösterip Yaptırma.
Kullanılan Eğitim Teknolojileri-Araç, Gereçler ve Kaynakça * Öğretmen * Öğrenci	Etkileşimli Tahta Deney grubu için Scratch programı, tablet, internet bağlantısı
Öğretme-Öğrenme Etkinlikleri:	
<ul style="list-style-type: none">• Dikkati Çekme• Güdüleme• Gözden Geçirme• Derse Geçiş• Bireysel Öğrenme Etkinlikleri (Ödev, deney, problem çözme vb.)• Grupla Öğrenme Etkinlikleri• Özet	<p>Öğrencilere önceki dönem öğrendikleri yazılım konusu ve yazılımın ne olduğu sorulur. Bildikleri çeşitli bilgisayar yazılımlarına örnek verilmeleri istenir. Yazılımların çeşitli amaçlar için geliştirildiğine dikkat çekilir. Her yazılımın bir sorunumuzu çözdüğü söylenir. Problem kavramı açıklanır.</p> <p>Öğrencilere günlük hayatta karşılaştıkları problemleri nasıl çözdükleri sorusu yöneltilir. Önce problemi anlamaya çalıştıkları ardından problemi en kısa yoldan çözmek istediklerine değinilir. Problemleri adım adım bazı işlemler yaparak çözdükleri söylenir. Bilgisayarların da problemleri verilen komutları adım adım uygulayarak çözdüklerine dikkat çekilir. Günlük hayattan örnek problem durumları ve çözüm aşamaları gösterilir.</p> <p>Nasıl ki bir film çekilmeden önce senaryosu yazılıyor, bir bina yapılmadan önce planı çiziliyorsa bilgisayar yazılımları da üretilmeye başlanmadan önce yazılımın komutlara nasıl tepki vereceği, hangi işlemleri yapacağı, problemleri nasıl çözeceğini gösteren bir plan yapılır. Bu plana algoritma adı verildiği açıklanır. Algoritma oluşturmamanın önemine değinilir. Plansız yapılan bir binada ne gibi sorunlar ortaya çıkabileceği sorulur. Düzeltmesi mümkün olmayan sorunlar olabileceğine dikkat çekilir. Kusursuz bir bina için nasıl ki plan çizimi şartsa kusursuz bir yazılım için de algoritma</p>

	<p>oluşturmak o derece önemli olduğu vurgulanır.</p> <p>Örnek olarak navigasyon cihazının amacı sorulur. Navigasyon cihazı, kullanıcının bir hedefe varabilmesi için adım adım işlemler yaptırması olarak açıklanabilir.</p> <ul style="list-style-type: none"> • Günlük hayattan bir probleme yönelik algoritma oluşturmaları istenir. • (Kontrol) Öğrencilerden bir arkadaşlarını komut vererek yürütmeleri istenir. • (Deney)Başlangıç olarak kedinin yürütmesi, koşması, bir arabanın hareket etmesi için gereken komutlarla uygulama yaptırılır.
--	--

BÖLÜM 3

Ölçme-Değerlendirme	
<ul style="list-style-type: none"> • Bireysel öğrenme etkinliklerine yönelik Ölçme-Değerlendirme • Grupla öğrenme etkinliklerine yönelik Ölçme-Değerlendirme • Öğrenme güçlüğü olan öğrenciler ve ileri düzeyde öğrenme hızında olan öğrenciler için ek Ölçme-Değerlendirme etkinlikleri 	<ul style="list-style-type: none"> • Öğrencilerden örnek verecekleri problemlerin çözüm yolunu bu adımlara uygun olarak anlatmaları istenir. • Öğrencilerin örnek verdikleri problemin çözüm yolunun yukarıda belirlenen adımlara uygunluğunu öğrenciyle birlikte analiz edilir. • Öğrencilerden dış fırcalamak için gerekli algoritma adımlarını bir kağıda doğru sırada planlayarak yazmaları istenir. • Öğrencilerden istenen örnek algoritmanın adımlarının doğruluğu değerlendirilir.

ÇALIŞMA PLANI

2.HAFTA

BÖLÜM 1

Dersin Adı:	Bilişim Teknolojileri ve Yazılım
Sınıf:	6. Sınıf
Ünitenin Adı/No:	Problem Çözme, Programlama
Konu	Algoritma ve Strateji Geliştirme (Akış şeması,sabit,değişkenler)
Önerilen Süre	40 dk + 40 dk

BÖLÜM 2

Öğrenci Kazanımları /Hedef ve Davranışlar	<ul style="list-style-type: none">• Akış şeması, değişken ve sabit kavramlarını açıklar.• Sabit ve değişken kullanarak basit matematik işlemlerine yönelik akış şeması oluşturur.• Farklı programların algoritmalarını inceleyerek en hızlı ve doğru çözümü seçer.
Ünite Kavramları ve Sembolleri/Davranış Örüntüsü	Akış şeması, sabit, değişken.
Öğretme-Öğrenme-Yöntem ve Teknikleri	Anlatım, Soru-Yanıt, Gösterip Yaptırma.
Kullanılan Eğitim Teknolojileri-Araç, Gereçler ve Kaynakça * Öğretmen * Öğrenci	Etkileşimli Tahta Deney grubu için Scratch programı, tablet, bilgisayar, internet bağlantısı
Öğretme-Öğrenme Etkinlikleri:	
<ul style="list-style-type: none">• Dikkati Çekme• Güdüleme• Gözden Geçirme• Derse Geçiş• Bireysel Öğrenme Etkinlikleri (Ödev, deney, problem çözme vb.)• Grupla Öğrenme• Özet	<p>Öğrencilere hayatın her alanında problemlere çözüm üretmek için algoritma kurulduğu anlatılır. Daha sonra bir algoritma yazılırken nelere dikkat edilmesi gerektiği ve algoritma hazırlama adımları anlatılır.</p> <p>Akış şeması, algoritmanın, görsel olarak ifade edilmesidir.</p> <p>Oluşturulan algoritmaların akış şeması ile ifade edilmesi algoritmanın daha iyi anlaşılabilir olmasını, sorunun çözüm basamaklarını, birbirleri ile ilişkilerini daha kolay grülmesini ve yanlışlıkların düzeltilebilmesini sağlar.</p> <p>Algoritma geliştirildikten sonra, daha iyi anlaşılabilir olması ve programlama dillerine aktarımı daha kolay olması nedeniyle, akış şeması haline getirilir. Böylece sorunun çözüm basamakları, birbirleri ile ilişkileri ve bilgi akışı daha kolay görülebilir ve yanlışlıklar düzeltilebilir.</p>

