



T.C.
EGE ÜNİVERSİTESİ
Eğitim Bilimleri Enstitüsü



**ARDUINO KULLANIMININ LİSE
ÖĞRENCİLERİNİN AKADEMİK BAŞARILARINA
VE PROGRAMLAMAYA YÖNELİK
TUTUMLARINA OLAN ETKİSİ**

Yüksek Lisans Tezi

Gönül ALTAY

Bilgisayar ve Öğretim Teknolojileri Eğitimi Anabilim Dalı

İzmir
2019

İZMİR

2019

T.C.

EGE ÜNİVERSİTESİ

Eğitim Bilimleri Enstitüsü

**ARDUINO KULLANIMININ LİSE ÖĞRENCİLERİNİN
AKADEMİK BAŞARILARINA VE PROGRAMLAMAYA
YÖNELİK TUTUMLARINA OLAN ETKİSİ**

Gönül ALTAY

Yüksek Lisans Programı

Bilgisayar Ve Öğretim Teknolojileri Eğitimi Ana Bilim Dalı

Danışman:

Doç. Dr. Tarık KIŞLA

Gönül ALTAY tarafından yüksek lisans tezi olarak sunulan “Arduino kullanımının lise öğrencilerinin akademik başarılarına ve programlamaya yönelik tutumlarına olan etkisi” başlıklı bu çalışma E.Ü. Lisansüstü Eğitim ve Öğretim Yönetmeliği ile EÜ Fen Bilimleri Enstitüsü Eğitim ve Öğretim Yönergesi'nin ilgili hükümleri uyarınca tarafımızdan değerlendirilerek savunmaya değer bulunmuş ve 05.08.2019 tarihinde yapılan tez savunma sınavında aday oybirliği ile başarılı bulunmuştur.

Jüri Üyeleri:

Jüri Başkanı : Prof. Dr. Mustafa Murat İnceoğlu

Raportör Üye : Doç. Dr. Tarık Kışla

Üye : Doç. Dr. Yasin Özarlan

İmza

.....
.....
.....

EGE ÜNİVERSİTESİ EĞİTİM BİLİMLERİ ENSTİTÜSÜ
ETİKKURALLARA UYGUNLUK BEYANI

EÜ Lisansüstü Eğitim ve Öğretim Yönetmeliğinin ilgili hükümleri uyarınca Yüksek Lisans Tezi olarak sunduğum “Arduino kullanımının lise öğrencilerinin akademik başarılarına ve programlamaya yönelik tutumlarına olan etkisi” başlıklı bu tezin kendi çalışmam olduğunu, sunduğum tüm sonuç, doküman, bilgi ve belgeleri bizzat ve bu tez çalışması kapsamında elde ettiğimi, bu tez çalışmasıyla elde edilmeyen bütün bilgi ve yorumlara atıf yaptığımı ve bunları kaynaklar listesinde usulüne uygun olarak verdiğimi, tez çalışması ve yazımı sırasında patent ve telif haklarını ihlal edici bir davranışımın olmadığını, bu tezin herhangi bir bölümünü bu üniversite veya diğer bir üniversite başka bir tez çalışması içinde sunmadığımı, bu tezin planlanmasında yazımına kadar bütün safhalarda bilimsel etik kurallarına uygun olarak davrandığımı ve aksinin ortaya çıkması durumunda her türlü yasal sonucu kabul edeceğimi beyan ederim.

Gönül ALTAY

ÖZET
ARDUINO KULLANIMININ LİSE ÖĞRENCİLERİNİN AKADEMİK
BAŞARILARINA VE PROGRAMLAMAYA YÖNELİK
TUTUMLARINA OLAN ETKİSİ

ALTAY, Gönül

Yüksek Lisans Tezi, Bilgisayar ve Öğretim Teknolojileri Eğitimi Anabilim Dalı

Tez Danışmanı: Doç. Dr. Tarık KIŞLA

Kasım 2019, 127 sayfa

Teknolojik gelişmelerin hızlı bir şekilde ilerlediği günümüz dünyasında, günlük yaşamda karşılaşılan problemler için yaratıcı çözümler üretebilen bireylere ihtiyaç duyulmaktadır. Bu durum öğrencilere, problemleri basamaklarına bölerek hızlı ve kolay çözüm sunan algoritmik düşünme eğitiminin verilmesinin önemini ortaya çıkarmıştır. Bu nedenle de okullarda küçük yaşlardan itibaren algoritmik düşünme temeline dayanan programlama eğitimi verilmektedir.

Programlama eğitiminde kullanılacak materyallerin ve öğretim yöntemlerinin doğru bir şekilde belirlenmesi, dersin hedeflerine ulaşmada başarılı olabilmek açısından önem arz etmektedir. Ayrıca bu eğitimlerde kullanılan yöntem, teknik ve materyallerin, öğrencilerin akademik başarıları üzerinde yaptığı etki de eğitimcilerin önem verdiği bir konudur. Bu tezde de lise öğrencilerinin programlama temellerini öğrenirken karşılaştıkları zorlukların giderilmesine yardımcı olmak amacıyla 8 haftalık Arduino kullanımına dayalı bir öğretim gerçekleştirilmiş ve tasarlanan bu dersin öğrencilerin akademik başarılarına ve programlamaya yönelik tutumlarına olan etkisi incelenmiştir. Araştırmada öncelikle, programlama konusunun önemi ve karşılaşılan problemler açıklanmış ve mesleki ve teknik eğitim alan öğrencilerin, Milli Eğitim Bakanlığı müfredatında belirlenen hedeflere ulaşması yolunda Arduino ile robotik dersler tasarlanmıştır. Veri toplama araçları olarak da akademik başarı testi ve programlamaya yönelik tutum ölçeği geliştirilmiştir. Bu araçlar deney ve kontrol gruplarından oluşan çalışma gruplarına deneysel işlem öncesinde ve sonrasında uygulanmış ve veriler toplanarak analiz edilmiştir. Verilerin analizinde de tekrarlı ANOVA yöntemi kullanılmıştır.

Çalışma sonucunda elde edilen bulgular, robotik dersini alan deney grubu öğrencileri ile dersi almayan kontrol grubu öğrencileri arasındaki akademik başarıda olumlu bir artış olduğunu göstermiştir. Ayrıca dört hafta robotik eğitime ara verildikten sonra öğrencilere uygulanan izleme testi sonuçlarına göre de, gerçekleştirilen öğretimin kalıcı olduğu söylenebilir. İki grup arasındaki programlamaya yönelik tutumda ise anlamlı bir fark bulunamamıştır.

Anahtar sözcükler: Arduino, programlama tutum ölçeği, akademik başarı testi, robotik programlama dersi



EXTENDED ABSTRACT
THE EFFECTS OF ARDUINO USE ON ACADEMIC
ACHIEVEMENTS OF AND ON ATTITUDES TOWARDS
COMPUTER PROGRAMMING OF HIGH SCHOOL STUDENTS

ALTAY, Gönül

Master Program, Computer and Instructional Technologies Education.

Advisor: Assoc. Dr. Tarık KIŞLA

November 2019, 127 pages

In today's world, where technological developments are advancing rapidly, it is needed to individuals who can produce creative solutions for the problems encountered in daily life. This situation revealed the importance of providing algorithmic thinking education which provides quick and easy solution by dividing the problems into steps. For this reason, from an early age programming education based on algorithmic thinking is taught in the schools.

Correct determination of the materials and teaching methods to be used in programming education is important in order to be successful in achieving the goals of the course. In addition, impact of the methods, techniques and materials used in these trainings on the academic achievement of the students is a important issue for educators. In this thesis, in order to help the high school students to overcome the difficulties encountered while learning the basics of programming, an 8-week Arduino based instruction was conducted and the effect of this course on students' academic achievement and programming attitudes was examined. Firstly, the importance of programming and the problems encountered were explained and robotic courses were designed with Arduino in order to achieve the goals determined in the curriculum of the Ministry of National Education. Academic achievement test and attitude scale towards programming were developed as data collection tools. These instruments were applied to the study groups consisting of experimental and control groups before and after the experimental procedure and the data were collected and analyzed. Repeated ANOVA method was used for data analysis too.

The results of the study showed that there was a positive increase in academic achievement between experimental group students taking robotics

course and control group students who did not take the course. In addition, according to the results of the permanency test applied to students after four weeks of robotic education is interrupted, it can be said that the teaching performed is permanent. There was no significant difference in as for the attitude towards programming between the two groups.

Keywords: Arduino, programming attitude scale, academic achievement test, robotic programming course



ÖNSÖZ

Programlama eğitiminde karşılaşılan zorluklar, programlama derslerindeki akademik başarının yeterince yüksek olmaması ve öğrencilerin programlamaya karşı olan olumsuz tutumları, araştırmacıları öğretimde yeni bir arayışa yönlendirmektedir. Bu noktada, teknolojik gelişmeler robotik programlamanın önemli olduğunu ve robotiğe olan talebin arttığını göstermiştir. Robotik programlamaya artan talep dikkate alındığında ise erken yaşlarda programlama eğitiminin yoğun bir şekilde verildiği mesleki eğitim kurumlarında öğrenim gören öğrenciler için robotik ders tasarımı gerçekleştirilmesi önem kazanmıştır.

Çalışmada kullanılacak materyaller belirlendikten sonra temini için gerekli araştırmalar yapılmış ve Ege Üniversitesi tarafından desteklenen 17-EĞF-002 nolu Bilimsel Araştırma Projesi gerçekleştirilmiştir. Çalışmanın tüm materyalleri de proje desteğiyle alınmıştır.

İZMİR

.../.../2019

Gönül ALTAY

TEŐEKKÜR

Bu alıőmada daima alıőmalarıma destek olup yardımlarını esirgemeyen, yürüdüėüm yolda her zaman bana ışık tutan deėerli hocam ve danıőmanım Do. Dr. Tarık KIŐLA' ya abalarından, gösterdiėi ilgi ve hoőgörüsünden dolayı saygılarımı ve teőekkürlerimi sunarım. alıőmanın materyallerinin alınması için gereken desteėi saėlayan Ege Üniversitesi' ne katkısından ötürü teőekkür ederim. Ayrıca manevi desteklerini her an yanımda hissettiėim eőőiz aileme de minnettarım.

... / ... / 2019

İmzası

Gönül ALTAY

ÖZGEÇMİŞ

E-Mail: gonulaltayy@gmail.com

EDUCATION:

Undergraduate: Süleyman Demirel University / Department of Electronics and Computer Education/ Computer Systems Education (2004 - 2008)

Graduate: Ege University / Graduate School of Natural and Applied Sciences / Computer and Instructional Technologies Education (2016 -2019)

Undergraduate: Ege University / Faculty of Engineering / Engineering Completion Program / Computer Engineering (2018 -)

WORK EXPERIENCE:

Simav Vocational and Technical Anatolian Vocational High School (2008 - 2009)

Turgutlu Vocational and Technical Anatolian High School (2009 - 2014)

B Class Job Security Expert (2013 -)

Manisa Yunusemre Vocational and Technical Anatolian High School (2014 - 2016)

Manisa Polinas Vocational and Technical Anatolian High School (2016 -)

AREA INFORMATION

Microsoft Office

Web Programming (ASP, ASP.Net, PHP, Javascript, HTML, CSS, JQuery)

Adobe Master Collection

Object-Oriented Programming (C ++, C #, Java)

Database Management (MS Access, SQL Management)

Electronic Applications (Arduino, Pic Programming, Assembly)

TRAINING PROGRAMS:

Web Design by Adobe Dreamweaver,Trainer (2009)

ISKUR Computer Management Course, Trainer (2009)

ISKUR Computer Maintenance and Repair Course, Trainer (2010)

2011450572, Basic Computer Using Course, Trainer, (2011)

Computer Management Course, Trainer (2015)

2015450426, Job Health and Security Training of Employees, Trainer, (2015)

2015450452, Job Health and Security Training of Employees, Trainer, (2015)

2015450508, Job Health and Security Training of Employees, Trainer, (2015)

Job Security Training in Heavy and Dangerous Jobs / Vocational Training Programs for all Factory, Trainer (2015-)

2016450416, Job Health and Security Training of Employees, Trainer, (2016)

2016450452, Job Health and Security Training of Employees, Trainer, (2016)

2017450698, Job Health and Security Training of Employees, Trainer, (2017)

CERTIFICATES AND TRAINING PROGRAMS RECEIVED:

2009450601, Candidate Basic Training Course / Achievement Certificate (2009)

2009450617, Preparatory Training Course For Candidate Staff / Achievement Certificate (2009)

Training Course on Visual Basic 6.0 / Achievement Certificate (2009)

2011450641, Seminar on First Aid and First Response in Schools / Participation Certificate (2011)

2013450493, Fatih Project - Seminar on Informative and Secure Use of Information Technologies and Internet / Participation Certificate (2013)

B Class Job Security Expertise Certificate (2013)

What is the difference? / Youthpass Certificate of Youth Education (2014)

2015450254, FATİH Project - Technology Use Course in Education / Achievement Certificate(2015)

2016451056, E-Exam Applications Course / Achievement Certificate (2016)

2016450910, Seminar on Special Education Services / Participation Certificate (2016)

Workshop on Supporting English Language Teaching of Teachers / General English and Vocational English Education / the UK (2017)

Codeweek 2018 / Web Programming, Coding with Arduinio, Coding with ASP.NET, Coding with Action Script Activities / Participation Certificates in The Four Events and Certificate of Excellent in Coding Literacy (2018)

Bilgekunduz International Informatics and Cognitive Thinking Event / Participation Certificate (2018)

Codeweek 2019 / We are Programmer, Coding for the future, I am a robot / Participation Certificates in The Three Events (2019)

Training on Increasing Vocational Qualifications for Workshop and Laboratory Teachers / Germany-Berlin Technical University / DAI-Labor / Machine Learning, Telecommunications, Internet of Things, Cybersecurity and Smart Energy Systems. (2019)

TUBITAK 2237-A-Program for Supporting Scientific Education Activities
”Analytical Nature - Clustering and Ordinance Techniques Program and Project
No: 1129B371801420 / Participation Certificate (2019)

TOBB-EYUDER / Vocational and Technical Training Workshop / Participation
Certificate (2019)

Cyber Security Training, Certificate of Completion / Cisco Networking Academy
(2019)

WiFi Wireless Networks Training, Certificate of Completion / Cisco Networking
Academy (2019)

CCNA Routing and Switching: Introduction to Networks Certificate / Cisco
Networking Academy (2019)

Microsoft Community Influencer (2019)

Microsoft Community Contributor (2019)

Certified Microsoft Innovative Educator (2019)

Technical Assistance for Improving Institutional Capacity in Employment,
Education and Social Policies Sectoral Operational Program Implementation /
Participation Certificate (2019)

Problems And Needs For Vocational And Technical Education For TR33 Region
(Manisa, Uşak, Afyonkarahisar, Kütahya), Zafer Development Agency, Ministry
of Industry and Technology (2019)

PROJECTS

They were in Here, Manisa (2011)

Innovation Approach in Education and Training / Everywhere is the School for Us
/ The First Degree of Provincial (2015)

Erasmus + KA102 Vocational Learner and Personnel Mobility / Interactive
Marketing Techniques / Student Internship for 2-Week in the UK and in
Germany/ Project Consultancy (2016)

Values in Education / E-Twinning Project (2016)

4006 Tubitak Science Fair / 20 Projects in Information Technology / Projects
Coordinator and Consultancy (2017)

Arduino for Learning to Learn / E-Twinning Project (2017)

How to Make School More Attractive and Creative / E-Twinning Project (2019)

The Scientific Research Project Supported by Ege University (BAP, 07-EĞF-002)
/ The Effects of Arduino Use on Academic Achievements of and on Attitudes

Towards Computer Programming of High School Students / Researcher (2017-2019)

ACADEMIC STUDIES

Investigation of Cyber Bullying Profiles Vocational High School Students' (Original name of study: "Meslek Lisesi Öğrencilerinin Siber Zorbalık Profillerinin İncelenmesi") / International Conference on New Trends in Education/ Ege University / Presentation (2018)