	<p>Bunun için simge ya da semboller kullanılır. Çizgiler, Dörtgen, daire vb. geometrik şekillerle ve izlenecek yolların oklarla ifade edilerek algoritmanın gösterilmesini sağlar. Başka bir deyişle her şekil, yapılacak bir işi veya komutu gösterir.</p> <p>Algoritma yazarken; Algoritma hazırlanırken, çözüm için yapılması gerekli işlemler, öncelik sıraları göz önünde bulundurularak ayrıntılı bir biçimde tanımlanmalıdır.</p> <p>Algoritma Hazırlama Adımları (3)</p> <ol style="list-style-type: none"> 1. Önce algoritmada kullanılacak değişken, sabit isimleri belirlenir. 2. Varsa veri girişleri yapılır. 3. Yapılacak işlem ve formüller yazılır. 4. Sonuçlar ekrana yazdırılır veya çevre birimlerine gönderilir. <p>Sabit: Programdaki değeri değişmeyen ifadelere "sabit" denir.</p> <p>Değişken: Programın her çalıştırılmasında, farklı değerler alan bilgi/bellek alanlarıdır.</p> <ul style="list-style-type: none"> • Verilen iki sayının toplamını ekrana yazan programın algoritması oluşturulur(Sabit) • Klavyeden girilen iki sayının toplamını ekrana yazan programın algoritması oluşturulur. (değişken) <p>Kontrol grubunda akış şeması çizilir, deney grubunda ise scratch programında yapılır.</p>
--	---

BÖLÜM 3

<p>Ölçme-Değerlendirme</p> <ul style="list-style-type: none"> • Bireysel öğrenme etkinliklerine yönelik Ölçme-Değerlendirme • Grupla öğrenme etkinliklerine yönelik Ölçme-Değerlendirme • Öğrenme güçlüğü olan öğrenciler ve ileri düzeyde öğrenme hızında olan öğrenciler için ek Ölçme-Değerlendirme etkinlikleri 	<ul style="list-style-type: none"> • Öğrencilere örnek olarak verilen hazır algoritmaların amacı sorulur. • Öğrencilerden istenen örnek algoritmanın adımlarının doğruluğu değerlendirilir. • Verilen algoritmada farklı değişken değerlerine göre algoritma incelenir.
---	--

ÇALIŞMA PLANI

3.HAFTA

BÖLÜM 1

Dersin Adı:	Bilişim Teknolojileri ve Yazılım
Sınıf:	6. Sınıf
Ünitenin Adı/No:	Problem Çözme, Programlama
Konu	Algoritma ve Strateji Geliştirme (Koşul yapıları)
Önerilen Süre	40 dk + 40 dk

BÖLÜM 2

Öğrenci Kazanımları /Hedef ve Davranışlar	<ul style="list-style-type: none">• Koşul yapısı içeren basit bir algoritma oluşturur.• Koşul yapısını ve değişken ile sabit içeren algoritma oluşturur.
Ünite Kavramları ve Sembolleri/Davranış Örüntüsü	Koşul, Eğer ise.
Öğretme-Öğrenme-Yöntem ve Teknikleri	Anlatım, Soru-Yanıt, Gösterip Yaptırma.
Kullanılan Eğitim Teknolojileri-Araç, Gereçler ve Kaynakça * Öğretmen * Öğrenci	Etkileşimli Tahta Deney grubu için Scratch programı, tablet, bilgisayar, internet bağlantısı
Öğretme-Öğrenme Etkinlikleri:	<p>Öğretmen “Eğer hava bugün güzel olursa bahçede top oynayabiliriz” diyerek burada bir koşul olduğuna dikkat çeker. Daha sonra öğrencilere günlük hayatta hangi durumlarda eğer ifadesini kullandıklarını sorar. Programlamada da aynı şekilde koşul kullanıldığı açıklanır.</p> <p>Koşul yapısında cümle kuruluşu şu şekildedir; Eğer sağlıyor ise, evet ise,den küçük ise gibi.</p> <p>Eğer not ortalaması 50’den küçük ise kaldı, 50 ve üzeri ise sınıfı geçer .</p> <ul style="list-style-type: none">• Birkaç öğrenciden öğretmenin verdiği örneğe benzer şekilde gerçek hayatta eğer ifadesini içeren birer örnek vermeleri istenir. Daha sonra öğretmen günlük hayatta kullanılan bir aygıtın koşul kullanma durumuna örnek verir. (Örr: Bilgisayarda kayıtlı bir videonun üzerine çift tıkladığında açılması.) <p>Etkinliklerde,</p>

	<p>Deney grubunda Scratch programı kullanılır.</p> <ul style="list-style-type: none"> • Eğer komutunu kullanan basit bir programın algoritması oluşturulur. • Eğer komutunu kullanacak şekilde değişken ve sabit içeren bir programın algoritması oluşturulur. <p>Örnek: Klavyeden girilen iki sayıdan büyük olanın ekrana yazdırıldığı bir algoritma oluşturulur.</p> <p>Örnek: Klavyeden girilen sayının pozitif, negatif, sıfır olduğunu ekrana yazdıran programın algoritmasını yazması istenir.</p>
--	--

BÖLÜM 3

<p>Ölçme-Değerlendirme</p> <ul style="list-style-type: none"> • Bireysel öğrenme etkinliklerine yönelik Ölçme-Değerlendirme • Grupla öğrenme etkinliklerine yönelik Ölçme-Değerlendirme • Öğrenme güçlüğü olan öğrenciler ve ileri düzeyde öğrenme hızında olan öğrenciler için ek Ölçme-Değerlendirme etkinlikleri 	<ul style="list-style-type: none"> • Öğrencilere, etkinlikleri yaparken, koşul yapısıyla ilgili doğru yanlış soruları sorularak verilen cevaplara göre değerlendirme yapılabilir. • Oluşturdukları algoritmalar öğrencilerin düzeylerine göre rubrik yardımıyla değerlendirilebilir.
---	--