Technology Integration in Education/ The 11th International Computer & Instructional Technologies Symposium (ICITS) / İnönü University / Presentation (2017)

Evaluation of Creative Problem Solving Process: Adaptation of the Scale to Turkish / The 11th International Computer & Instructional Technologies Symposium (ICITS) / İnönü University / Presentation (2017)

Evaluation of Creative Problem Solving Process: Adaptation of the Scale to Turkish / Ulakbim Dergipark Balıkesir Necatibey Education Journal / Article (2018)

Evaluation of the Creative Problem Solving Procedure by Cooperative Method of Basic Electronics and Measurement Course / XIV. European Conference on Social and Behavioral Sciences / Odessa, Ukraine / Presentation and Articles (2017)

Computer Programming with Arduino: Example Course Desing / Azerbaijan University / Proceedings (2018)

Development Study on Achievement Test for Programming Fundamentals Course (Original name of study:"Programlama Temelleri Dersi İçin Başarı Testi Geliştirme Çalışması")/ 12th International Computer & Instructional Technologies Symposium (ICITS) / Ege University / Presentation (2018)

Attitude Scale Towards Computer Programming and Psychometric Properties / Ulakbim Dergipark Ege Education Journal / Article (2018)

AWARDS

2019 Global Teacher Award, AKS Worldwide Pvt. Ltd. (2019)

İÇİNDEKİLER

Sayfa

İÇ KAPAK	i
KABUL ONAY SAYFASI	ii
ETİK KURALLARA UYGUNLUK BEYANI.....	iii
ÖZET	iv
EXTENDED ABSTRACT	vi
ÖNSÖZ	viii
TEŞEKKÜR	ix
ÖZGEÇMİŞ	x
İÇİNDEKİLER	xiv
ŞEKİLLER DİZİNİ	xvii
TABLolar DİZİNİ.....	xviii
SİMGELER VE KISALTMALAR DİZİNİ	xx
BÖLÜM I.....	1
GİRİŞ.....	1
1.1. Problem Tanımı.....	1
1.2. Araştırmanın Amacı	3
1.3. Araştırmanın Önemi.....	3
1.4. Alt Problemler	5
BÖLÜM II	7

2.1. Programlama.....	7
2.2. Programlama Eğitimi	8
2.2.1. Yurt Dışında Programlama Eğitimi	10
2.2.2. Yurtiçinde Programlama Eğitimi	12
2.2.3. Programlama Eğitiminin Yapılandırmacı Öğrenme Kuramıyla İlişkisi	16
2.3. Meslek Liselerindeki Durum.....	17
2.3.1. Mesleki Ve Teknik Eğitimin Tarihçesi.....	17
2.3.2. Mesleki Eğitimin Önemi	19
2.3.3. Öğrenci Profilleri.....	20
2.4. Programlama Öğretimindeki Zorluklar	22
2.4.1. Programlama Dillerinin Yapısı.....	22
2.4.2. Öğrenci Özellikleri	23
2.4.3. Programlama Konuları	24
2.4.4. Programlamaya Karşı Tutum	24
2.5. Robotik Programlama Nedir Ve Özellikleri Nelerdir?	25
2.6. Arduino ve Kullanım Özellikleri.....	29
2.7. Arduino' nun Programlamada Kullanımı.....	33
2.8. Araştırmayla İlgili Çalışmalar	35
2.8.1. Öğrencilerin Programlama Becerileri Ve Akademik Başarı İle İlgili Çalışmalar	35
2.8.2. Bilgisayar Programlamaya Karşı Tutum İle İlgili Çalışmalar.....	39
BÖLÜM III.....	42
YÖNTEM	42

3.1. Araştırmanın Modeli.....	42
3.1.1. Araştırmanın Deneysel Modeli	42
3.2. Çalışma Grubu	45
3.3. Araştırma Süreci	46
3.3.1. Hazırlık Aşaması.....	47
3.3.2. Çalışma Konularının Belirlenmesi Ve Ders İşlenişi.....	49
3.3.3. Materyal Temini.....	51
3.3.4. Uygulama Aşaması	51
3.4. Veri Toplama Araçları.....	54
3.4.1. Akademik Başarı Testi	54
3.4.2. Programlamaya Yönelik Tutum Ölçeği	57
3.5. Verilerin Analizi	58
3.6. Araştırmacının Rolü.....	58
3.7. Çalışmanın Güvenirliği ve Geçerlilik.....	59
BÖLÜM IV	61
BULGULAR.....	61
4.1. Akademik Başarı Testi ANOVA Analizleri	61
4.2. Programlamaya Yönelik Tutum Ölçeği ANOVA Analizleri	68
BÖLÜM V	73
KAYNAKÇA	78
EKLER	100

ŞEKİLLER DİZİNİ

<u>Şekil</u>	<u>Sayfa</u>
1 Mesleki eğitim süreci	21
2 Arduino Genuino UNO	30
3 Arduino IDE arayüzü	31
4 2004'ten bu yana Arduino kelimesinin Google arama trendi	33
5 Çalışması süreci	42
6 Haftalık ders tasarımı	51
7 Devre şeması örneği	52
8 Öğrenenlerin işbirlikli öğrenme yaklaşımı ile devre tasarımları	52
9 Öğrenenlerin bireysel kodlama ortamı	53
10 Ders sırasında yazılan bir kodlama örneği	53
11 Test sonuçlarının grafiksel gösterimi	66

TABLolar DİZİNİ

<u>Tablo</u>	<u>Sayfa</u>
1 Programlama dillerinin yapılarına göre sınıflandırılması.....	7
2 5.,6.,7. ve 8. Sınıflar için problem çözme ve programlama ünitesindeki konular, kazanım sayıları ve süreleri	13
3 Bilişim teknolojileri ve yazılım dersi öğretim programı	15
4 Araştırmanın akademik başarıya ilişkin deneysel desen tasarımı	43
5 Araştırmanın programlamaya yönelik tutuma ilişkin deneysel desen tasarımı	43
6 Araştırma deseni	44
7 Araştırma grupları.....	45
8 Sekiz haftalık uygulama süreci.....	48
9 Başarı testinin madde güçlük ve ayırt edicilik indeksleri.....	55
10 Akademik başarı testi uygulama isimleri	61
11 Grup frekansları	61
12 Test ortalamaları	62
13 Mauchy's Test of Sphericity analiz sonuçları	62
14 Grup içi etki testi	63
15 Test sonuçlarının karşılaştırılması	65
16 Bağımlı değişken isimleri.....	69

TABLULAR DİZİNİ(devam)

<u>Tablo</u>	<u>Sayfa</u>
17 Grup isimleri.....	69
18 Grupların test ortalamaları.....	69
19 Mauchly's test of Sphericity.....	70
20 Grup içi etki testi	70



SİMGELER VE KISALTMALAR DİZİNİ

<u>Kısaltmalar</u>	<u>Açıklama</u>
CEDEFOP	European Centre for the Development of Vocational Training
LİMME	Lise Mezunlarına Meslek Edindirme Projesi
MEB	Milli Eğitim Bakanlığı
MEGEP	Mesleki Eğitim ve Öğretim Sistemini Güçlendirme Projesi
METEK	Türkiye’de Mesleki ve Teknik Eğitimin Kalitesinin Geliştirilmesi Projesi
METEP	Mesleki ve Teknik Eğitim Projesi
MESGEP	Mesleki Becerilerin Geliştirilmesi Projesi
METGE;	Mesleki ve Teknik Eğitimi Geliştirme Projesi
OSANOR	Okul-Sanayi Ortaklaşa Eğitim Projesi
OSEP	Okul-Sanayi Eğitim Programları Projesi
SSRVM	Sri Sri Ravishankar Vidya Mandir
YÖK	Yüksek Öğretim Kurumu

BÖLÜM I

GİRİŞ

1.1. Problem Tanımı

Programlama eğitimi, günümüz teknolojisinin geldiği noktada bilgisayar teknolojisinin hedeflerine ulaşılabilmesi için gerekli olan bir yapıtaş ve eğitim alanıdır. Kişiyeye disiplinlerarası çalışma becerisi kazandırması bakımından da önemlidir.

Dünyadaki tüm ülkelerin daha üretken bir hale gelebilmesi bakımından oldukça büyük bir öneme sahip (Demirer ve Sak, 2016) olan programlama eğitiminin, kişiyeye bilişsel olarak pek çok katkı sağladığı savunulmaktadır (Çakıroğlu, Sarı ve Akkan, 2011; Fessakis, Gouli & Mavroudi, 2013). Akpınar ve Altun (2014), programlama sayesinde bireyin derslere karşı motivasyonunun arttığını, problem çözme, uzamsal ve analitik düşünme, işbirlikli çalışma ve öğrenme becerileri gibi üst düzey bilişsel becerilerinin geliştiğini vurgulamaktadır. Ayrıca programlama, verileri analiz etme, düzenleme, somutlaştırma ve çözüm önerilerini genelleme olarak da birçok zihinsel işlevi kapsamaktadır (Kalelioğlu vd, 2016). Alan yazında programlama eğitiminin ilk ve orta dereceli eğitim kurumlarının öğrencilerine kazandırdığı disiplinleri ve bilişsel becerileri belirlemek amacıyla birçok deneysel çalışma yapılmıştır (Burke, 2012; Fessakis vd., 2013; Gonzalez vd.,2017; Kalelioğlu, 2015; Kahn vd., 2011; Kobsiripat, 2015; Lin & Liu, 2012; Tekerek ve Altan., 2014; Qian & Lehman, 2016; Yünkül vd., 2017).Uzun yıllar boyunca araştırmacıların yaptıkları bu çalışmalar sonucunda elde edilen olumlu verilere dayanarak da birçok Avrupa ülkesi K12 (ilk ve orta dereceli eğitim kurumları) öğretim müfredatlarına programlama eğitimini dâhil etmişleridir. 2015 yılında Avrupa Okul Ağı tarafından yapılan bir araştırmaya göre 18 Avrupa ülkesinin programlama eğitimini ilköğretim öğretim programına dâhil ettiğini ve bu ülkelerin farklı sebeplerden dolayı öğretim programlarında programlama eğitimlerine yer verdikleri belirtilmiştir. Bu ülkelerin programlama eğitimini öğretim programlarına dâhil etme sebepleri ise şunlardır (Saygıner ve Tüzün 2017):

- Mantıksal Düşünme Becerisini Destekleme
- Problem Çözmeyi Destekleme
- Öğrencileri Bilişim Teknolojilerinin İçine Çekme

- Kodlama Becerilerini Destekleme
- Bilişim Teknolojilerinin İstihdamını Destekleme

Ülkemizde yapılan programlama eğitimi ise, öğrencilerin küçük yaştan itibaren ulaştığı bilgilerden yeni bilgiler üretebilen, ürettiği bilgileri pazarlayabilen, bilgiyi güce ve paraya dönüştürebilen kişiler olarak yetiştirilmesi için, ilköğretim ve ortaöğretim programlarında zorunlu ders olarak verilmektedir. Mesleki ve teknik eğitim veren kurumlardaki bilişim teknolojileri alanındaki öğrenciler ise, 11. sınıftaki dal derslerinde yoğun bir programlama eğitimi almaktadırlar. Bu dallardaki derslere hazırlık amacıyla da 10. Sınıfta “Programlama Temelleri” dersi verilmektedir.

2006-2007 eğitim-öğretim yılından itibaren mesleki ve teknik eğitim kurumlarında öğrenmeyi öğretip, öğrenciyi aktif kılan, kalıcı ve etkili öğrenmeyi gerçekleştiren, bireysel bir şekilde ilerlemeyi sağlayan modüler öğretim yöntemi benimsenmiştir. Fakat yapılan araştırmalarda modül içeriklerindeki görsel unsurların azlığı, öğrencilerin ilgisini çekebilme niteliğinin azlığı, modülün dilinin açık ve anlaşılır olmayışı, uygulamaya yeteri kadar ağırlık verilmeyişi, bilgi ve becerilerin gelişimini destekleyememesi gibi eksiklikler bulunmuştur (Gömlüksüz ve Erten, 2010). Mesleki ve teknik eğitim kurumlarındaki bilişim teknolojileri alan dersleri incelendiğinde ise bu derslerde C#, VBasic, PHP, ASP.Net, Javascript, Actionscript gibi programlama dilleri üzerinden eğitim verildiği görülmektedir. Fakat kullanılan bu programlama dilleri, giriş seviyesindeki öğrenciler tarafından oldukça zor bulunmakta ve daha süreç başından zor kabul etmeleri sebebiyle derse karşı olumsuz tutum sergilenmektedir. Bu yüzden de öğrenciler derste başarısız olmaktadır (Altun ve Mazman, 2012). Soyut olan bilgi işlem alanının, yalnızca "ekranda" görünen verilerin, çok fazla ilgi çekmediği ve öğrencilerin odaklanma sorunu yaşadıkları belirtilmektedir (Kafai vd., 2016). Bu yüzden öğrencilerin ders başarısını, derse karşı tutumlarını ve motivasyonunu arttırmak için bilgiyi kendilerinin yapılandırabileceği ortamlar sağlanmalıdır (Roth & Roychoudhury, 1994). Bu kapsamda programlama eğitiminde çeşitli teknolojik tasarımlar kullanılmaya başlanmıştır. Bunlardan en çok kullanılan teknoloji örnekleri ise şunlardır: Bloctopus (Sadler vd., 2015), ShrinkyCircuits (Lo & Paulos, 2014), LightUp (Chan vd., 2013), Scratch (Maloney vd, 2010), LilyPad Arduino (Buechley vd., 2008) ve Arduino Uno (Mellis vd., 2007). Bu teknolojilerin tasarımcıları, genellikle belirli öğrenme

hedeflerini öğretmeyi amaç edinmişlerdir. Örneğin; programlamayı öğretmek için Scratch, veya elektronik devreleri öğretmek için LittleBits kullanılmaktadır (Kafai vd., 2016). Programlama eğitiminde kullanılan bu teknolojik tasarımlar dünyada son yıllarda hızla artmış ve programlama eğitiminde karşılaşılan sorunları giderebilmek amacıyla da kullanılmaya başlanmıştır. Bu kapsamda teknolojik tasarımları içinde barındıran robot programlama eğitim öğretime dâhil edilmiştir.

Robot programlama öğrenciler arasında rekabet oluşturması, takım çalışmasına elverişli yöntemlerin kullanılmasına olanak sağlaması ve programlama sürecini daha ilgi çekici hale getirmesi sebebiyle rağbet görmektedir (Ersoy vd., 2011). Robotiklerin somut yapısı, nispeten soyut-sayısal kavram ve becerileri başarılı bir şekilde öğretmek için kullanılmaktadır (Witherspoon, 2018). Robot programlama materyalleri için çeşitli donanım platformları vardır fakat öğrenciler kolay erişilebilen, basit, ucuz, kolay bulunabilen donanımları tercih etmektedir. Bu yüzden bu çalışmada Arduino Uno kullanılmıştır.

Arduino Uno, bir giriş/çıkış kartıdır ve fiziksel bir programlama platformudur. Alan yazında Arduino ile programlama eğitimi hakkında yapılan incelemelerde programlama dillerinin öğretiminde soyut kavramlar ile çalışılmasından kaynaklı olumsuzlukların giderilmesinde Arduino ve benzeri platformların etkili bir çözüm olduğu, programlama ile ilgili soyut kavramların somutlaşmasına yardımcı olacağı, öğrencilerin yazdıkları kodu fiziksel olarak çalışır durumda gözlemleyebilecekleri belirtilmiştir (Ersoy vd. 2011; Gupta vd., 2012; Qiu vd., 2013; Kasalak, 2017). Cevahir ve Özdemir'e (2017) göre öğrenciler, giriş seviyesindeki programlama bilgi ve becerilerini yüksek seviyede bir elektronik bilgisine sahip olmadan, Arduino ile somut elektronik devre elemanları üzerinde kolayca uygulayabilmekte ve böylece Arduino kullanımı, kullanıcıların programlamaya olan ilgilerini arttırabilmektedir.

1.2. Araştırmanın Amacı

Bu çalışmada, "Programlama Temelleri" dersinde Arduino kullanımının, öğrencilerin akademik başarılarına ve programlamaya yönelik tutumlarına olan etkisinin belirlenmesi amaçlanmıştır.

1.3. Araştırmanın Önemi

Dünyanın her yerinde yaygın olarak verilen programlama eğitimi, programlama diline özgü kavramların karmaşıklığı ve soyutluğundan dolayı giriş seviyesindeki öğrenciler için zorlu bir engel olduğu görülmektedir (Yükseltürk ve

Altıok, 2016). Bu zorlukların, programlama becerilerinin öğrenilmesine engel olduğu ve öğrencilerin zamanla programlamadan soğumasına ve çaba sarf etmemesine yol açtığı düşünülmektedir (Ersoy vd., 2011).

Ülkemizde ve dünyada öğrencilere programlama öğretilmesi ile ilgili yapılan çalışmalarda, öğretim tasarımı ve medya araştırmaları ön plana çıkmakta (Demir, 2015) ve araştırmacılar öğrenci değişkenleri, medya ve medya karşılaştırmaları üzerinde yoğunlaşmaktadırlar (Erdoğan ve Çağiltay, 2009). Yapılan bu çalışma da, araştırma yöntemi ve uygun medya araçları ile metodun birlikte kullanılması bakımından önemlidir.

Öğretim teknolojileri alanının asıl amacı uygun yöntemleri ve medya araçlarını birlikte kullanarak öğrenmeyi daha iyi hale getirmektir. Endüstri 4.0 ile gelen büyük veri ve nesnelerin interneti, akıllı cihazlar ve 3D baskı teknolojileri gibi gelişmelere uyum sağlayabilecek şekilde üretim yetisi kazanmış insan kaynağına sahip olabilmek için ise öğrenme ortamlarının geliştirilmesi gerektiği öngörülmektedir. Naps ve diğerleri (2003), programlama dersinde öğrenme ortamının nasıl düzenlenmesi ile ilgili yaptıkları çalışmada, öğrencilerin kendilerine özgü tasarımlarını yapabilmelerinin ve verilen programlama örneklerini çözüme ulaştırabilmelerinin önemli olduğunu belirtmektedir. Bu kapsamda robotik eğitim, öğrencileri araştırmaya ve keşfetmeye daha çok teşvik ederek onların teknolojik açıdan bilgilenmesini ve takım çalışması yapabilme becerilerinde artış sağlaması sebebiyle günümüz eğitim dünyasında yerini almıştır (Şabanoviç ve Yannier, 2003). Eğitimde kullanılan robotik uygulamalar, öğrenmenin daha kalıcı ve anlamlı olmasını ve problemlere yaratıcı çözümler üretilmesini sağlamaktadır (Koç ve Böyük, 2013). Bu nedenle bu çalışmada da, programlama eğitimi sürecinde Arduino ile robotik uygulamalar geliştirilerek ve robotik programlamanın programlama eğitimindeki avantajları kullanılarak programlama öğrenimindeki zorlukların giderilebilmesi bakımından önem kazanmaktadır.

Programlama eğitiminin kendine özgü zorluklarının yanında bir diğer problem de programlama dersini alan öğrencilerin profilleridir. Günümüzde meslek liseleri akademik başarı düzeyi düşük öğrenci profili, mesleki eğitime gidenlerin toplumda olumsuz algılanması, ailenin beyaz yaka saplantısı, bireyin eğitiminde ailenin baskın etkisi gibi sorunlar içermektedir (MEB, 2012b; Kalkınma, 2014). Bu durum akademik başarısı yüksek olan öğrencilerin meslek

liselerini tercih etmemelerine neden olmaktadır. Fakat meslek liselerinin bilişim teknolojileri alanında yoğun bir programlama eğitimi verilmektedir. Bu yüzden bu çalışma, meslek liselerinde programlama öğretimi sırasında karşılaşılan zorlukları giderebilmeyi ve robotik programlamanın avantajlarını kullanabilmeyi öngörmektedir.

Ülkemizde, yapılan alanyazın incelemesinde öğrencilerin programlama becerilerini geliştirmek amacıyla birçok uygulama yapılmıştır. Benzer ve Erümit'in (2017) 2008-2017 yılları arasında ülkemizde programlama öğretimine dayalı yapılan 21 adet yüksek lisans ve 8 adet doktora tezi üzerinde yaptıkları araştırmaya göre Programlama eğitimi ile ilgili çeşitli çalışmaların yapıldığı, yapılan çalışmaların da genel olarak ortaokul, ön lisans ve lisans seviyesindeki öğrencilere yönelik olduğu görülmüştür. Böylece mesleki ve teknik eğitim alan lise öğrencilerine yönelik araştırma amacına benzer bir çalışmanın yapılmadığı sonucuna varılmış ve hedef kitle meslek lisesi öğrencileri olarak seçilmiştir. Ayrıca çalışma, liselerde farklı disiplinlerdeki öğretim sürecine yönelik Arduino uygulama yazılımlarının geliştirilmesine de ışık tutacaktır.

1.4. Alt Problemler

Araştırma amacına bağlı olarak şu sorulara cevap aranmaktadır;

1. Öğrenci başarısına yönelik olarak "Programlama Temelleri" dersinde;
 - a. Arduino kullanımı, deney grubu öğrencilerinin ön-test, son-test ve izleme testi başarıları arasında anlamlı bir fark oluşturmakta mıdır?
 - b. Kontrol grubu öğrencilerinin ön-test son-test başarıları arasında anlamlı bir fark oluşturmakta mıdır?
 - c. Arduino kullanımı, deney ve kontrol grubu öğrencileri arasındaki son-test başarı puanlarında anlamlı bir fark oluşturmakta mıdır?
2. Bilgisayar programlamaya karşı tutumlarına yönelik "Programlama temelleri" dersinde;
 - a. Arduino kullanımı, deney grubu öğrencilerinin bilgisayar programlamaya yönelik tutumları arasında anlamlı bir fark oluşturmakta mıdır?
 - b. Kontrol grubu öğrencilerinin bilgisayar programlamaya yönelik tutumları arasında anlamlı bir fark oluşturmakta mıdır?

c. Arduino kullanımı, deney ve kontrol grubu öğrencileri arasındaki bilgisayar programlamaya yönelik tutumları arasında anlamlı bir fark oluşturmakta mıdır?



BÖLÜM II

İLGİLİ YAYIN VE ARAŞTIRMALAR

Araştırmanın kuramsal çerçeve boyutunda, programlamanın tanımı yapılmakta ve programlama eğitimi hakkında bilgiler verilmektedir. Ayrıca Dünya’da ve ülkemizde programlama eğitimi ile ilgili yapılmış ve yapılmakta olan çalışmalardan örnekler verilmektedir. Programlama eğitiminde kullanılan öğretim yaklaşımı olarak da yapılandırmacılık kuramının işbirlikli öğrenme yaklaşımı seçilmiştir.

2.1.Programlama

Programlama, bir problemi tanımlama, problemi analiz etme, problemin çözümüne yönelik olarak algoritma geliştirme süreci olarak tanımlanmaktadır (Yünkül vd., 2017). Başka bir deyişle, gerçek hayatın modellenmesi olarak değerlendirilen programlama (Çölkesen, 2002), insan hayatına ilişkin bir olaya yönelik çözüm tasarımının bilgisayar ortamına aktarımıdır. Programcılar soyut olan programlamayı gerçek hayata yaklaştırmaya çalışırlar. Bu yüzden bir problemin çözümü için program gerçekleştirilirken komutlar düşünülmeden programlama yapısı için gerekli algoritma tasarlanır (Çölkesen, 2002). Daha sonra kullanıcılar, gerekli komutları bir programlama dili kullanarak yazar, derler ve çalıştırır (Arabacıoğlu, Bülbül ve Filiz, 2007).

1955’li yıllarda İngilizce olarak programlama ile ilgili kelimeler kısaltılarak oluşturulan programlama dilleri, insanlar tarafından ilgi görmüş ve daha sonraki yıllarda gelişerek çoğalmıştır. Gelişim yıllarına göre bilgisayar programlama dilleri, yapılarına göre tablo 1’de görüldüğü şekilde sınıflandırılmıştır (MEGEP, 2011; Kesici ve Kocabaş, 2007).

Tablo 1. *Programlama Dillerinin Yapılarına Göre Sınıflandırılması*

Kuşak	Programlama Dili	Periyod
1	Makine dili	1940-1950 aras ı
2	Assembly dili	1950’li yıllardan itibaren
3	Yüksek seviyeli diller	1960’li yıllardan itibaren
4	Çok yüksek seviyeli diller	1970’li yıllardan itibaren
5	Yapay zekâya yönelik diller	1980’li yıllardan itibaren

Kullanılacak programlama dili çözülecek probleme göre seçilir. Yazılan kodlar derleyici yardımıyla makine diline yani bilgisayarın anlayacağı biçime çevrilir. Programlama dillerinin sınıfları ise şu şekilde özetlenebilir (Elkner vd., 2010):

- Makine dili: Doğrudan makine komutlarından oluşan alçak seviyeli programlama dilidir (1 ve 0'lardan oluşur).
- Assembly dili: Anımsatıcı simgelerden oluşan sembolik makine dilidir.
- Yüksek seviyeli diller: Kaynak kodları kısa, derleme süreleri ise uzun olur.
- Çok yüksek seviyeli diller ve yapay zekâya yönelik diller: Az kod çok iş mantığıyla hareket eden esnek komutlardan oluşan programlama dilleridir.

Birçok farklı programlama dilleri olsa da sonuç olarak makine diline dönüşmeleri gerekmektedir. Çünkü makine dili bilgisayarın doğal dilidir. “1” ve “0” ‘lardan oluşan komutlardır ve bunlar doğrudan makineye hitap eder (MEGEP, 2011). Bunun için kullanılan yapıya derleyici denir. Derleyiciler, bir programlama dilinde yazılmış olan kaynak kodu makine koduna çeviren yazılımlardır. Derleyiciler sayesinde yazılan program satır satır ilerletilerek derlenir ve bu işlem sırasında programda yapılmış bir hata var ise işlem durdurularak programcıya hata ile ilgili bilgi verilir (Kesici ve Kocabaş, 2007).

2.2.Programlama Eğitimi

Programlama eğitimi, günümüz teknolojisinin gelişim sürecinde önemli bir eğitim alanı ve yazılım çalışmalarının var olmasını sağlayan bir yapıya sahiptir. Bu nedenle hem ülkemizde hem de Dünya’da bu eğitim alanında uzman bireyler yetiştirilmesi önem kazanmaktadır. Burada dikkat edilmesi gereken nokta programlama becerisinin, bireyin düşünme becerilerine olan katkısıdır (Kert ve Uğraş, 2009).

Günümüz bilgi ve teknolojisinin ve bilgisayarın hızla gelişmesi sonucu öğrenen becerileri de değişmekte ve daha çok teknoloji odaklı beceriler üzerine odaklanmaktadır (Gökoğlu, 2017). Bu beceriler arasında iletişim ve işbirliği, yaratıcılık, yenilik, dijital vatandaşlık, araştırma, karar verme, bilgi akışı ve programlama kavramları söylenebilir (ISTE, 2007; Gökoğlu, 2017). Programlama, bilgisayar tabanlı sistemlerin problemleri çözmeye yönlendirilmesi ve istekler doğrultusunda çalışmasını sağlamak için gereken temel bir beceridir (Köse ve Tüfekçi, 2015). Teknolojik okuryazarlık olarak da adlandırılan

programlama becerisi günümüz bireylerinin sahip olması gereken üst düzey bilişsel becerileri de kullanmayı gerektirir (Yükseltürk ve Altıok, 2015). Bu özelliğiyle programlama, farklı bilim ve araştırma alanlarında da temel düzeyde sahip olunması gereken bir yeterlilik olmuştur (Köse ve Tüfekçi, 2015). Bilgiye olan ihtiyacın artması ve bilgiye hızlı erişim isteği son zamanlarda programlama yaklaşımının eğitim-öğretim ortamlarına hızla yayılmasına sebep olmuştur (Kafai & Burke, 2015; Sterritt vd., 2015). Bilgisayar programlamaya artan ilgi ve istek, bireylere ilgili bilgi ve becerilerin kazandırılması hususunda eğitim süreçlerini yeniden şekillendirmeye yardımcı olmuştur (Gökoğlu, 2017). Akçay ve Çoklar' a (2016) göre programlama becerisi, bireylere eleştirel ve analitik düşünme, problem çözme, sorgulama ve algoritmik düşünme becerileri kazandırmakta ve geliştirmektedir. Ayrıca programlama becerisine sahip bir birey, küçük projelerin entegrasyonu ile ürüne yönelik daha karmaşık problemlere çözüm üretebilir ve yaparak öğrenme alışkanlığı kazanabilir (Akpınar ve Altun, 2014; Çakıroğlu vd., 2011; Fesakis & Serafeim, 2009). Bu yüzden bu becerilen öğrenilmesi gelişen teknolojiyle beraber daha da önemli olmaktadır (Akpınar ve Altun, 2014). Öğrencilerin en az bir programlama dilini etkili bir biçimde kullanması çok yönlü düşünebilmeyi, daha iyisini isteme arzusunu tetiklemeyi sağlamaktadır (Akçay ve Çoklar, 2016). Erdoğan'a (2005) göre programlama yapan kişilerde soyut düşünme becerisi yapmayanlara göre daha çok gelişmiştir. Ayrıca programlama yapan bireyler;

- öğrenmeye daha açık ve hevesli (Çetin, 2012),
- okuduğunu anlama ve yorumlama yeteneği yüksek,
- matematik ve satranç derslerinde başarılı (Erdoğan, 2005; Robins, Rountree & Rountree, 2003),
- programlama dillerinin çoğunun İngilizce olması sebebiyle İngilizce derslerinde akademik başarısı yüksek (Nowaczyk, 1984) bireylerdir.

Alanyazında programlamanın, ilk ve orta dereceli eğitim kurumlarındaki öğrencilerde sistematik düşünme yetisi kazandırdığı (Kafai & Burke, 2013), ve kariyer seçimlerinde kalıcılığı etkilediği (Margolis vd., 2010) ortaya konmuştur. Alan yazında yapılan araştırmaya göre tüm dünyada programlama eğitimi genel olarak lisans düzeyinde yapılmaktadır (Karabak ve Güneş, 2013). Ancak son zamanlarda programlama eğitimi yaygınlaştırmak ve sevdirmek amacıyla

ilköğretim öğretim programlarına eklenmiştir. Bazı ülkeler ilköğretim öğretim programlarındaki zaten var olan programlama derslerinde düzenlemeler yapıp içeriklerini güncellerken, bazı ülkeler ise programlama derslerini ilköğretim öğretim programlarına ilk kez dâhil etmişlerdir (Saygıner ve Tüzün 2017).

2.2.1. Yurt Dışında Programlama Eğitimi

Avrupa Okul Ağı (European Schoolnet) tarafından 2015 yılında yapılan bir araştırmaya göre, 18 Avrupa ülkesi ilköğretim öğretim programlarına programlama derslerini dâhil etmiştir. Bu ülkelerin programlama eğitimini öğretim programlarına dâhil etme sebepleri şunlardır:

- Mantıksal Düşünme Becerisini Destekleme
- Problem Çözmeyi Destekleme
- Öğrencileri Bilişim Teknolojilerinin İçine Çekme
- Kodlama Becerilerini Destekleme
- Bilişim Teknolojileri Alanındaki İstihdamı Destekleme
- Diğer Anahtar Yeterlilikleri Destekleme (Saygıner ve Tüzün 2017)

Programlama eğitimine önem veren ülkeler arasında bulunan İngiltere, 2013 yılında ilköğretim öğretim programına programlama derslerini eklemiş ve 5-16 yaş grubuna kodlama eğitimini 2014 yılında başlatmıştır. 2014 yılını ülke genelinde “Kodlama Yılı” olarak ilan ederek Avrupa Birliği’ nin “Çocuklara Kod Öğretme” farkındalığını yaratma girişiminin de bir parçası olmuştur. 2014 yılının Eylül ayından itibaren 5-6, 7-11 ve 11-14 yaş grubu bireyler için üç basamakta eğitimlere başlanmıştır (Özdemir, 2015).