ÇALIŞMA PLANI

4.HAFTA

BÖLÜM 1

Dersin Adı:	Bilişim Teknolojileri ve Yazılım
Sınıf:	6. Sınıf
Ünitenin Adı/No:	Problem Çözme, Programlama
Konu	Algoritma ve Strateji Geliştirme (Döngü yapıları)
Önerilen Süre	40 dk + 40 dk

BÖLÜM 2

Öğrenci Kazanımları /Hedef ve Davranışlar	<ul style="list-style-type: none">Algoritmada döngü kavramını açıklar.Algoritma oluştururken tekrarlayan işlemlerde sayaç kullanır.Algoritma ve akış şemalarındaki hataları düzeltir.
Ünite Kavramları ve Sembolleri/Davranış Örüntüsü	Döngü, sayaç.
Öğretme-Öğrenme-Yöntem ve Teknikleri	Anlatım, Soru-Yanıt, Gösterip Yaptırma.
Kullanılan Eğitim Teknolojileri-Araç, Gereçler ve Kaynakça * Öğretmen * Öğrenci	Etkileşimli Tahta Deney grubu için Scratch programı, tablet, bilgisayar, internet bağlantısı
Öğretme-Öğrenme Etkinlikleri:	
<ul style="list-style-type: none">Dikkati ÇekmeGüdülemeGözden GeçirmeDerse GeçişBireysel Öğrenme Etkinlikleri (Ödev, deney, problem çözme vb.)Özet	<p>Öğretmen “Trafik lambasında bekleyen bir yayanın karşıya geçmeden önce kırmızı ışık yanana kadar trafik lambasına bakması gerektiğini ” söyleyerek burada tekrarlayan bir işlem olduğuna dikkat çeker. Daha sonra öğrencilere günlük hayatta hangi durumlarda tekrarlayan işlemlerin olduğu sorulur. Programlamada da aynı şekilde bazı işlemlerin döngü olarak belli sayıda tekrar ettiği açıklanır.</p> <p>Programlardaki belirli işlem bloklarını, verilen sayıda gerçekleştiren işlem akış çevrimlerine “döngü” denir.</p> <p>Bu yapı kullanılırken, döngü sayacı, koşul bilgisi ve sayacın artım bilgisi verilmelidir. Döngü sayacı kullanılmıyorsa sadece döngüye devam edebilmek için gerekli olan koşul bilgisi verilmelidir.</p> <p>Döngü yapısının olduğu cümlede Sayaç kullanılmış ise; Sayaç'ı 1 artır, Sayaç<10 ise 3. Adım'a git şeklinde ifadeler kullanılır.</p> <ul style="list-style-type: none">Birkaç öğrenciden öğretmenin verdiği örneğe benzer şekilde gerçek hayatta tekrar eden işlemleri içeren birer örnek vermeleri istenir.

	<p>Daha sonra öğretmen günlük hayatta kullanılan bir aracın belli sayıda tekrar ettikten sonraki kullanma durumuna örnek verir. (Ör: E-posta hesabına erişmeye çalışırken şifreyi 3 kez yanlış girerek başka bir web sayfasına yönlendiriliriz .)</p> <p>Etkinliklerde,</p> <p>Deney grubunda Scratch programı kullanılır.</p> <ul style="list-style-type: none"> • Sayaç kullanarak ekrana 5 kez “Bilişim” yazdıran programın algoritması oluşturulur. • Sayı tahmin oyununun algoritması oluşturulur. <p>Oluşturulan algoritmaların düzgün çalışıp çalışmadığı incelenir.</p>
--	---

BÖLÜM 3

<p>Ölçme-Değerlendirme</p> <ul style="list-style-type: none"> • Bireysel öğrenme etkinliklerine yönelik Ölçme-Değerlendirme • Grupla öğrenme etkinliklerine yönelik Ölçme-Değerlendirme • Öğrenme güçlüğü olan öğrenciler ve ileri düzeyde öğrenme hızında olan öğrenciler için ek Ölçme-Değerlendirme etkinlikleri 	<ul style="list-style-type: none"> • Öğrencilere, etkinlikleri yaparken, döngü yapılarıyla ilgili doğru yanlış soruları sorularak verilen cevaplara göre değerlendirme yapılabilir. • Hatalı olarak verilen algoritmadaki hataları öğrencilerin düzeltilmesi istenir.
---	---

EK 2. ALGORİTMA GELİŞTİRMEYE İLİŞKİN ÖZ YETERLİK ALGISI ÖLÇEĞİ

ALGORİTMA ve STRATEJİ GELİŞTİRME ÖZYETERLİK ALGISI ÖLÇEĞİ

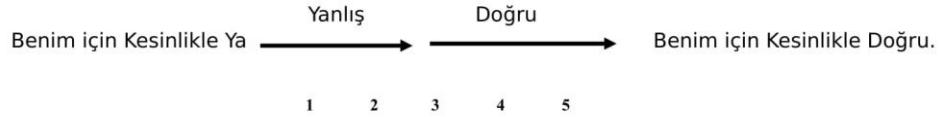
Değerli öğrenci,

Bu ölçek Bilişim Teknolojileri ve Yazılım dersi "Algoritma" konusundaki görevleri gerçekleştirebileceğinize yönelik kendinizle ilgili algılarınızı belirlemek amacıyla yapılan bilimsel bir araştırmanın yürütülmesi amacıyla hazırlanmıştır. Ölçekte yer alan sorulara verdiğiniz yanıtlar, kesinlikle **size not vermek** ya da sizi **eleştirmek** amacıyla **kullanılmayacaktır**. Bu soruların herkes için geçerli **doğru yanıtları bulunmamaktadır**. Bu nedenle lütfen aşağıda verilen tüm soruları dikkatle okuyarak cevabınızı, ifadenin karşısındaki seçeneklerden sizin için en uygun olanı işaretleyerek belirtiniz.