Birinci basamak: Bu basamakta, görsellerle ve kolay anlaşılır talimatlarla 5-6 yaşındaki çocuklara algoritmanın ne olduğu anlatılmakta ve dijital içerikleri kontrol edip tekrar kullanma gibi bilgisayar becerileri kazandırılmaktadır.

İkinci basamak: Bu basamakta, veri toplama ve analiz etme gibi çalışmalar yapmalarını sağlayan atölyelerde 7-11 yaş grubundaki öğrenciler, karmaşık problemlere çözümler üretebilmekte ve gerektiğinde hataları ayıklayabilecek düzeye gelmektedirler.

Üçüncü Basamak: Bu basamakta, 11-14 yaşındaki öğrencilerden kendi istedikleri iki veya daha fazla programlama diline hâkim olması beklenmektedir.

İngiltere Hükümeti, öğrencileri bu seviyelere taşımaya yardımcı olacak öğretmenleri eğitmek için ise Microsoft ve Google gibi şirketlerle anlaşmalar imzalayarak hazırlık programları oluşturmaktadır.

Güney Kore’de ise programlama eğitimi, 2014 yılından itibaren öğretim programına kademeli olarak eklenmiş ve 2018 yılının başından itibaren ülkede uygulanmaya başlanmıştır. Pilot aşama olarak 2014 yılında 72 okul seçilmiş ve 2016 yılında 900 okulda yeni öğretim programlarına geçiş yapılmıştır. Bu derslerde öğrenciler, programlama dillerini kullanarak basit robotlar üretmeyi, oyunlar tasarlamayı ve hareketli öyküler oluşturmayı öğrenmektedir. Güney Kore hükümeti, öğretmenlere, eğitim ekibi ve yazılım eğitimi programcısı takımının her birine akıllı fen laboratuvarı, robotik, STEAM (Science, Technology, Engineering, Art and Math) eğitimi alanlarında destek vermiştir (Battal, 2017).

Amerika Birleşik Devletleri’nde, 2013 yılından beri kodlama eğitimi, sivil toplum kuruluşları, hükümet, Google ve Microsoft gibi şirketlerin desteğiyle yürütülmektedir. Aynı yıl Bill Gates ve Mark Zuckerberg gibi isimlerin de desteğini alan “Code.org” platformu kurulmuştur. Bu platformdan öğretmen ve öğrenciler ücretsiz olarak kodlama eğitimi alabilmektedir (Öndeş, 2016). Code.Org: Scratch, Alice, Robomind, Kodu Game Lab, MIT App Inventor, Light Bot gibi programlama araçlarını kullanmaktadır (Demirer ve Sak, 2016).

Teknolojiye giderek daha fazla yatırım yapan Estonya’da, ProTiger programının yaklaşımı sayesinde kodlama, robotik ve bilişim teknolojilerine aşına, yüksek seviyede teknoloji okuryazarlığına sahip bireyler yetiştirmek hedeflenmiştir. 2012 yılında ilköğretim 1. Sınıftan itibaren verilen kodlama eğitiminin önceliği öğrencilere problem çözme, eleştirel ve sorgulayıcı düşünme yeterliliğinin kazandırılmasıdır. Böylece kendi kendine araştırarak öğrenebilen bir öğrenci mühendislik bilimi, bilişim, kodlama gibi konularda gelişmeleri takip ederek gelişimini sürdürebilmektedir (Yıldız, 2018).

Avustralya’da 2015 yılında temel programlama dilleri ve kod eğitimi küçük yaşlardan itibaren verilmeye başlanmıştır. Bu düzenlemeye göre 5 yaşındaki ilkökul 1.sınıf öğrencisine temel kodlama eğitimi verilirken 7 yaşından itibaren de temel programlama mantığı öğretilmektedir (Kahraman, 2015).

Mühendislik eğitimine önem veren Hindistan, bilgisayar eğitimini her kademedeki öğretim programına eklemiştir. İlkokul 1. ve 4. Sınıflar arasında öğrenim gören öğrenciler, öncelikle algoritmanın temelleri üzerine

yoğunlaşmaktadır. Liseye kadar olan dönemde ise Basic programlama dili öğretilmektedir. Öğrenciler burada değişkenler, sabitler, karar yapıları, döngüler gibi konuları öğrenmektedir. Lisede ise ileri düzey programlar yazabilecekleri şekilde eğitim almaktadırlar (SSRVM, 2007).

İrlanda'da programlama eğitimi 2011 yılında James Whelton ve Bill Liao tarafından kurulan CoderDojo ile başlamıştır. CoderDojo, genç nesillerin kodlama işlemlerinin daha geniş kitlelerde uygulanmasını amaçlayan amatörce başlayan ve bilgisayar yazılımının ilk evrelerini öğretmeyi hedefleyen bir girişimdir. Bir bilgisayar kulübü olarak da nitelendirilen CoderDojo için düzenli olarak bir araya gelen bireyler, yazılım dünyasını küçüklere ve gençlere tanıtır, onların kod yazıp çalışan programlar üretebilmelerini sağlamaktadır (Stage-Co, 2017).

Macaristan'da Teknoloji firması Prezi'nin girişimiyle başlatılan Programcı Kızlar adıyla yürütülen proje kapsamında "kızların bu işe uygun olmadığı" yönündeki önyargıyı yıkmak amacıyla ortaokulda öğrenim gören kız öğrenciler programlama dili eğitimi almaktadırlar. Daha sonra bu proje sivil toplum örgütleri tarafından da desteklenmiştir (Euronews, 2015).

Fransa'da 2015 yılından itibaren okul öncesi dönemde temel programlama eğitimi verilmeye başlanmıştır. Bu kadar erken yaşta programlama eğitimine başlamadaki amaç, öğrencilerin görsel düşüncelerini ve sezgilerini geliştirmek olarak belirtilmiştir (Monroy-Hernandez & Resnick, 2008).

Dünya genelinde her yaştan öğrencinin kodlama öğrenebilmesi adına 2012 yılında Avrupa Okul Ağı (European Schoolnet) kuruluşu tarafından başlatılan EU Code Week (Avrupa Kod Haftası) organizasyonu ile öğrencilerin bilgisayar programlama yeteneği ve becerilerini geliştirmek aynı zamanda farkındalık yaratmak amacıyla çeşitli kodlama etkinlikleri gerçekleştirilmektedir. Ülkemizde Milli Eğitim Bakanlığı'nın da desteklediği bu organizasyonda öğretmenler veya okullar her sene belirlenen tarihler arasında düzenleyecekleri etkinlikler ile organizasyona dâhil olmaktadır (Dijital Eğitim, 2018).

Programlama eğitimi konusunda Dünya'da yapılan bu etkinlikler ve çalışmalar ülkemizde de dikkat çekmiş ve öğrenciler programlama eğitimine dâhil edilmiştir.

2.2.2. Yurtiçinde Programlama Eğitimi

Tüm dünyada olduğu gibi ülkemizde de programlama eğitimine verilen önem zamanla artmıştır. İlk kez 1998 yılında Milli Eğitim Bakanlığı tarafından

Seçmeli Bilişim Teknolojileri dersi, ilköğretim öğretim programlarına dâhil edilmiştir (Uzgun ve Aykaç, 2016). Buradaki esas amaç, temel bilgisayar okuryazarlığını öğrenciye kazandırmaktır. Bu alanda öğretmen yetiştirmek için ise 1998 yılında Yüksek Öğretim Kurulunu' nun eğitim fakültelerini yeniden yapılandırması sonucu Bilgisayar ve Öğretim Teknolojileri Öğretmenliği alanları açılmıştır (YÖK, 1998). 2012 yılına kadar dersin öğretim programında sürekli bir yenilik yapılmasına rağmen, zorunlu ders kapsamına alınmamıştır. Dersin zorunlu olmayışı derse verilen önemi azaltmış ve sadece bilgisayar donanımı olan sınıflara sahip okullardaki öğrenciler bu ders ile buluşturulmuştur (Uzgun ve Aykaç, 2016). Daha sonra Talim Terbiye Kurulu Başkanlığı'nın (TTKB) 2012-06-25 tarihinde yayınlanan 69 sayılı kararı ile “Bilişim Teknolojileri” dersinin adı değişerek “Bilişim Teknolojileri ve Yazılım” dersi olmuş (MEB, 2012a) ve haftada iki ders saati olmak üzere 5. ve 6. sınıflarda zorunlu, 7. ve 8. sınıflarda ise seçmeli ders kapsamına alınmıştır.

5. , 6., 7. ve 8. sınıflar için geliştirilen öğretim programında ünite temelli yaklaşım esas alınmıştır ve üniteler için hazırlanan kazanım sayılarında en yüksek oran problem çözme ve programlama ünitesinde yer almıştır. Tablo 2’de 5., 6., 7. ve 8. sınıflar için problem çözme ve programlama ünitesindeki konular, kazanım sayıları ve süreleri verilmiştir (MEB, 2018a; 2018b; 2018c).

Tablo 2. 5 , 6.,7. Ve 8. Sınıflar İçin Problem Çözme Ve Programlama Ünitesindeki Konular, Kazanım Sayıları Ve Süreleri

Ünite Adı	Konu Adı	Toplam Kazanım Sayıları	Süre Ders Saati	Oran %
			5. SINIF	
		27	36	50
			6. SINIF	
Problem Çözme Ve Programlama	1. Problem Çözme Kavramları ve Yaklaşımları 2. Programlama	25	36	50
			7. SINIF	
		14	32	44
			8. SINIF	
		15	32	44

5., 6., 7. ve 8. sınıflar için hazırlanan öğretim programındaki Problem Çözme ve Programlama ünitesinde; algoritma tasarımına ilişkin anlayış

geliştirme, problem çözmek için değişken, sıralı mantık, karar yapısı, döngü ve fonksiyon yapılarını kullanma becerilerinin kazandırılması amaçlanmıştır (MEB, 2018a; 2018b).

Ortaöğretimde ise Talim ve Terbiye Kurulu Başkanlığı'nın 2016 yılında 65 sayılı kararı ile Ortaöğretim Bilgisayar Bilimi Dersi Öğretim Programı, 2017-2018 eğitim ve öğretim yılında uygulamaya konulmuştur. TTKB'nin 2018 yılı 23 sayılı kararı ile de 2018-2019 yılından geçerli olmak üzere yürürlükten kaldırılıp güncellenerek yeni bir program yayınlanmıştır. Haftada 2 ders saati olarak düzenlenen "Bilgisayar Bilimi" dersi seçmeli ders kapsamına alınmıştır. Öğretim programı problem çözme ve programlama konularına odaklanan ders, istenilen sınıfta kur yaklaşımı ile tercih edilmektedir. Bu ders kapsamında "Kur 1" ve "Kur 2" olmak üzere iki kur vardır. Ders kapsamında robot kitleri, tabletler ve bilgisayarlar kullanılabildiği gibi farklı teknolojilere sahip olan kurumlar öğretim programındaki ünitelere göre çeşitli uygulamalar geliştirebilmektedir. *Kur 1* seviyesinde temel konular ele alınarak programlamaya giriş düzeyinde istenilen programlama dili seçilebilir. *Kur 2* düzeyinde ise programlama öğretimi için gerekli olan tüm yaklaşımlar seçeneklerle sunulmuştur. Bu derste, sınıf düzeyi fark etmeden öğrenci tarafından ilk defa seçilmesi durumunda Kur 1'e ait kazanımlar uygulanmaktadır. Ders ikinci defa seçilirse Kur 2'ye ait kazanımlar uygulanmaktadır (MEB, 2018c).

Mesleki ve teknik liselerde ise bilişim teknolojileri alanı yer almaktadır. Bu okullarda Anadolu Teknik ve Anadolu Meslek Programları bulunmaktadır. 9.sınıfta ortak programda eğitim alan öğrenciler 10. sınıfta Anadolu Meslek programlarında yer alan alanlara ayrılmaktadır. Anadolu Teknik programlarına ise Liseye Giriş Sınavı ile 9. sınıftan itibaren ilgili alanlara yerleşilmektedir. Bu alanlarda 10. sınıf seviyesinde alan ortak dersler alınmakta, 11. ve 12. sınıflarda ise dal dersleri alınmaktadır. Alanda Ağ İşletmenliği ve Siber Güvenlik, Bilgisayar Teknik Servisi, Web Programcılığı ve Veri Tabanı Programcılığı olmak üzere toplam dört dal bulunmaktadır. Öğrenciler, 10. Sınıf seviyesinde haftada 14, 11. Sınıf seviyesinde haftada 22 ve 12. Sınıf seviyesinde işletmede beceri eğitimi dersi ile birlikte haftada 28 saat bilişim dersleri almaktadır.

Yükseköğretim programlarında ise lisans seviyesinde bilgisayar mühendislikleri bölümlerinde yoğun programlama eğitimi verilmektedir. Sistem Programlama, Nesne Tabanlı Programlama, Algoritma Analizi gibi dersler alan

öğrenciler programlama becerilerini geliştirmektedir (Akçay ve Çoklar, 2018). Önlisans seviyesinde ise meslek yüksekokullarında bulunan bilgisayar programcılığı alanlarında algoritmik düşünme ve planlama yaklaşımı uygulamalarında öneriler sunan, bilişim problemlerinin çözümü için yazılım bileşenleri geliştirebilen öğrencilerin yetişmesi hedeflenmektedir (YÖK, 2011).

TTKB'nin 27/06/2018 tarihinde yayınlanan 91 sayılı karar ile de ilkökul 1., 2., 3. ve 4. sınıflarda Bilişim Teknolojileri ve Yazılım dersi öğretim programına alınmış ve dersin işlenmesinde basamaklı öğretim yaklaşımının kullanılması önerilmiştir. Basamaklı öğretim yaklaşımı, Öğrencilerin öğrenme biçimlerinin, hazırbulunuşluk düzeylerinin, bilgiyi işleme becerilerinin farklı olabileceği anlayışına dayanmaktadır (MEB, 2018e). Bu yaklaşım, dört temel basamaktan oluşmakta ve öğrencilerin problem çözme süreçlerinde bilişim teknolojilerini nasıl kullanabileceği üzerinde yoğunlaşmaktadır.

1. Basamak- Başlangıç Düzeyi (D1): İşlem akışlarına ve temel kavramlara ilişkin uygulamalar yer almaktadır.
2. Basamak- Orta Düzey (D2): Programlama mantığına giriş yapılmış ve bilişim teknolojileri ile ilgili detaylara yer verilmiştir.
3. Basamak- Gelişmiş Düzey (D3): Blok tabanlı programlama ortamlarında uygulamalar geliştirilmesine ve bilişim teknolojilerinin günlük hayattaki kullanımına ilişkin bilgiler yer almaktadır.
4. Basamak- İleri Düzey (D4): Programlama süreçlerindeki karmaşık uygulamalara ve bilişim teknolojilerinin güvenli ve doğru kullanımı yer almaktadır.

Tablo 3.'te Bilişim Teknolojileri ve Yazılım dersi öğretim programına ilişkin kazanım sayıları ve tema adları yer almaktadır.