Öncelikle aşağıdaki soruları cevaplayınız.

Sınıfınız: () 6. Sınıf () 7. Sınıf	Cinsiyetiniz : () Kız () Erkek	Yaşınız: () 11 ve altı () 14 () 12 () 15 () 13
---	--	---

Soruları yanıtlamak için aşağıdaki ölçütleri kullanınız. Soruda geçen ifade sizin için **kesinlikle doğru ise (5)**'i; sizinle ilgili **kesinlikle yanlışsa (1)**'i işaretleyin. Eğer ifadenin size göre doğruluğu bunlardan farklı ise sizin için en uygun düzeyi gösteren (1)'le (5) arasındaki rakamı işaretleyin.



Soru No Öz Yeterlik Algısı

1. Algoritmasını çıkarmam için verilen problemi anlayabilirim.	(1) (2) (3) (4) (5)
2. Probleme uygun programlama algoritmasını geliştirebilirim.	(1) (2) (3) (4) (5)
3. Algoritma geliştirirken adımları düzgün planlayabilirim.	(1) (2) (3) (4) (5)
4. Algoritmamdaki hataları kendim belirleyebilirim.	(1) (2) (3) (4) (5)
5. Algoritmamda belirlediğim hataları düzeltebilirim.	(1) (2) (3) (4) (5)
6. Geliştirdiğim algoritmanın akış diyagramını çizebilirim.	(1) (2) (3) (4) (5)
7. Günlük hayatta karşılaştığım bir problemin çözümüne yönelik algoritma geliştirebilirim.	(1) (2) (3) (4) (5)
8. Bir başkasının daha sonra anlayabileceği bir program algoritması geliştirebilirim.	(1) (2) (3) (4) (5)
9. Verilen üç sayının ortalamasını bulan bir programın algoritmasını geliştirebilirim.	(1) (2) (3) (4) (5)
10. Algoritma geliştirirken programlamanın temel yapılarını(döngü, koşullu yapı vb.) kullanabilirim.	(1) (2) (3) (4) (5)
11. Programlamada gerekli olan değişken ve sabitleri algoritmamda kullanabilirim.	(1) (2) (3) (4) (5)
12. Verilen bir program algoritmasının nasıl çalışacağını yorumlayabilirim.	(1) (2) (3) (4) (5)
13. Başkasının geliştirdiği bir algoritmadaki hataları bulabilirim.	(1) (2) (3) (4) (5)
14. Verilen akış şemasında hata varsa belirleyebilirim	(1) (2) (3) (4) (5)
15. Verilen bir algoritmanın mantıksal hatalarını giderip algoritmayı çalışır hale getirebilirim.	(1) (2) (3) (4) (5)
16. Verilen sürede algoritmamı başarıyla hazırlayabilirim.	(1) (2) (3) (4) (5)

EK 3. GÜDÜLENME VE ÖĞRENME STRATEJİLERİ ÖLÇEĞİ

Soruları yanıtlamak için aşağıdaki ölçütleri kullanın. Soruda geçen ifade sizin için **kesinlikle doğru ise (7)**'yi; sizinle ilgili **kesinlikle yanlışsa (1)**'i işaretleyin. Eğer ifadenin size göre doğruluğu bunlardan farklı ise sizin için en uygun düzeyi gösteren (1)'le (7) arasındaki rakamı işaretleyin.

Soru No	GÜDÜLENME							
	Benim için Kesinlikle Yanlış.	1	2	3	4	5	6	7
1	Bunun gibi bir derste beni gerçekten çalışmaya zorlayacağına inandığım ders materyallerini tercih ederim, bu sayede yeni şeyler öğrenebilirim.	(1)	(2)	(3)	(4)	(5)	(6)	(7)
2	Ancak uygun bir şekilde çalışırsam bu dersin konularını öğrenebilirim.	(1)	(2)	(3)	(4)	(5)	(6)	(7)
3	Sınavdayken diğer öğrencilerden daha yetersiz olduğumu düşünürüm.	(1)	(2)	(3)	(4)	(5)	(6)	(7)
4	Bu derste öğrendiklerimi diğer derslerde de kullanabilirim.	(1)	(2)	(3)	(4)	(5)	(6)	(7)
5	Bu dersten çok iyi bir not alacağıma inanıyorum.	(1)	(2)	(3)	(4)	(5)	(6)	(7)
6	Bu derste okumam için verilecek en zor konuları bile anlayacağımdan eminim.	(1)	(2)	(3)	(4)	(5)	(6)	(7)
7	Benim için en tatmin edici şey sınıfta iyi bir not almaktır.	(1)	(2)	(3)	(4)	(5)	(6)	(7)
8	Sınavda soruları çözerken, sınav kağıdının diğer bölümlerindeki yanıtlayamayacağım soruları düşünürüm.	(1)	(2)	(3)	(4)	(5)	(6)	(7)
9	Eğer bu dersi öğrenemiyorsam bu benim kendi hatamdır.	(1)	(2)	(3)	(4)	(5)	(6)	(7)
10	Bu derste verilen kaynakları (kaynak materyalleri) öğrenmek benim için önemlidir.	(1)	(2)	(3)	(4)	(5)	(6)	(7)
11	Bu derste benim için en önemli şey, genel not ortalamamı yükseltmektir, yani bu derste asıl amacım iyi bir not almaktır.	(1)	(2)	(3)	(4)	(5)	(6)	(7)
12	Bu derste anlatılan temel kavramları anlayabileceğim konusunda kendime güveniyorum.	(1)	(2)	(3)	(4)	(5)	(6)	(7)
13	Eğer yapabilirsem, bu sınıftaki diğer öğrencilerin hepsinden daha yüksek not almak isterim.	(1)	(2)	(3)	(4)	(5)	(6)	(7)
14	Sınavdayken başarısızlığı ve bunun doğuracağı sonuçları düşünürüm.	(1)	(2)	(3)	(4)	(5)	(6)	(7)
15	Bu derste öğretmenin anlatacağı en zor konuyu bile anlayacağıma güveniyorum.	(1)	(2)	(3)	(4)	(5)	(6)	(7)
16	Bunun gibi bir derste, zor olsalar bile, bende merak uyandıran ders materyallerini tercih ederim.	(1)	(2)	(3)	(4)	(5)	(6)	(7)
17	Bu dersle ilgili konulara oldukça ilgi duyuyorum.	(1)	(2)	(3)	(4)	(5)	(6)	(7)
18	Yeterince çalışırsam dersi anlayabilirim.	(1)	(2)	(3)	(4)	(5)	(6)	(7)
19	Sınavdayken kendimi rahatsız ve morali bozuk hissedirim.	(1)	(2)	(3)	(4)	(5)	(6)	(7)
20	Bu dersteki ödevleri ve sınavları mükemmel yapabileceğim konusunda kendime güveniyorum.	(1)	(2)	(3)	(4)	(5)	(6)	(7)
21	Bu derste başarılı olmayı bekliyorum.	(1)	(2)	(3)	(4)	(5)	(6)	(7)
22	Bu derste benim için en tatmin edici şey içeriği mümkün olduğunca çok anlayabilmektir.	(1)	(2)	(3)	(4)	(5)	(6)	(7)
23	Bence bu derste kullanılan materyaller dersi öğrenmem için faydalıdır.	(1)	(2)	(3)	(4)	(5)	(6)	(7)
24	Eğer olanak tanırsa, iyi not almamı sağlamayacak olsa bile en iyi şekilde öğrenmemi sağlayacak ödevleri seçerim.	(1)	(2)	(3)	(4)	(5)	(6)	(7)
25	Dersi yeterince anlayamıyorsam, bu yeterince çalışmadığım içindir.	(1)	(2)	(3)	(4)	(5)	(6)	(7)
26	Bu dersin konularını seviyorum.	(1)	(2)	(3)	(4)	(5)	(6)	(7)
27	Bu dersin konularını öğrenmek benim için çok önemlidir.	(1)	(2)	(3)	(4)	(5)	(6)	(7)
28	Sınavdayken kalbimin hızla çarptığını hissedirim.	(1)	(2)	(3)	(4)	(5)	(6)	(7)
29	Eminim ki bu derste öğretilen tüm becerileri ustalıkla yapabilirim.	(1)	(2)	(3)	(4)	(5)	(6)	(7)
30	Sınıfta başarılı olmak isterim; çünkü yeteneğimi aileme, arkadaşlarıma, üstlerime ve diğerlerine göstermek benim için önemlidir.	(1)	(2)	(3)	(4)	(5)	(6)	(7)
31	Dersin zorluğunu, öğretmeni ve becerilerimi dikkate aldığımda, bence bu derste başarılı olurum.	(1)	(2)	(3)	(4)	(5)	(6)	(7)