Tablo 3. *Bilişim Teknolojileri Ve Yazılım Dersi Öğretim Programı*

Tema Adı	Düzeyi	Kazanım Sayıları	Toplam Kazanım Sayıları
Problem Çözme Ve Programlama	1.Düzey	7	41
	2.Düzey	11	
	3.Düzey	10	
	4.Düzey	13	
Toplam			86

Yapılan açıklamalardan ve tablo 2 ile tablo 3'te gösterilen verilerden elde edilen sonuca göre, ülkemizde verilen tüm eğitim seviyelerindeki bilgisayar derslerindeki temel hedef, öğrencilere problem çözme mantığını kavratarak programlama becerilerini geliştirmektir. Bu yüzden, Türkiye Yeterlilik Çerçevesi'nde belirlenen,

- Matematiksel yetkinlik ve bilim/teknolojide temel yetkinlikler
- Dijital yetkinlik
- Öğrenmeyi öğrenme
- Bağımsız çalışabilme ve sorumluluk alabilme yetkinliği
- Bilgi-işlemsel düşünme
- Yenilikçi ve yaratıcı tasarım
- Sosyal ve dijital vatandaşlık
- Bilgiyi yapılandırma
- Kendini geliştirerek karşılaştığı problemleri çözme

gibi yetkinliklere ulaşabilmek için Talim Terbiye Kurulu Başkanlığı, ders içeriklerine programlama becerilerini geliştirmeye yönelik uygulamalar eklemiştir. Bu da ülkemizde programlama eğitimine verilen önemi gözler önüne sermektedir. Öğrenciler bilişim derslerinde, bilgiye ulaşma ve yapılandırma süreçlerinde algoritma tasarımı ve uygun programlama yaklaşımını seçme ve uygulama konusunda kazanımlar edinmektedirler.

2.2.3. Programlama Eğitiminin Yapılandırmacı Öğrenme Kuramıyla İlişkisi

Öğrencilere bilimsel süreç becerilerini kazandırmak ve onları düşünme ve keşfetme olanaklarıyla buluşturmak için programlama eğitiminde seçilen öğretim yaklaşımı önem kazanmaktadır. Bu yüzden bireylere erken yaşlardan itibaren verilen programlama eğitimi yapılandırmacı öğrenme kuramıyla ilişkilendirilerek işbirliğine dayalı öğrenme yaklaşımıyla birlikte değerlendirilebilmektedir (Kert ve Uğraş, 2009).

Yapılandırmacı öğrenme kuramı, öğrencilerin düşünerek, anlayarak, kendi davranışlarını kontrol ederek öğrenmeleri gerektiğini vurgulamaktadır (Akpınar ve Ergin, 2005). Bu kuramın benimsendiği bir öğrenme ortamının sağladığı durumlar şöyledir (Akpınar ve Ergin, 2005; Açıkgöz, 1992):

- Öğrencilerin bireysel farklılıkları ve gelişim özellikleri dikkate alınarak derslerde farklı çalışma yapmalarına teşvik edilir.

- Çeşitli öğretim materyalleri kullanılarak, öğretim ilgi çekici, sürükleyici, verimli ve ekonomik hale getirilir.
- Öğrencilerin bireysel sorumluluk duygularını açığa çıkarır.
- Analiz, sınıflandırma gibi bilişsel kavramların öğrenciler tarafından kullanılması için fırsatlar verilir.
- Öğrenme pasif bir alma süreci değil, aktif bir süreçtir.
- Derslerde, öğrenciler diyalog içinde olarak, bilgi paylaşımında bulunarak işbirlikli şekilde sürdürülür.

Programlama eğitimi verilirken de yapılandırmacı öğrenme kuramının bu özelliklerinden yararlanarak etkin katılım ve etkileşimli bir öğrenme ortamı sağlamak için küçük grup tekniklerinin kullanılması öngörülmektedir. Alanyazına göre bu gruplarda öğrenciler, bilgiyi seçme, analiz etme, uygulama ve geliştirme işlemlerinde birbirleriyle işbirliği yaparak sonuca ulaşabilirler. İşbirlikçi öğrenme yaklaşımı ile öğrencilerden proje grupları oluşturulabilir, öğrenciler birbirlerinden kazandıkları bilgilerle yeni algoritmalar geliştirebilir, problemlere farklı çözümler geliştirebilirler (Erdem ve Demirel, 2002). Ayrıca öğrendikleri programlama dilini kullanarak çeşitli programlar üretebilirler.

Alanyazında işbirlikli öğrenme ortamları ile ilgili çeşitli araştırmalar yapılmış ve bu ortamların

- kalıcı öğrenmeye,
- öğrencilerin planlı çalışma becerilerine,
- ortak bir amaca yönelik çalışmaya,
- öğrencilerin akademik başarısını arttırmaya,
- işbirlikli çalışma becerilerine,
- derse katılıma ve derse karşı tutuma,
- öğrencilerin iletişim becerilerinin gelişimine,
- öğrencilerin problem çözme becerilerine,

eğitsel katkılarının olduğu belirtilmiştir (Bilgin ve Karaduman, 2005; Doymuş vd., 2004; Keskin ve Kılıç, 2016; Sezer ve Tokcan, 2003; Ünal ve Çakır, 2016).

2.3. Meslek Liselerindeki Durum

2.3.1. Mesleki Ve Teknik Eğitimin Tarihçesi

Mesleki eğitim Anadolu'da 12. Yüzyılda Selçuklular devrinde Ahilikle başlamıştır. Gençleri bir meslek sahibi yapmak için öncelikle ayakkabıcı esnafını

teşkilatlandırmış olan Ahi Evran, faydalı sosyal ve ekonomik görevler yapmış ve etkinliğini 17.yüzyıla kadar sürdürmüştür. Daha sonra Osmanlı Devleti'nin egemenliğindeki çeşitli dinler arasında ortak çalışma zorunluluğunun belirmesiyle Gedik adını taşıyan yeni bir teşkilat kurulmuş ve bu teşkilat, 19. yüzyılda Tanzimat' ın ilanına kadar devam etmiştir (Yörük vd. 2002). Bugünkü meslek liselerinin temelleri de 19. yüzyılın ikinci yarısından sonra atılmış ve ülke ihtiyaçlarına göre geliştirilmesine Cumhuriyet döneminde başlanabilmektedir (Dönmez ve Azizoğlu, 2010).

Milli Eğitim Bakanlığı 1926 yılında bu okullarla ilgili görev üstlenmiş ve 1941 yılında Mesleki ve Teknik Öğretim Müsteşarlığı kurulmuştur. Bu Müsteşarlık bünyesinde Erkek Teknik Öğretim Genel Müdürlüğü, Kız Teknik Öğretim Genel Müdürlüğü, Ticaret Öğretimi Genel Müdürlüğü yer almıştır (Yörük vd, 2002: 301).

1973 yılına kadar lise statüsünde görünmeyen mesleki ve teknik okullar, öğrencileri doğrudan mesleğe hazırlama işlevini üstlenirken genel eğitim veren ortaöğretim okulları ise yükseköğretime hazırlama işlevini üstlenmişlerdir. Daha sonra 1739 sayılı Milli Eğitim Temel Kanunu ile mesleki ve teknik okullara da lise statüsü kazandırılarak mezun olan öğrenciler yükseköğretime geçiş hakkını elde etmişlerdir (Altın vd, 2007:291).

1986 yılında 3308 sayılı Mesleki Eğitim Kanunu'nun çıkarılmasıyla birlikte meslek lisesi öğrencilerinin İşletmede Beceri Eğitimi dersi adı altında işyerlerinde uygulamalı eğitim almaları sağlanmıştır. Ardından mesleki eğitimin gelişimi ve planlanması için Bakanlıkta Mesleki Eğitim Kurulu, illerde ise İl Mesleki Eğitim Kurulları oluşturulmuştur (Altın vd, 2007:291).

1997 yılına kadar ortaöğretimler iki kademededen (1. Kademe: Ortaokul, 2. Kademe: Lise) oluşmaktaydı. 4306 sayılı İlköğretim ve Eğitim Kanunu'nun çıkmasıyla 5 yıl olan zorunlu eğitim 8 yıla çıkarılarak ortaokul kademesi ilköğretim bünyesine alınmıştır. Böylece mesleki eğitimlerde ortaokul uygulamasına son verilmiştir (Altın vd, 2007:291). 2011 yılında ise Millî Eğitim Bakanlığında meslekî ve teknik eğitimin yürütülmesinden sorumlu altı ayrı birim 652 sayılı kararname ile Mesleki ve Teknik Eğitim Genel Müdürlüğü adı ile tek çatı altına toplanmış ve günümüzdeki halini almıştır (MEB, 2014).

2.3.2. Mesleki Eğitimin Önemi

Bilim ve teknolojiadaki gelişme ve değişimler toplum yaşamını kültürel, politik ve ekonomik açıdan derinden etkilemekte ve böylece nitelikli insan gücüne ihtiyaç duyulmaktadır. Bu konuda mesleki ve teknik eğitim sistemine büyük sorumluluklar yüklenmektedir (Yörük vd, 2002: 301). Mesleki ve teknik eğitimin temel amacı, teknolojiyi kullanan, yorumlayan, geliştiren vasıflı insan gücünün yetiştirilmesidir. Bu nedenle ülkenin üretim ve rekabet gücünün artırılması için mesleki eğitim kritik bir öneme sahiptir.

Mesleki ve Teknik eğitim ile öğrenciler esnek bir yapı içinde kendi ilgi ve yeteneklerine göre hem bir üst öğrenime hem de iş hayatına hazırlanmaktadır (MEB, 2014). Dünyada söz sahibi olmak için güçlü bir ekonomiye sahip olunması gerektiğini tüm dünyaya kanıtlayan İngiltere, Avusturya, Almanya, Fransa gibi ülkeler, mesleki ve teknik eğitime önem vermekte ve teknoloji ile eğitimi etkili bir şekilde birleştirmektedir (Uçar ve Özerbaş, 2013). Gelişmiş ülkelerde işgücü piyasasının ihtiyaç duyduğu bilgi ve becerilerin eğitim sistemleri ile karşılanması gerekmektedir. Aksi durumda istihdam sorunları ortaya çıkacaktır. Bu yüzden bu ülkelerde mesleki eğitimden iş dünyasına hızlı bir geçiş olduğu ve mesleki eğitim mezunlarının genel eğitim mezunlarına göre iş bulma ve sürekli bir işe sahip olma olasılıklarının daha fazla olduğu görülmektedir (CEDEFOP, 2013).

Konu ile ilgili olarak OECD, APEC, UNESCO gibi kurumların yanı sıra üniversitelerdeki akademisyenler ve ülkelerin eğitim bakanlıkları 21. Yüzyıl yeterliliklerine yönelik çalışmalar yapmışlardır. Analitik beceriler, iletişim, karar verme, bilgi işleme, eleştirel düşünme, yaratıcılık, yenilikçilik, problem çözme, yardımlaşma, verimlilik, medya okuryazarlığı, liderlik ve sorumluluk gibi beceriler günümüz iş dünyasında kurum yöneticilerinin ve işverenlerin personellerinden bekledikleri becerilerdir. Bu kapsamda, mesleki ve teknik eğitim öğrencilerinin akademik bilgilerinin artırılması 21. Yüzyıl becerilerini kullanabileceği temelleri oluşturmaktadır.

Ülkemizde de mesleki eğitimde birçok yenilikler yapılmış ve ekonomik gelişmeler sayesinde yeni arayışlar içine girilmiştir. Bunun için mesleki eğitim öğretim programlarının hazırlanmasında işverenlerden görüşler alınmış, temel becerilerin (Türkçe, Matematik, Fen Bilimleri vb.) yeterince öğretilerek bilgi toplumunun ihtiyaçlarına cevap verebilecek uygulamalı mesleki eğitim verilmesi kararlaştırılmıştır (MEB, 2011). Çünkü öğrencilerin, öğretmenlerin gözetiminde

uygulamalı eğitim almaları öğretim programının hedeflerine ulaşmada kolaylık sağlamaktadır (MEB, 2011). Bu uygulamaları desteklemek için de birçok proje ve programlar (MEGEP; METEP; OSANOR, METGE; LİMME; MESGEP; OSEP; METEK) yürütülmüştür (Adıgüzel ve Berk, 2009).

2.3.3. Öğrenci Profilleri

1998 tarihinde yapılan yönetmelik değişikliği ile meslek liselerinde öğrenim gören öğrencilere yükseköğrenime giriş sınavında puan kısıtlaması getirilmiştir. Buna göre genel liselerde (Fen lisesi, Anadolu Lisesi vb) öğrenim gören öğrencilerin yükseköğrenim sınav puanları hesaplanırken Orta Öğretim Başarı Puanı 0,5 ile çarpılırken, mesleki eğitim almış öğrencilerin Orta Öğretim Başarı Puanı 0,2 ile çarpılmaktaydı. Üniversite eğitimi almak isteyen bir meslek lisesi öğrencisinin ya bir önlisans programını, ya da kendi alanıyla ilgili teknik eğitim fakültelerindeki lisans alanlarından birini seçmesi gerekmektedir. Bu durum, yükseköğrenim sınavında meslek lisesi öğrencilerini diğer öğrencilerden yaklaşık 30 puan gerisinde başlatmasına sebep olmuştur. Bu sebeple akademik başarısı yüksek öğrenciler genel liselere gitmeyi tercih etmişlerdir. Millî Eğitim Bakanlığının 28/9/2009 tarihli ve 25377 sayılı yazısı üzerine, 28/3/1983 tarihli ve 2809 sayılı Kanununun ek 30 uncu maddesine göre, Bakanlar Kurulu'nca 2/11/2009 tarihinde teknik eğitim fakültelerinin kapanmasıyla (Memurlar, 2018) da meslek liselerini üniversitede bir lisans programı okuma kaygısı olmayan (Dönmez ve Azizoğlu, 2010), sosyo-kültürel ve ekonomik düzeyi zayıf ailelerin çocukları (MEB, 2011) tercih etmeye başlamıştır.

Meslek liselerinin üniversite yolu kapalı, başarısı düşük öğrencilerin tercih ettikleri okullar imajını yıkmak için 2012 yılında uygulanmak üzere 2011 yılında katsayı problemi kaldırılmıştır. Böylece lise seviyesindeki tüm öğrencilerin eşit şekilde rekabet etmesi sağlanmıştır. Buna göre mesleki ve teknik eğitimde mesleki yönlendirmede öğrencilerin ilgileri ve yetenekleri yeterince dikkate alındığı ve öğrencileri meslek seçimine yönlendirecek rehberlik sistemi ile 21. yüzyıl yeterliliklerine ulaşılacağı belirtilmektedir (MEB, 2011).

2.3.4. Meslek liselerinin öğretim programı

Mesleki eğitim sisteminde 2006 yılındaki dokuzuncu kalkınma planında iş gücü piyasasının gerektirdiği temel becerilere sahip, ekip halinde çalışabilen, sorumluluk alabilen, sorun çözme ve karar verme becerilerinde yetkin bireyler yetiştirilmesi vurgulanmıştır (Dokuzuncu Kalkınma Planı, 2006: 40). Sektörün talebine cevap verebilecek olan bu yetkinliklere sahip nitelikli eleman yetiştirebilmek için ise 2000 yılında Türkiye ile Avrupa Birliği arasında

imzalanan antlaşmaya göre Türkiye'deki "Mesleki Eğitim ve Öğretim Sisteminin Güçlendirilmesi Projesi" (MEGEP) hayata geçirilmiştir. MEGEP ile meslek liselerinde modüler öğretim yaklaşımı benimsenmiştir. Bu sisteme göre her modül sonunda gerekli yeterliliklere sahip olan öğrenciye sertifika, yeterli sayıda sertifikaya sahip öğrenciye ise diploma verilmesi öngörülmüştür (Uçar ve Özerbaş 2013). Bu proje ülkemizde 2006-2007 eğitim ve öğretim yılı itibariyle tüm meslekî ve teknik eğitim kurumlarında kademeli olarak uygulamaya konulmuştur. Böylece mesleki ve teknik eğitim kurumlarında 9. sınıfların ortak okutulmasına ve 10. sınıftan itibaren öğrencilerin mesleki alan ve dallara ayrılmasına başlanmıştır. Öğrenciler, eğitim gördükleri alan ve dalda işyeri açma belgesine sahip olmakla birlikte sektörde çalışmak üzere teknisyen unvanı ile mezun olmaktadır. Ayrıca öğrenciler, transkript belgeleri ile yurt dışı iş başvurularında, diplomalarını ve eğitim sertifikalarını açıklama ve sunma konusunda yardımcı olan EUROPASS Sertifika Eklerine de sahip olmaktadır (MEB, 2018d). Şekil 1' de mesleki eğitim süreci yer almaktadır.