EK 4. PROGRAMLAMA BAŞARI TESTİ






Değerli öğrenci,

Bu formda Bilişim Teknolojileri ve Yazılım dersinde kullanılmak üzere "Algoritma ve Strateji Geliştirme" konusu için bilimsel bir araştırmanın yürütülmesi amacıyla hazırlanmış başarı testi soruları bulunmaktadır. Testte yer alan sorulara verdiğiniz yanıtlar, kesinlikle size not vermek ya da sizi eleştirmek amacıyla kullanılmayacaktır.

Bu nedenle lütfen aşağıda verilen tüm soruları dikkatle okuyarak cevabınızı işaretleyiniz.

Cinsiyetiniz: () Kız () Erkek

Sınıf: () 6. Sınıf () 7. Sınıf

<p>1. Bir problemi çözebilmek için gerekli olan sıralı mantıksal adımların tümüne ne denir? a) Algoritma b) Akış Şeması c) Program d) Yazılım</p>	<p>6. Algoritmanın görsel olarak simge ya da sembollerle ifade edilmiş şekline ne denir? a) Programlama Dili b) Akış Şeması c) Program d) Yazılım</p>
<p>2. Programlama aşamaları aşağıda karışık olarak verilmiştir. Doğru sıralama hangisidir? I. Algoritma yazma-Akış şeması çizme II. Problemi tanımlama III. Programın kodlanması IV. Programı bilgisayarda çalıştırıp deneme V. Çözüm yöntemi belirleme a) I-V-III-II-IV b) II-V-I-III-IV c) II-V-III-IV-I d) I-II-III-IV-V</p>	<p>7. Aşağıda algoritması verilen programın hangi adımında koşul yapısı kullanılmıştır? Adım1 = Başla Adım2 = Sayaç=0 Adım3 = Sayaç'ı 1 artır Adım4 = Ekranı "AYŞE" yaz Adım5 = Eğer sayaç<6 ise Adım 3'e git Adım6 = Bitir a) Adım2 b) Adım3 c) Adım4 d) Adım5</p>
<p>3. Aşağıda algoritması verilen programda ekran çıktısı nedir? Adım 1 = Başla Adım 2 = a=20 Adım 3 = a + 12 'yi ekrana yaz Adım 4 = Bitir A)8 B)12 C)20 D)32</p>	<p>8. Aşağıdaki verilen algoritma tablosunda kuş karakterini () domuz karakterine () ulaştırabilmek için soru işareti(?) ile gösterilen 3. Adıma hangisi gelmelidir?  a)  b)  c)  d) </p>

<p>13. Aşağıda algoritması verilen program çalıştığında klavyeden sırasıyla "İlayda", "Kaya" yazısı yazılırsa ekran çıktısı hangisi olur?</p> <p>Adım 1 = Başla Adım 2 = "Adınızı giriniz!" yaz Adım 3 = Ad'ı oku Adım 4 = "Soyadınızı giriniz!" yaz Adım 5 = Soyad'ı oku Adım 6 = Ekran Ad ve Soyad'ı yaz Adım 7 = Bitir</p> <p>a) İlayda b) Kaya c) Hoş geldin d) İlayda Kaya</p>	<p>17. Aşağıda algoritması verilen programın çalıştırılması durumunda klavyeden "5" rakamı girilirse çıktı ne olur?</p> <p>Adım 1 = Başla Adım 2 = Ekran "Bir sayı giriniz" yaz Adım 2 = Sayı'yı oku Adım 3 = Eğer (Sayı < 5) ise "Sayı 5'den küçüktür" yaz Değilse ise "Sayı 5'den büyüktür" yaz Adım 4 = Bitir</p> <p>A) Ekran "Sayı 5'dir" yazısı yazar B) Ekran "Sayı 5'den küçüktür" yazısı yazar C) Ekran "Sayı 5'den büyüktür" yazısı yazar D) Program çalışmaz</p>
<p>Adım 1 = Başla Adım 2 = Ekran "5 rakamını giriniz" yaz Adım 3 = Girilen Sayı'yı oku Adım 4 = "Eğer Sayı=5 ise "Doğru girdiniz" yaz Adım 5'e git Değilse "Yanlış girdiniz" yaz Adım 2'ye git Adım 5 = Bitir</p> <p>14. Yukarıda algoritması verilen programla ilgili olarak aşağıdakilerden hangisi söylenemez?</p> <p>a) Program çalıştığında kullanıcıdan 5 rakamını yazması beklenir b) Kullanıcı klavyeden 5 rakamını girdiğinde "Doğru girdiniz" mesajı çıkar ve program durur. c) Kullanıcı 6 rakamını girdiğinde "Yanlış girdiniz" mesajı çıkar ve program durur. d) Klavyeden 5 rakamı girilene kadar program çalışmaya devam eder.</p>	<p>18. Adım 1 = Başla Adım 2 = Ekran "Adınızı giriniz" yaz Adım 3 = Ad'ı oku Adım 4 = Ekran "Soyadınızı giriniz" yaz Adım 5 = Soyad'ı oku Adım 6 = Ekran "Unvanınızı giriniz" yaz Adım 7 = Unvan'ı oku Adım 8 = Mesaj = "Merhaba" Adım 9 = Unvan, Soyad, Mesaj'ı ekrana sırasıyla yaz Adım 10 = Bitir</p> <p>Yukarıda algoritması verilen programa klavyeden sırasıyla "Aslı", "Kaya", "Miss" ifa deleri girilmiştir. Programın ekran çıktısı aşağıdakilerden hangisidir?</p> <p>a) Merhaba Miss Aslı Kaya b) Miss Aslı Kaya Merhaba c) Miss Kaya Merhaba d) Merhaba Aslı Kaya</p>

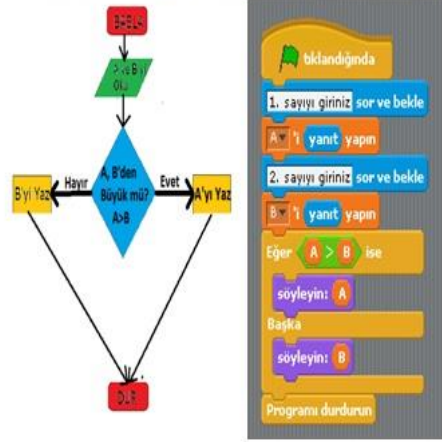
<p>19. Adım 1 = Başla Adım 2 = "Birinci sayıyı giriniz" yaz Adım 3 = sayı1'i oku Adım 4 = "İkinci sayıyı giriniz" yaz Adım 5 = sayı2'yi oku Adım 6 = Toplam = sayı1 + sayı2 Adım 7 = Sonuç = Toplam / 2 Adım 8 = Ekran Sonuç yaz Adım 9 = Bitir</p> <p>Yukarıda algoritması verilen programın amacını nedir?</p> <p>a) Klavyeden girilen iki sayıyı ekrana yazar b) Klavyeden girilen iki sayının ortalamasını ekrana yazar c) Klavyeden girilen iki sayının toplamını ekrana yazar d) Klavyeden girilen iki sayıdan büyük olanı ekrana yazar</p>	<p>22. Aşağıda algoritması verilen programda Adım 8 algoritma dan çıkarılırsa programın çalışması nasıl değişir?</p> <p>Adım 1 = Başla Adım 2 = Sayaç=0 Adım 3 = Sayaç'ı 1 artır Adım 4 = Ekran "Sıcaklık değeri giriniz" yaz Adım 5 = Sıcaklık'ı oku Adım 6 = Eğer Sıcaklık < 0 ise Ekran katı yaz. Adım 7 = Eğer Sıcaklık > 0 ve Sıcaklık < 100 ise Ekran sıvı yaz Adım 8 = Eğer Sayaç < 5 ise Adım 3'e git Adım 9 = Bitir.</p> <p>A) Döngü 5 kez tekrar etmez, yalnızca 1 kez çalışır ve durur B) "Sıcaklık değeri giriniz" mesajından sonra program durur C) Sürekli tekrar eden sonsuz bir döngü olur D) Programda bir değişiklik olmaz</p>
<p>Adım 1 = Başla Adım 2 = Mesaj = "Başarılı" Adım 3 = Sayaç = 1 Adım 4 = Ekran "Notu giriniz" yaz Adım 5 = Not'u oku Adım 6 = Eğer Sayaç > 0 ise Adım 4'e git Adım 7 = Eğer Not > 50 ise Ekran Mesaj'ı yaz Adım 8 = Bitir</p> <p>20. Yukarıda algoritması verilen program çalıştığı zaman mantık hatası oluşmakta ve sürekli tekrar eden sonsuz bir döngü oluşmaktadır. Programı çalışır hale getirmek için aşağıdakilerden hangisinin yapılması gerekmektedir?</p> <p>a) Adım 1 ile Adım 2'yi yer değiştirmek b) Adım 6'da "Adım 4'e git ifadesini "Adım 1'e git" olarak değiştirmek c) Adım 6 ile Adım 7'yi yer değiştirmek d) Adım 2'yi algoritma dan çıkarmak</p>	<p>Adım 1 = Başla Adım 2 = Ekran "Bir sayı giriniz" yaz Adım 3 = sayı'yı oku Adım 4 = Eğer sayı > 0 ise pozitif yaz Adım 5 = Eğer sayı < 0 ise negatif yaz Adım 6 = Eğer sayı = 0 ise sıfır yaz Adım 7 = Bitir</p> <p>23. Yukarıda algoritması verilen program ile ilgili olarak aşağıdakilerden hangisi söylenemez?</p> <p>A) Klavyeden girilen sayının çift, tek olduğunu ekrana yazar. B) Klavyeden girilen sayının pozitif, negatif, sıfır olduğunu ekrana yazar. C) Klavyeden "5" rakamı girildiğinde ekrana "pozitif" mesajı çıkar D) Program döngü olma dışı için sadece bir kez çalışır ve durur</p>