Mesleki ve Teknik Eğitim Genel Müdürlüğü'ne bağlı okullarda 60 alan ve 226 dal mevcuttur. Her alan ve dallardaki dersler için modüller yazılmış ve yazılmaya devam etmektedir. Gömleksiz ve Erten'in 2010 yılında yaptığı bir araştırmaya göre meslek liselerinde öğrencilerin modüllerin içeriklerine yönelik yaşadığı güçlükleri ortaya çıkarmıştır. Öğrenciler modüllerde,

- Görsel unsurların azlığını
- Modül içi etkinliklerin eksikliğini
- Öğrencilerin ilgisini çekebilme niteliğinin azlığını
- Modüllerde gereksiz bilgilere yer verildiğini
- Modülün dilinin açık ve anlaşılır olmadığını
- Uygulamaya yeteri kadar ağırlık verilmediğini ifade etmişlerdir.

Buna göre modüllerin öğrenci merkezli öğrenmeyi sağlayacak şekilde olmadığı ve öğrencilerin iş sektörünün ihtiyaçlarını karşılayacak biçimde yetişmedikleri (Gömleksiz ve Erten, 2010) sonucuna varılmaktadır. Bu yüzden

mesleki eğitim-öğretim programına programlama temel becerilerinin bütünleştirilmesi için çeşitli eğitim faaliyetlerinin ve materyallerinin geliştirilmesi gerekmektedir (MEB, 2011).

2.4. Programlama Öğretimindeki Zorluklar

Alanyazında programlama öğretimi ile ilgili yapılan çalışmalarda programlamaya yeni başlayan öğrencilerin birçok zorlukla karşılaştıkları sonucuna varılmıştır (Esteves & Mendes, 2004; Gomes & Mendes, 2007; Jenkins, 2002; Ko, 2007; Linn & Clancy 1992). İlk defa programlama eğitimi alan her üç öğrenciden birinin programlama dersinde karşılaştıkları zorluklardan dolayı başarı sağlayamadığı belirtilmiştir (Bennedsen & Caspersen, 2007). Bu araştırmalardan elde edilen sonuçlar dört başlık altında toplanmıştır.

2.4.1. Programlama Dillerinin Yapısı

Araştırmalar öğrencilerin programlamanın kavramlarını, programlama dillerinin dinamik yapısını ve çalışma mantığını anlamada zorluk çektiklerini ortaya çıkarmıştır. En yaygın programlama dilleri olan Nesne Tabanlı Programlama Dillerinin sahip olduğu soyut yapının nasıl işlediğini, çözüme nasıl ulaştığını öğrenciler zihinlerinde canlandıramadığı için dersin anlaşılması ve kavranması zorlaşmaktadır. Nesne Tabanlı Dillerindeki sınıflar, nesnelere, referanslar gibi kavramları çoğu öğrenci anlamakta ve bu özellikleri problemleri çözmek için kullanmakta başarısız olmaktadır (Esteves & Mendes, 2004). Bununla birlikte programlama dillerinin karmaşık sözdizimi de bu dillerin öğreniminde öğrencilere zorluklar yaşatmakta, öğrencilerin yazılan bir programı takip etmelerinde yetersiz kalmalarına sebep olmaktadır. Bu yüzden programlarda yapılan hataları bulmak için yoğun çaba sarf edilmekte ve zaman kaybedilmektedir. Öğrenciler program yazarken yazım hatası olmayan program yazmaya odaklanmakta ve mantıksal çıkarımlar yapmamaktadırlar (Linn & Clancy 1992).

Programlama dilleri dinamik bir yapıya sahiptir, çoğu öğrenme materyali (örneğin ders kitapları) bu dillerin dinamik davranışını açıklamakta yetersiz kalmaktadır. Çünkü programlama dillerinin öğreniminde, diğer birçok dersle karşılaştırıldığında daha çok uygulama etkinliklerine ihtiyaç duyulmaktadır. Bu yüzden dinamik bir yapıya sahip olan bu dersin sadece kitaplarla, düz anlatım vb. yöntemlerle anlatılması oldukça güçtür. Alan yazına göre de programlama

derslerinde hem teorik hem de uygulamalı bir ders yaklaşımı benimsenmelidir (Gomes & Mendes, 2007).

2.4.2. Öğrenci Özellikleri

Bilgisayar programlamada başarılı öğrencilerin demografik özellikleri, tutumları ve yetenekleri ile ilgili çok sayıda makale yayınlanmıştır (Bergin & Reilly, 2005; Byrne & Lyons, 2001; Gomes & Mendes, 2007, Konechi, 2014; Kinnunen vd., 2007; Simon vd., 2006; Wilson & Shrock, 2001).

Wilson ve Shrock (2001) yaptıkları çalışmada, öğrencilerin önceki programlama deneyimleri, çalışma stili tercihi, matematik geçmişi, cinsiyet, öğrencinin başarı veya başarısızlık için kendi özneliği ve öğrencinin kaygısı gibi faktörleri incelemiştir. Tüm değişkenler arasında başarı öncüsünün öğrencinin kaygı düzeyi olduğunu vurgulamışlardır. Bununla birlikte mantıksal ilişkilendirme ve problem çözme üzerine odaklanmanın da önemini belirtmişlerdir. Öğrencilerin problemlere çözüm bulmada yeterlilik geliştirmelerine yardımcı olmak için alternatif öğretim yaklaşımlarını araştırmanın gerekli hale geldiği söylenmiştir (Rogerson & Scott, 2010).

Her öğrenci çeşitli yollarla ve materyallerle farklı öğrenme biçimine sahiptir Bazıları öğrenmeyi yalnız bir süreç olarak görürken, bazıları ise akranlarıyla tartışma gibi daha dinamik bir öğrenme ortamını tercih edebilir (Gomes & Mendes, 2007). Fakat Jenkins (2002), Simon ve diğerleri (2006), birçok öğrencinin kendi kişiselleştirilmiş eğitim ortamı imkânlarına sahip olduğunu, asıl sorunun ise öğrencilerin sahip olduğu programlama becerisi olduğunu belirtmişlerdir. Ayrıca yapılan diğer araştırmalar, öğrencilerin diğer disiplinlerden gelen alışkanlıklarla (formül veya prosedür ezberleme vb) programlamanın temel kavramlarını anlamadan programlama öğrenme yoluna gittiklerini göstermiştir (Jenkins, 2002; Lahtinen, Ala-Mutka, & Järvinen, 2005; Gomes & Mendes, 2007). Ancak programlama farklı bir çalışma metodu gerektirir. Öğrencilerin olabildiğince fazla programlama problemi çözmesi gerekmektedir (Gomes & Mendes, 2007).

Cinsiyet değişkenini inceleyen araştırmalarda da kız öğrenciler, erkek öğrencilerle eşit derecede yüksek puanlar elde etmişlerdir (Byrne & Lyons, 2001). Burada üzerinde durulması gereken konunun öğrencinin cinsiyetinin değil, sahip olduğu matematiksel bilgi ve analitik düşünme becerisi olduğu vurgulanmıştır. Ayrıca, öğrencilerin programlamanın gerçek hayatta kullanım alanlarını

kestirememeleri, programlama dersi için gerekli olan soyut düşünme ve çözüm üretme yeteneklerindeki eksikliğin de programlama öğreniminin önündeki engeller olduğu belirtilmiştir (Saygıner ve Tüzün, 2017).

2.4.3. Programlama Konuları

Programlama öğreniminin zorluklarından biri olan programlama konularının yapısı öğrencilerde başarısızlığa sebep olmaktadır. Buna çözüm olarak konu seçiminin ve yapısının önemli olduğu ve basitten karmaşığa doğru sıralama yapılmasının gerektiği belirtilmiştir (Demir, 2015).

Öğretmenler, öğrencilerin en çok zorlandıkları konuların “diziler” ve “döngüler” konularının olduğunu söylemiştir. Öğrenciler dizilerdeki aynı veri tipinden birçok değer tutabilme durumunu günlük hayatla ilişkilendiremediklerini belirtmişlerdir. Buna ek olarak öğrenciler, problemin çözümüne yönelik algoritma üretmekte ve bunu akış şemaları ile ifade etmekte zorlanmakta, ayrıca değişken yaklaşımı ve bir değişkene değer atamanın nasıl gerçekleştirildiği konularında da zorlanmaktadır (Saygıner ve Tüzün, 2017).

2.4.4. Programlamaya Karşı Tutum

Tutum, birey davranışlarını saptayan ve davranışların geri planında kendini gösteren psikolojik bir parametredir (Smith, 1968). Bu kapsamda eğitim öğretim ortamlarında derslere yönelik öğrenci tutumları derslerdeki öğrenme davranışlarının temelini oluşturmaktadır.

Alanyazında bilgisayar programlama dersinin öğrenci davranışları üzerinde bir takım sosyal ve duygusal değişimler oluşturduğu görülmektedir (Demir, 2015). Öğrencilerde oluşan üzüntü, dikkatin dağılması, tekrar görev almama, çözüm üretmekten vazgeçme, programlamanın zor ve asla anlayamayacaklarını düşünmeleri gibi olumsuz tutumlar sebebiyle programlama öğreniminden uzaklaşmaktadır (Beck, Emery & Greenberg, 2005). Bu yüzden öğrenciler problemlere çözüm üretmekte, zihinsel şemalar oluşturmakta zorlanmakta ve programlamaya karşı aktif olamamaktadır.

Çoğu öğrenci, programlama çalışmak için yeterli motivasyona sahip değildir, çünkü öğrenciden öğrenciye geçen programlama ile ilgili son derece olumsuz bir algı vardır. Öğrenciler programlama derslerinde zorlayıcı bir etkenle karşılaşırsa onları motive olarak görmek zorlaşacaktır (Gomes & Mendes, 2007).

Erol ve Kurt (2017), yükseköğrenim öğrencilerinin programlamaya karşı tutumlarını belirlemek için yaptıkları çalışmada, katılımcıların programlama

dilleri ve programlama öğrenmeyle ilgili olumlu ve olumsuz tutumlarına ilişkin görüşlerini belirlemişlerdir. Buna göre öğrencilerin olumlu tutum görüşlerinin;

- programlamanın eğlenceli olmasına,
- bilgisayara hükmetme hissi,
- bir şeyler üretme düşüncesinden kaynaklandığını,

olumsuz tutum görüşlerinin ise;

- temel mantığı kavrayamama,
- zor olması,
- başarısız olma,
- sınav kaygısı,
- sıkıcı olması,
- problemin analizinde yaşanan zorluklardan kaynaklandığı sonucunu elde etmişlerdir.

Ayrıca katılımcıların tutumlarında cinsiyete göre herhangi bir değişim olmadığı, erkek ve kadın katılımcıların programlamaya karşı tutumlarının benzer olduğu sonucuna ulaşılmıştır.

2.5. Robotik Programlama Nedir Ve Özellikleri Nelerdir?

Öğrenciler bir programlama dili öğrenirken dilin kavram ve işlem açısından soyut kalması nedeniyle zorlanmakta ve bu bilgileri somutlaştırmaya ihtiyaç duymaktadırlar. Robotik programlama da bu süreci somutlaştırmaya yardımcı olmaktadır (Ersoy vd., 2011). Robotikler somut bağlamı, nispeten soyut sayısal kavram ve becerileri başarılı bir şekilde öğretmek için kullanılabilir (Witherspoon, 2018).

Robotik kitler, birçok farklı etkinlik tarzını ve birçok farklı öğrenme biçimini desteklemek için çeşitli şekillerde kullanılabilir. Ancak her bir kit, bazı etkinlik ve öğrenme biçimini diğerlerinden daha iyi destekler. Örneğin, LEGO Mindstorms robotik kiti özellikle mobil robotlar geliştirmek için çok uygundur. Robotik kitler, öğrencilerin yalnızca hareket değil, aynı zamanda ışık, ses ve müzikle ilgili sanatsal ürünler yaratmasını da sağlamaktadır. “Müzik Yapımcıları” temalı bir çalışmada, katılımcılar programlanabilir yeni müzik aletleri; “İnteraktif Takı Atölyesi”nde çeşitli giyilebilir ürünler; bir “Storybook Scenes Atölyesi”nde tanıdık bir kitaba veya filme dayanarak robotik karakterler yaratmışlardır (Rust vd., 2008).

Dünyada artan bir hızla yaygınlaşmaya başlayan robotik programlamada amaç, akıllı nesnelere, programlanabilir robotlar, kendin yap kitleri ve setleri gibi fiziksel bir araçla programlama eğitimini vermektir. Bu kapsamda ülkeler, Parallax Robotics Kitleri (Robotics Arduino Shield Kit), Makeblock Kitleri (Educational Robot Kit), Lego Mindstorms Kitleri, Robo Mind ve Primo, VEX IQ Platformu Kitleri (Starter Kits), Fischertechnik Kitleri gibi araçlar ve bu araçların desteklediği programlama dillerini (Arduino, mBlock, Mindstorm Nxt Education, Microsoft Robotics Developer Studio R4, Parallax Propeller C ve ROBOT C) kullanmaktadır (Numanoğlu ve Keser, 2017). Bu ortamların gelişimi ve robotik eğitime verilen önemin artması ile de birçok firma robot eğitim kitleri ve setleri üretmeye başlamıştır.

Robotik programlama, birden fazla eğitimsel yönü kapsamaktadır ve öğrencilerin problem çözme becerilerinin, yaratıcılıklarının, eleştirel düşüncelerinin ve işbirlikli çalışma becerilerinin geliştirilmesini sağlamaktadır (Alimisis & Kynigos, 2009). Robotların tasarlanması ve programlanması süreci öğrencilere önemli mühendislik, matematik ve bilgisayar bilimi kavramlarını öğretmektedir (Druin & Hendler, 2000). Ayrıca robotik eğitim, öğrencilerin araştırma tutumlarını geliştirebilmekte ve varsayım yapmalarına, deney yapabilmelerine ve soyut düşünme becerilerini geliştirmelerine olanak sağlamaktadır. Bu nedenle robotik (öğrenme nesnesi olarak görülen) yoluyla yapılan öğrenme, okul derslerinin daha geniş spektrumuna ait diğer bilişsel alanlar için de değerli görülmektedir (Alimisis & Kynigos, 2009).

Alanyazın incelendiğinde robotik eğitim ile ilgili yapılan birçok çalışma yer almaktadır. Hubwieser ve diğerleri (2015), aralarında Almanya, İngiltere, Fransa, Rusya ve Finlandiya gibi ülkelerin de bulunduğu 12 farklı ülkedeki ortaöğretim seviyesindeki bilgisayar programlama eğitimleri hakkında bilgi edinmek için 14 farklı çalışma gerçekleştirmiştir. Yapılan tüm çalışmalarda verilen programlama eğitiminin amaç ve hedeflerine yönelik kullanılan donanım aracının ve programlama dilinin değerlendirilmesi yapılmıştır. Buna göre bu çalışmalarda, kullanılan programlama araçlarının Arduino, Raspberry Pi, LEGO Mindstorms gibi donanım ve elektronik tabanlı sistemlerin tercih edildiği görülmüştür. Programlama dilleri olarak da seçilen elektronik ortamın (Arduino Uno kart için Arduino IDE Processing/Wiring dili) sahip olduğu kendi dillerinin yanında C#, Java, Python gibi diller kullanılmıştır.

Gezici ve diğeri (2017), önlisans öğrencileri ile programlama öğrenimindeki soyut kavramları somutlaştırmak ve gerçek dünya ile ilişki kurarak öğrenimi kolaylaştırmak amacıyla 14 haftalık uygulamalı ve teorik bir ders içeriği geliştirmişlerdir. Bu ders içeriğinde öğrenciler arduino kullanarak özgün robotlar tasarlamış ve esnek programlama yeteneklerini geliştirmişlerdir. Ayrıca öğrenciler farklı derslerde öğrendikleri bilgileri bu çalışmada bir araya getirmiş ve programlama dersleri hakkında olumlu görüş belirtmişlerdir.

Yadagiri ve diğeri (2015), programlamanın temel kavram ve bilgilerini lise seviyesindeki öğrencilere öğretmen için robotik programlama ile bir robot tasarlamışlardır. Blok tabanlı görsel programlama aracı ile geliştirilen bu robot, labirent üzerinde çıkış bulmayı amaçlayan bir eğitim oyunu tabanına sahiptir. Yapılan öğrenci değerlendirmesinde, öğrencilerin programlamayı eğlenceli, hızlı ve kolay bir şekilde öğrendikleri belirtilmiştir.

Karim, Lemaignan ve Mondada (2015), robot kullanımının ortaöğretim seviyesindeki öğrenmede yeni stratejiler geliştirilebileceğini, farklı derslerle bütünleştirilebileceğini ve sınıfları yeniden şekillendirebileceğini incelemek amacıyla bir çalışma gerçekleştirmiştir. Bu çalışma, Fizik ve Matematik derslerinde robotik platformların kullanımına dayanmaktadır. Çalışma sonunda bu platformların öğrenmede olumlu bir gelişmeye katkı sağladığı görülmüştür. Ayrıca robotların, eğitim faaliyetlerinin öğrenilmesinde, yaratıcı düşüncenin geliştirilmesinde ve problem çözme becerilerinin geliştirilmesinde de olumlu bir rol oynadığını savunmaktadırlar. Bununla birlikte robotlarla etkileşim, motivasyonu, bağlılığı ve eğitime karşı tutumu da olumlu yönde etkilediği belirtilmiştir. İlgili araştırmacılar robot tasarımı ve montaj işleminin basitleştirilmesi, sezgisel, görsel ve sürükle-bırak programlamanın dâhil edilmesi ve eğitici robot platformlarının aşamalı olarak azalan maliyetiyle, eğitim teknolojisinde yeni bir çağın ortaya çıkışını gösterdiğini savunmaktadır.

Merino ve diğeri (2016), kablosuz iletişim için STEM eğitim laboratuvarında yaygın bir şekilde kullanılan kablosuz robotik eğitim platformlarını (Wheeled Robot based on NI myRIO, Dr Robot Wireless Networked Autonomous, Makeblock mBot - STEM Educational Robot Kit, LEGO MIDSTORMS, DFRobotShop Rover v2 Bluetooth Kit) incelemişlerdir. Çalışmada sonucunda kullanılacak platformlarda koruma özellikleri, esneklik,

eşzamanlılık, yeniden yapılandırma olanakları, uygunluk, ölçeklenebilirlik ve prototipsel yapı gibi özelliklere de sahip olması gerektiği tespit edilmiştir.

Çavas ve diğerleri (2012) tarafından yapılan bir çalışmada ise ilköğretim fen ve teknoloji müfredatındaki 6. ve 7. sınıf öğrencilerinin fen bilgisi dersi becerilerindeki başarıları üzerine Lego Mindstorms tabanlı robotik dersinin etkisini araştırmışlardır. Yapılan çalışmada sorgulamaya dayalı robotik etkinlikler gerçekleştirilmiştir. Öğrenciler gruplar halinde işbirliği içinde çalışmış, teknolojik tasarımlar ve robot programlamayı kullanarak bilimsel araştırma teknikleri üzerine yönlendirilmişlerdir. Araştırma sonucunda yapılandırmacılık kuramı ışığında öğrencilerin gerçek dünya problemlerine yaratıcı çözümler üretebildikleri belirtilmiştir.

Mauch (2001), öğrencilerin problem çözme tekniklerini keşfetmelerine ve matematiksel problemleri çözmeye yardımcı olmak için LEGO Mindstorms robotik kiti etkinliklerine dayalı bir çalışma yapmıştır. Çalışmada ayrıca LEGO Mindstorms robotik kitinin öğrenciler arasında artan bir merak uyandırabileceği ve problemlere yaklaşırken kendi düşünce süreçlerini incelemelerini teşvik etmenin yollarını tartışmışlardır. Araştırmacı derslerde, işbirlikli öğrenme yaklaşımı kullanılarak problem çözme ile ilgili olarak robotik eğitimin faydalarını da incelemiştir. Çalışma sonucunda öğrencilerin robotlarını düzeltmek için kullandıkları teknik ve stratejilerin, gelecekteki sorunlarla karşılaştıklarında onlara yardımcı olacağı belirtilmiştir. Çünkü robotik sistemlerle çalışırken, bir robotun neden belirli bir işlevi yerine getiremediği hemen belli değildir. Bu yüzden öğrencilerin hem programın hem de robotun yapısını kontrol etmeyi öğrendikleri vurgulanmıştır. Ayrıca robotik sistemlerin öğrencilere, eğer bir problemi çözmek istiyorsa, sadece robotun neden doğru çalışmadığını tespit etmekle kalmayıp, sorunu diğer grup üyelerine de ifade edebilmelerini de öğrettiği belirtilmiştir. Böylece robotik sistemler ile gerçekleştirilen bir eğitim, öğrencilere, problemlere hem bir çözüm sunmasını hem de cevabı belirlemede yer alan süreçleri belirtmeleri gerektiğini göstermiştir.

Resinovic (2015) tarafından yapılan bir çalışmada ise insansı robot Nao kullanılarak öğrencilerin programlama becerilerini geliştirmeleri incelenmiştir. Bu çalışmada robotik eğitimin öğrencilere hızlı ve kolay bir şekilde programlama öğreteceği ve motivasyonu arttıracacağı vurgulanmıştır.

Numanoğlu ve Keser (2017), öğrencilere değişkenler, diziler, karar yapısı ve döngüler gibi programlamanın temel kavramlarını öğretmek amacıyla mBlock programlama ortamı kullanarak bir çalışma gerçekleştirmişlerdir. Bu çalışmadan elde edilen sonuçlara göre robotik eğitimle verilen programlama öğretimindeki soyut kavramlar somutlaştırılmış ve öğrenciler yazdıkları programın etkisini anında gördükleri için de bilgi işlemsel düşünme becerileri hızlı şekilde geliştirilmiştir.

Saleiro ve diğerleri (2013), robotik sistemleri (Raspberry Pi, Arduino, PIC microcontroller vb.) kullanarak verdikleri eğitimde, öğrencilerin matematiksel düşünme becerilerini geliştirdiklerini ortaya koyan bir çalışma yapmışlardır. Bu sistemlerin sahip oldukları yazılımlara ek bir yükleme olmadığı için de ilkökul seviyesindeki öğrencilerin bile kolay ve rahat kullanabileceklerini belirtmişlerdir.

Yapılan çalışmalar incelendiğinde hem ilk ve orta dereceli eğitim kurumlarında hem de yükseköğretimde programlamaya yönelik yapılan birçok çalışma, bilgi işleme sürecine, çaba ve yansıtma faaliyetlerine ve problem çözmeyi öğrenmeye ilişkin olumlu sonuçlar ortaya çıkarmıştır (Lye & Koh, 2014). Bu nedenle Avrupa Birliği, üye devletlerin eğitim sisteminde bilişimsel düşünmeyi en uygun ve koordine bir şekilde sunmak için deneyimleri ve kaynakları paylaşmayı amaçlayan TACCLE3 gibi projeler finanse etmektedir (Rees, García-Penalvo, Jormanainen, Tuul & Reimann, 2016).

2.6. Arduino ve Kullanım Özellikleri

Arduino, öğretim topluluğu içinde giderek yaygınlaşan bir giriş çıkış kartı ve Processing/Wiring dilinin bir uygulamasını içeren, açık kaynak kodlu geliştirilen, kolay kullanımlı, esnek, donanım ve yazılım tabanlı fiziksel bir programlama platformudur (Rubio vd., 2013; Arduino, 2019). Basit elektronik öğrenim uygulamalarında kullanılmakta ve programlanabilir sensörler veya LED'ler ile iletişim kurarak eylemler gerçekleştirmekte veya bir bilgisayarla çevre birimi olarak çalıştırılabilmektedir (Herger & Bodarky, 2015). Dileyen herkesin hazır basılmış ve bileşenleri yerleştirilmiş halde alabilecekleri veya baskı devreleri indirerek kendi devrelerini basabilecekleri elektronik ürünlerdir (Yılmaz, 2014). Çok çeşitli geliştiriciler, her türlü hesaplama sistemi için bir geliştirme platformu olarak Arduino' yu seçmiştir. Ayrıca açık kaynak kodlu olması sayesinde fikirlerini, projelerini ve çözümlerini paylaşan geniş bir kullanıcı topluluğu tarafından desteklenmektedir (Rubio vd., 2013). Arduino ile tek başına

kullanılarak interaktif nesnelere geliştirilebileceği gibi bilgisayar üzerinde çalışan yazılımlar ile bağlantı kurularak çeşitli ürünler oluşturulabilmektedir (Yılmaz, 2014).

Arduino gibi açık kaynak platformlarının yükselişinden önce, donanım ve yazılım prototipleme karmaşık ve pahalı olup, bu nedenden yirminci yüzyılın sonlarında birçok üniversite ve araştırma merkezi daha ucuz ve daha kolay alternatifler geliştirmeye başlamıştır. Fakat bu çözümler genel olmayıp, buldukları kurumun dışında popüler kullanımları mevcut değildi. Bu durum 2005 yılında IVREA enstitüsündeki proje öğrencisi Massimo Banzi'nin Arduino'yu ortaya çıkarmasına kadar devam etmiştir. Massimo Banzi Arduino'yu, programlama konusunda daha önce tecrübesi bulunmayan öğrencilerin fiziksel dünyayı dijital dünyaya bağlayan çalışma prototipleri oluşturmasına yardımcı olmak için tanıtmıştır (Arduino, 2019).

Arduino'nun Esplora, Mega ADK, Leonardo, Genuino MEGA 2560, Genuino UNO ve LilyPad gibi değişik modelleri vardır. Üzerinde Atmega 328, Atmega 168 ve Atmega 2560 gibi mikrodenetleyicilerden biri vardır ve her biri farklı özelliklere sahiptir. Bu modellerden en çok kullanılanı ise 6 analog girdi pinine, 14 dijital girdi-çıkış pinine ve Atmega 328-P mikrodenetleyici çipine sahip olan Arduino Genuino UNO modelidir.

Şekil 2. Arduino Genuino UNO (Arduino, 2019)



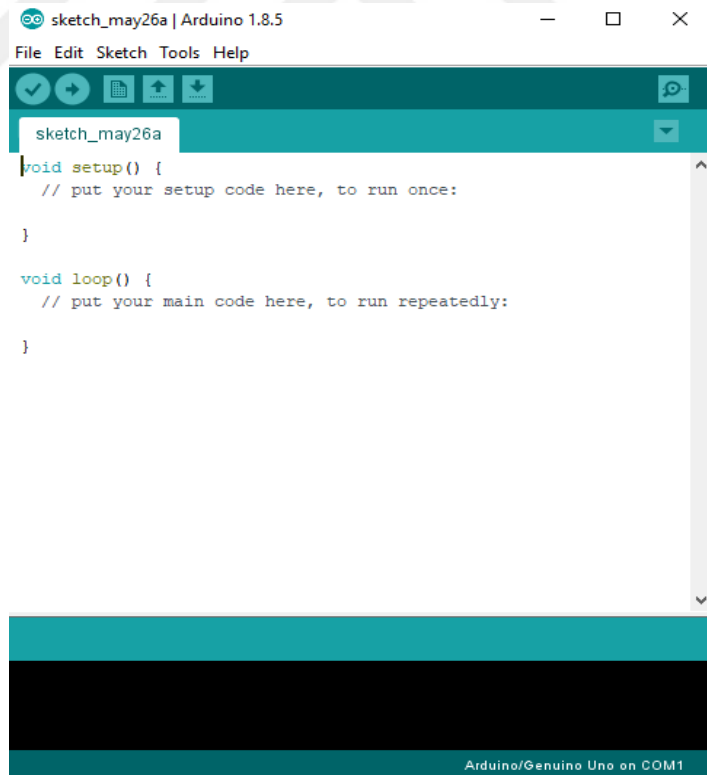
Şekil 2'de Arduino Genuino UNO'nun görseli verilmiştir. Bu modele üzerindeki bileşenler şu şekildedir.

- FTDI USB çip
- USB ve Power Jackları

- Güç pinleri
- 6 adet Analog giriş - çıkış pinleri
- 14 adet Dijital giriş - çıkış pinleri
- Bellek üniteleri (RAM, ROM vb.)
- Mikrodenetleyici
- Reset Butonu

Arduino bir elektronik kart olmasının yanında bir de kullanıcılara ücretsiz sunduğu platform bağımsız bir programlama arayüzüne sahiptir. Arduino IDE olarak adlandırılan bu arayüz MAC OS, Linux ve Windows gibi sistemlerde kolaylıkla kullanılabilir. Fakat bazı araştırmacılar IDE arayüzünün yetersiz olduğunu, iyi tasarlanmadığını, dokümantasyon bakımından etkin olmadığını yaptıkları çalışmalarda ortaya koymuşlardır (Balogh, 2010). Ancak bu problemler hem sürekli gelen güncellemelerle hem de farklı programlama dilleri ile kullanıma izin vermesi sebebiyle çözümlendiği söylenebilir.

Şekil 3. *Arduino IDE Arayüzü*



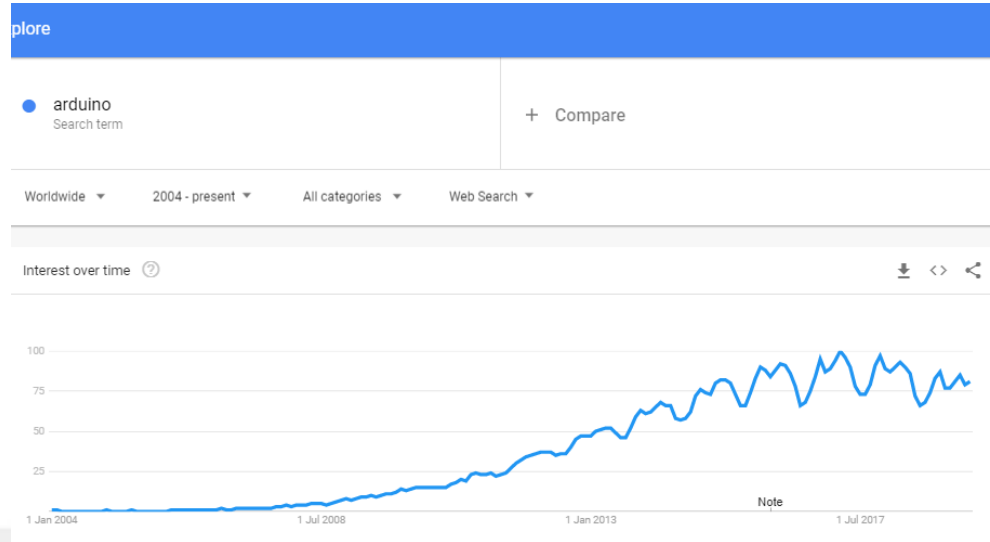
Arduino'nun açıklığı ve kullanım kolaylığı, mikro denetleyici tabanlı elektronik projelerin toplu olarak benimsenmesine yol açmış ve Maker Hareketi'nin yaratılmasında bir katalizör olmuştur (Arduino, 2019). Arduino,

programlama yönüne veya mimari yönüne fazla bakmadan mikrokontrolörlere bir giriş sağlamaktadır. Arduino IDE'de kullanılan Processing uygulanması, C'yi ya da herhangi bir dili daha yeni öğrenmeye yeni başlayanlar için geçişi kolaylaştıran C / C++ dilinin diyalektini kullanmaktadır. Ayrıca Arduino çevresinde pek çok grafik dili oluşturulmuştur (örn. Visuino, Ardublock, mBlock, miniblock, Snap4Arduino veya Embrio). Bu da onu programcı olmayanlar için bile erişilebilir kılmaktadır (Reas & Fry, 2014). Çok sayıda çevrimiçi eğitici ve soru sormak için geniş bir kullanıcı tabanına sahiptir. Bu nedenle, öğretme -öğrenme girişimi ve özellikle de bilişsel düşünme ve problem çözme süreçlerini geliştirmek için uygun bir platform olarak seçilmiştir (Barr & Stephenson, 2011; Grover & Pea, 2013).