21. Aşağıda akış diyagramı verilen programın amacı nedir?



- a) Klavyeden girilen iki sayıyı ekrana yazdırır
- b) Klavyeden girilen iki sayıdan küçük olanı ekrana yazdırır
- c) Klavyeden girilen iki sayıdan büyük olanı ekrana yazdırır
- d) Klavyeden girilen iki sayının toplamını ekrana yazdırır

24. Aşağıda akış diyagramı verilen programın amacı nedir?



- a) Klavyeden girilen iki sayıyı ekrana yazdırır
- b) Klavyeden girilen iki sayıdan küçük olanı ekrana yazdırır
- c) Klavyeden girilen iki sayıdan büyük olanı ekrana yazdırır
- d) Klavyeden girilen bir sayının pozitif, negatif olduğunu ekrana yazdırır

25. Ekrana 3 kez "Tebrikler!" yazdıran program algoritması aşağıdakilerden hangisidir.

- A) A1=Başla
A2= Ekrana "Tebrikler!" yaz
A3= Bitir
- B) A1=Başla
A2= Mesaj= "Tebrikler!"
A3=Ekrana Mesaj'ı yaz
A4= Bitir
- C) A1=Başla
A2= Sayaç=0
A3=Sayaç'ı 1 artır
A4=Ekrana "Tebrikler!" yaz
A5=Eğer sayaç<3 ise A3'e git
A6=Bitir
- D) A1=Başla
A2= Ekrana "Notumuzu giriniz" yaz
A3=Not'u oku
A4=Eğer Not>50 ise A4'e git
A4=Ekrana "Tebrikler!" yaz
A5= Bitir

EK 5. ÖZ-YETERLİK ALGISI ÖLÇEĞİ İÇİN UZMAN GÖRÜŞÜ ÖRNEĞİ


ALGORİTMA ve STRATEJİ GELİŞTİRME ÖZYETERLİK ALGISI ÖLÇEĞİ UZMAN GÖRÜŞ FORMU

Değerli Uzman:

Bu formda ortaokul öğrencilerinin "Algoritma ve Strateji Geliştirme" ile ilgili öz yeterlik algılarını belirlemek için geliştirilen ölçek maddeleri bulunmaktadır. Sizden ricam formdaki maddelerin "Algoritma ve Strateji Geliştirme" konusunu ölçmeye uygunluğu konusunda görüş bildirmenizdir. Vakit ayırdığınız için teşekkür ederiz.

Lütfen maddeler uygun ise maddelerin yanında yer alan uygun kutucuğunu, değilse uygun değil kutucuğunu işaretleyiniz. Maddelerin uygun olmaması durumunda en sağda yer alan boşluğa madde ile ilgili görüşlerinizi bildiriniz.

Madde	Uygun	Uygun değil	Değerlendirme
1. Algoritmasını çıkarmam için verilen problemi anlayabilirim.		✓	Çok iyi değil. Daha fazla çalışması
2. Probleme uygun programlama algoritmasını çizebilirim.	✓		
3. Algoritma çizerken akışı düzgün planlayabilirim.	✓		
4. Algoritmadaki hataları kendim belirleyebilirim.	✓		
5. Algoritmamda belirlediğim hataları düzeltebilirim.	✓		
6. Tasarladığım algoritmanın akış diyagramını çizebilirim.	✓		
7. Günlük hayatta karşılaştığım bir problemin çözümüne yönelik algoritma geliştirebilirim.	✓		
8. Bir başkasının daha sonra anlayabileceği ve üzerinde eklemeler yapabileceği bir program algoritması geliştirebilirim.	✓		
9. Verilen üç sayının ortalamasını bulan bir programın algoritmasını geliştirebilirim.	✓		
10. Programlamanın temel yapılarını (döngü, koşullu yapı vb.) algoritma tasarlarken kullanabilirim.			Algoritma tasarlarken programlamanın temel yapılarını kullanabilirim
11. Programlamada gereken ^{gerektiği} değişken ve sabitleri algoritmamda kullanabilirim.			
12. Verilen bir program algoritmasının nasıl çalışacağını yorumlayabilirim.	✓		
13. Başkasının hazırladığı bir algoritmadaki hataları bulabilirim.	✓		
14. Akış şemasında hata varsa belirleyebilirim	✓		
15. Geliştirdiğim bir algoritmanın mantıksal hatalarını giderip algoritmayı çalışır hale getirebilirim.	✓		
16. Algoritma geliştirmek için verilen sınırlı sürede başarıyla algoritmamı hazırlayabilirim.	✓		

Uzmanın Adı Soyadı:  *Barış Yeterde*

Barış Yeterde "tasarladığım algoritma" başarıyla "geliştirdiğim algoritma" yapıyor. Başarılar dilerim.

EK 6. BAŞARI TESTİ UZMAN GÖRÜŞÜ ÖRNEĞİ

<p>Adım1 = Başla Adım2 = Trafik lambasına bak Adım3 = Lamba Yeşil yanıyor mı? (Yeşil yanarsa 6. Adıma git) Adım4 = Adım5 = Lamba Kırmızı yanıyor mu? (Kırmızı yanarsa 1. Adıma git) Adım6 = Xine Adım7 = Bitir</p> <p>11. Yukarıda bir otomobillerin trafik ışıklarında giry algoritmasını vermiştir. Buna göre boş bırakılan yere girecek girecekleri adı ve hangisidir?</p> <p>a) Lamba Sarı yanıyor mı? Sarı yanarsa 6. Adıma git b) Lamba Sarı yanıyor mı? Sarı yanarsa 2. Adıma git c) Lamba Sarı yanıyor mı? Sarı yanarsa 7. Adıma git d) Lamba Sarı yanıyor mı? Sarı yanarsa 3. Adıma git Cevap: B</p>	Uygun	Konusu Uygun	Uygun Deği								
<p>12. Aşağıda algoritması verilen program kapsamında klavyeden yazıyla "Haydi", "Kaya" yazan ifadeleri ekran çıktısı hangisi olur?</p> <p>Adım 1 = Başla Adım 2 = "Adınızı giriniz" yaz Adım 3 = Ad'ı oku Adım 4 = "Soyadınızı giriniz" yaz Adım 5 = Soyad'ı oku Adım 6 = Ekrana Ad ve Soyad'ı yaz Adım 7 = Bitir</p> <p>Ekran Çıktısı a) Haydi Kaya c) Haydi Kaya d) Haydi Kaya</p>	Uygun	Konusu Uygun	Uygun Deği								
<p>13. Aşağıda algoritması verilen programda ekran çıktısı nedir?</p> <p>Adım 1 = Başla Adım 2 = x=20 y=12 Adım 3 = x=y-y Adım 4 = x'i ekrana yaz Adım 5 = Bitir</p> <p>AK B) 12 C) 20 D) 12</p> <p>Cevap: D</p>	Uygun	Konusu Uygun	Uygun Deği								
<p>14. Aşağıdaki program algoritmasına göre ilgili ifadelerin doğru tablodaki gireceği boşlukları ekran çıktısı tabloya yazınız.</p> <p>Adım 1 = Başla Adım 2 = Ekran "De sayı giriniz" yaz Adım 3 = Sayı'yı oku Adım 4 = Eğer Sayı > 10 ise, Sonuç = Sayı - 3 Adım 5 = Eğer Sayı < 10 ise, Sonuç = Sayı + 10 Adım 6 = Eğer Sayı < 10 ise, Sonuç = Sayı Adım 7 = Ekran Sonuç'ı yaz Adım 8 = Bitir</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>Sayı</th> <th>Ekran Çıktısı</th> </tr> </thead> <tbody> <tr> <td>10</td> <td></td> </tr> <tr> <td>6</td> <td></td> </tr> <tr> <td>20</td> <td></td> </tr> </tbody> </table>	Sayı	Ekran Çıktısı	10		6		20		Uygun	Konusu Uygun	Uygun Deği
Sayı	Ekran Çıktısı										
10											
6											
20											
<p>15. Adım 1 = Başla Adım2 = Ekran " Adınızı giriniz" yaz Adım3 = Ad'ı oku Adım4 = Ekran " Soyadınızı giriniz" yaz Adım 5 = Soyad'ı oku Adım6 = Ekran "Ünvanınızı giriniz" yaz Adım7 = Ünvan'ı oku Adım 8 = Menü "Merhaba" Adım 9 = Ünav, Soyad, Menü'ü ekrana yazıyla yaz Adım 10 = Bitir</p> <p>Yukarıda algoritması verilen programa klavyeden yazıyla "Ali", "Kaya", "Mev" ifadeleri girilmiştir. Programın ekran çıktısı aşağıdaki listeden hangisidir?</p> <p>a) Merhaba Ali Kaya Mev b) Ali Kaya Mev c) Ali Kaya Mev d) Merhaba Ali Kaya</p>	Uygun	Konusu Uygun	Uygun Deği								
<p>16. Adım 1 = Başla Adım2 = Ekran sayısını giriniz Adım3 = Ekran sayısını oku Adım4 = Çıkartma sayısına giriniz Adım5 = Çıkartma sayısını oku Adım 6 = Ekran (sayı - çıkartma) x 3 ve (çıkartma / sayı) - çıkartma x 2 Adım7 = Toplamı ekrana yaz Adım8 = Ekran Toplam'ı yaz Adım9 = Bitir</p> <p>Yukarıda algoritması verilen programa klavyeden sırasıyla 2 ve 5 rakamları girilmiştir. Programın ekran çıktısı aşağıdaki listeden hangisidir?</p> <p>a) 8 b) 16 c) 17 d) 19</p> <p>Cevap: B</p>	Uygun	Konusu Uygun	Uygun Deği								

EK 7. RUBRİK DEĞERLENDİRME FORMU

Rubrik (Öğretmen Değerlendirmesi)

(Ders esnasında yapılan etkinliğe ilişkin)

MADDELER	1	2	3	4	5
1- Verilen problem örneğine çözüm bulma					
2- Algoritmanın mantıksal yapısını oluşturma.					
3- Algoritma adımlarını doğru sırada oluşturma.					
4- Algoritma cümlelerini eksiksiz oluşturma.					
5- Algoritmanın düzgün çalışıp çalışmadığını inceleme					

5 puan	Öğrenci etkinliği, doğru, sade başarılı ve net bir şekilde tamamlamıştır
4 puan	Öğrenci etkinliği tamamlamıştır ancak daha kısa işlem sayısı ile çözümler mümkündür. (Öğrenci çözümünde çözümü uzatan, daha çok zaman alan, sadeleştirilebilecek kısımlar bulunmaktadır)
3 puan	Öğrenci etkinliği tek hata ile tamamlamıştır.
2 puan	Öğrenci etkinliği iki hata ile tamamlamıştır.
1 puan	Öğrenci etkinliği tamamlamıştır ikiden fazla hata/eksik bulunmaktadır.
0 puan	Öğrenci etkinliği tamamlayamamıştır.

ÖZGEÇMİŞ VE İLETİŞİM BİLGİSİ

Funda BAKIRCI, 1987 yılında Ankara'da doğdu. İlköğretimini ve liseyi Ankarad'da tamamladı. Samsun Ondokuz Mayıs Üniversitesi Eğitim Fakültesi'nde Bilgisayar ve Öğretim Teknolojileri Eğitimi bölümünden 2009 yılında mezun oldu. 2009 yılında Milli Eğitim Bakanlığı'na Bilişim Teknolojileri öğretmeni olarak atandı. 2013 yılında Sakarya Üniversitesi Eğitim Bilimleri Enstitüsü Bilgisayar ve Öğretim Teknolojileri Eğitimi Bölümü Anabilim Dalı'nda yüksek lisans eğitimine başladı. Yabancı dili İngilizcedir.

E-posta: fundabakirci0619@gmail.com