Yıllar içinde farkındalığın ve ona olan ilginin arttığı Arduino, 2004'ten itibaren Google aramalarında çok aranan bir kelime olarak karşımıza çıkmıştır (Şekil 4). Bunun nedenleri arasında aşağıda verilen bilgiler yer almaktadır (Yılmaz, 2014).

- Arduino kullanarak çevresiyle etkileşime girebilen kolay sistemler tasarlanabilir.
- Açık kaynak kodlu bir geliştirme platformudur.
- Üzerinde Atmega firmasının 32 ve 8 bit mikro denetleyicileri bulunur.
- Mikrodenetleyicileri Arduino kütüphaneleri ile kolaylıkla programlanabilir.
- Analog ve dijital veriler, üzerinde bulunan analog ve dijital girişler sayesinde işlenebilir.
- Sensörlerden alınan veriler çeşitli çalışmalar için kullanılabilir.
- Ses, ışık, hareket vs gibi çıktılar üretilebilir.

Şekil 4. 2004' Ten Bu Yana Arduino Kelimesinin Google Arama Trendi



2.7. Arduino' nun Programlamada Kullanımı

Alanyazında hem ilk ve orta dereceli eğitim kurumlarında öğrenim gören öğrencilerin hem de üniversite öğrencilerinin programlama becerilerini öğrenmeleri ve geliştirebilmeleri için Arduino kullanarak çeşitli çalışmalar yapılmıştır. Örneğin;

Junior ve diğerleri (2013) Ortaöğretim öğrencileriyle Arduino Uno platformuna dayanan düşük maliyetli bir eğitim robotu setini, adım adım yaklaşım ve blok yapılı bir ortam kullanarak, standart ve tescilli bileşenlere göre diğer alternatiflerden daha önemli avantajlar bildirerek test etmiştir.

Assaf (2014), robotlar için evrensel bir programlama dili olan BotSpeak'i ilk ve orta dereceli eğitim kurumlarının öğrencilerine tanıtarak, Arduino'yu tercih edilen üç platformdan (Raspberry Pi ve LEGO Mindstorms ile birlikte) biri olarak seçmiştir. Arduino, ortaokul öğrencilerine nesne yönelimli paradigmayı öğretmek için Squeak e-oyuncakları ile birlikte kullanılmış ve bilişsel alanda olumlu sonuçlar elde edilmiştir (Jang, Lee, & Kim, 2015).

Portekiz 'de (Guimaraes) Minho Üniversitesi'nde düzenlenen üç günlük bir kamp olan RoboParty' de, elektronik mühendisliği, makine mühendisliği ve okul çağındaki çocuklara programlama öğretmek için takım temelli bir yaklaşım seçilerek çeşitli kültürel ve sportif etkinlikler düzenlenmiştir. Bu etkinliklerde Arduino tabanlı bir robotik kiti olarak kullanılan Bot'n Roll One sayesinde katılımcılar arasında mühendislik alanına ilgi artmıştır (Eguchi, 2014).

Kafai ve Burke (2013), lise öğrencileri ile yaptığı çalışmada yaratıcı özgürlüğün, öğrencilerin projelerini tasarlarken büyük bir mülkiyet hissi oluşturmalarına yardımcı olduğunu gözlemlemişlerdir.

Üniversite düzeyinde, Rubio, Hierro ve Pablo (2013) tarafından Arduino modüllerinin etkililiğinin araştırıldığı çalışmada, geleneksel yöntemlerin kullanıldığı programlama dersleri ile Arduino modüllerinin kullanıldığı programlama dersleri karşılaştırılmış ve araştırma sonucunda öğrencilerin Arduino ile programlama yapmayı daha çok sevdiğini ve daha fazla sayıda öğrencinin programlamayı öğrenebildiği tespit edilmiştir. Benzer şekilde, Aldridge, Brandt ve Parikh (2016), Arduino Uno ile bir PBL stratejisi kullanarak, lisans öğrencilerinin, özerk eğitim robotlarını kendi başlarına inşa ederken deneyimledikleri tasarım sürecini gözlemlemişlerdir. Bütçe kısıtlamaları olduğunda bu platformun en uygun olduğu sonucuna varmışlardır.

Kasalak (2017), ortalama 35'şer kişilik 5. ve 6. sınıf öğrencileriyle haftada 2 ders saati olarak zorunlu dersler arasında yer alan bilişim teknolojileri ve yazılım dersi kapsamında 5 hafta boyunca Arduino ve Scratch ile robotik programlama etkinlikleri gerçekleştirmiştir. Çalışma sonunda robotik kodlama etkinliklerinin öğrencilerin oldukça ilgilerini çektiğini, etkinliklere istekli olarak katıldıklarını ve etkinlikler süresince eğlendiklerini, öğrencilerin etkinliklerin okulda daha başarılı olmalarına yardımcı olacağına inandıklarını belirtmiştir.

Qiu ve diğerleri (2013), 12-17 yaş aralığındaki 36 öğrenci ile 3 haftasonu boyunca toplam 36 saatlik Arduino kullanarak robotik etkinlikler düzenlemişlerdir. Çalışma sonunda, öğrencilerin teknolojik özgüven, derse katılım ve çalışma durumlarında önemli artışlar görmüş, oturumun başlangıcından sonuna kadar etkinliklere ilgi gösterdikleri sonucuna varmıştır.

Gupta ve diğerleri (2012), 15-18 yaşları arasındaki 150 öğrenci ile temel programlama ve donanım becerileri hakkında bilgi sahibi olmak için Arduino kullanarak yaptıkları çalışmada öğrencilerin verilen tüm problemlerin çözümü için birbirleriyle işbirliği yapmaları konusunda hevesli olduklarını belirtmiştir. Geri bildirimlere dayanarak, öğrencilerin hayal gücünü ekran üzerinde canlandırmaya veya arayüz kullanarak somut oyunlara dönüştürme fırsatını elde ettiklerini görmüşlerdir.

Sohn (2014), 5 hafta boyunca LED tasarımına dayalı 10 bölümlük içerikten oluşan Arduino tabanlı programlama eğitimi verdikleri 26 öğrenci ile

gerçekleştirdikleri çalışmada, geleneksel programlama eğitimlerine kıyasla öğrenme akışlarının, öğrencilerin problem çözme yeteneği, ürün oluşturma teknikleri gibi alanlarda daha etkili oldukları görülmüştür.

Özdemirci ve diğerleri (2017), Arduino setinin programlama dersinde uygulanması üzerine gerçekleştirdiği çalışmasında, öğrencilerin donanımsal açıdan ve kodlama açısından eğitimlerine olumlu yönde katkı sağladığını belirtmiştir. Öğrencilerin Arduino Uno uygulama seti ile ilgili görevlendirmelerde donanımsal açıdan, algoritma geliştirme ve oluşturulan uygulamanın yazılıma dökülmesi ile ilgili görevlendirmelerde kodlama açısından etkili olduğu sonucuna varılmıştır.

Bununla birlikte, mikrodenetleyici tabanlı giriş seviyesindeki bir programlama eğitiminde Arduino'nun kullanılması artık yeni bir fikir değildir, ancak mevcut çalışmadaki yenilik seçilen ekran destekli öğrenme stratejisinden gelmektedir.

2.8. Araştırmayla İlgili Çalışmalar

2.8.1. Öğrencilerin Programlama Becerileri Ve Akademik Başarı İle İlgili Çalışmalar

Alan yazında öğrencilere programlama becerilerini öğretmek amacıyla hem K12 düzeyinde hem yükseköğretim düzeyinde birçok çalışma yapılmıştır.

Çocuklara erken yaşta kodlama öğretmek amacıyla üniversiteler, sivil toplum kuruluşları, kamu kurum ve kuruluşları ile şirketler çeşitli etkinlikler organize etmişlerdir. Buna örnek olarak 2017 yılında Microsoft Türkiye ve bir sivil toplum kuruluşunun organize ettiği “Minik Parmaklar Geleceği Programlıyor” projesi geliştirilmiştir. Bu proje ile çocukların programlama dili ile tanışmaları ve kod okuryazarlığının artması hedeflenmiş ve Türkiye'nin farklı illerinde sınıf ortamında 8-12 yaş arasındaki çocuklara, 8 saatlik “KODU Programlama Dili” eğitimi verilmiştir (Habitat, 2018).

Gonzalez ve diğerleri (2017), programlama eğitiminin bireylerin problem çözme becerisi üzerindeki etkisini incelemek amacıyla, yaşları 10 ile 16 yaş aralığında olan 1251 İspanyol öğrenciye 2 hafta boyunca Scratch ve Code.org platformlarında programlama etkinlikleri düzenlemişlerdir. Etkinlikler sonunda yapılan problem çözme test sonuçlarında, programlama öğrenimini ile öğrencilerin akıl yürütme kabiliyetleri ($r = 0.44$) ve problem çözme yetenekleri ($r = 0.67$) arasında istatistiksel olarak pozitif yönde anlamlı korelasyon bulunmuştur.

Yünkül ve diğerleri (2017), Scratch platformunda bir sömestr dönemi boyunca 12 yaşındaki 69 öğrenci ile yaptıkları programlama eğitimi çalışmasında, öğrencilerin algoritmik düşünme ve yaratıcı düşünme becerileriyle problem çözüme yeteneklerinde anlamlı artış gözlemişlerdir.

Qian ve Lehman (2016), matematiksel beceri ve programlama eğitimi arasındaki ilişkiyi incelemek amacıyla, yaşları 12 ile 14 olan 69 öğrenciye 14 hafta boyunca giriş seviyesinde Pascal programlama dili dersleri vermiş ve yapılan nicel ölçümler sonucunda matematik becerisi ve programlama performansı arasında güçlü ilişkiler bulmuşlardır.

Kodlama alanındaki bazı projeler ise dezavantajlı durumdaki kız öğrencilerine yöneliktir. Bir sosyal girişimcinin geliştirdiği “KızKode” projesi ile daha çok erkekler tarafından tercih edilen bir sektör gibi görünen yazılım sektöründe kadınların da aktif rol oynamaları için kız çocuklarına C programlama dili öğretilmiştir. Bununla birlikte Arduino devre kartı ve yazılımı ile mekanik kontrol uygulamaları yapılmıştır. Küçük yaştaki çocuklara da Scratch programı ile algoritma eğitimi verilmiştir. Bu projedeki başlıca amaç kodlama eğitiminde çocukların bilime ilgi duymalarını sağlamak ve çocuklarda farkındalık yaratmaktır (Baş, 2016).

Akçay ve Çoklar (2018), 107 bilişim teknolojileri aday öğretmeni ile yaptığı çalışmada, öğrencilerin yüzde elli seviyesinde programlama öz yeterliliğinin bulunduğunu buna karşın sorgulama ve problem çözüme becerilerinin ise yüksek olduğu sonucuna varmıştır. Ayrıca programlama öz yeterliliği yüksek olan öğrencilerin sorgulama becerisinin de yüksek olduğunu belirtmiştir.

Kobsiripat (2015), Scratch programlama ortamlarını kullanarak bir sömestr dönemi boyunca verdiği programlama eğitiminin öğrencilerin yaratıcılığını geliştirdiği sonucunu elde etmiştir.

Kalelioğlu (2015), 10 yaşındaki öğrencilerle yaptığı çalışmada programlama eğitiminin öğrencilerin problem çözüme becerilerinde kendilerine olan güvenlerini geliştirdiğini ortaya çıkarmıştır. Bu nedenle, programlama etkinliklerinin öğrenci gelişimine katkı sağlaması ve sadece tüketicileri değil, üretken insanlar olabilmeleri için kendi oyunlarını yazma şansı vermesi nedeniyle, programlamanın eğitime erken entegrasyon yapılması gerektiğini, sonuç ne olursa olsun, öğrencilerin mümkün olan en kısa zamanda, öğrencilerin gelişimini,

ihtiyaçlarını ve beklentilerini göz önünde bulunduracak şekilde planlanmış programlama aktivitelerine giriş yapmaları gerektiğini belirtmiştir.

Tekerek ve diğerleri (2014), Scratch programlama platformunda 60 ortaokul öğrencisiyle yaptıkları çalışmada, öğrencilerin dersi eğlenceli bulduklarını ve onların derse karşı olumlu tutum sergiledikleri sonucuna ulaşmışlardır.

2014 yılında çeşitli üniversitelerin ve Türkiye Bilişim Derneği' nin desteğiyle "Bilgisayar Programlama Çocuk Oyuncağı" adlı bir etkinlik organize edilmiş ve ilköğretim ile ortaöğretim düzeyindeki 100.000 öğrencinin program yazması hedeflenmiştir. Bu etkinliğin amacı, öğrencileri erken yaşta programlama araçları ile tanıştırmak ve program yazma farkındalığını arttırmaktır (Demirer ve Sak, 2016).

Kim, Chunk ve Yu (2013), bilgisayar programlamaya dayalı yaratıcı düşünme davranışları için bir öğretim programı önerdikleri çalışmada, öğrenciler gerçek yaşamda karşılaştıkları sorunları çözmek için programlama becerilerini kullanarak yaratıcı düşünmeye teşvik edilmiştir. Araştırmaya 30 üstün yetenekli olmak üzere 149 öğrenci katılmıştır. Sonuç olarak deney grubunda bulunan öğrencilerin problem çözme becerileri, kontrol grubuna göre daha fazla olduğu görülmüştür.

Fessakis ve diğerleri (2013), anaokulu öğrencileriyle yaptığı çalışmada Logo tabanlı bir yazılımla verilen programlama eğitiminin onların matematiksel kavramları anlama, problem çözme ve sosyal becerilerini geliştirdiği sonucuna varmıştır.

Burke'un (2012), 7 hafta boyunca ortaokul öğrencileri ile Scratch ortamlarında yaptığı etkinlikler sonucunda öğrenciler, programlama öğrenimini eğlenceli ve yaratıcılıklarını geliştirdiğini belirtmişlerdir.

Çetin (2012), 8 hafta boyunca öğrencileri problem çözme becerilerinin geliştirilmesinde programlama eğitiminin etkisini araştırmıştır. Bu araştırmada programlama eğitimi ile ilgili hem velilerden hem öğrencilerden görüşler alınmıştır. Alınan geri bildirimlere göre, verilen eğitimden memnun kalınmış, programlama eğitiminin problem çözme becerilerini arttırdığı ve bilgisayar kullanımına karşı olumlu tutum oluştuğu belirtilmiştir.

Lin ve Liu'nun (2012) 9 ve 10 yaşlarındaki öğrencilerle, Denner ve Werner'ın (2007) 126 ortaokul öğrencisiyle yaptıkları çalışmalarda, programlama

eğitiminin, öğrencilerin kendilerine güvenlerini geliştirdiğini ve onların problem çözüme, işbirlikli çalışma özelliklerinde artış oluşturduğu sonucuna ulaşımlardır.

Kahn ve diğlerleri (2011), 10 hafta boyunca ortaokul öğrencileri ile yaptığı çalışmada, programlama eğitiminin öğrencilerin derse karşı motivasyonlarını arttırdığını, onları keşfetmeye yönlendirdiğini belirtmiştir.

Yüksek öğrenim öğrencileri üzerinde yapılan 5 deneysel çalışmanın sonuçları, programlamayı deney grubundaki öğrencilerin metafor, zihin haritalama ve işbirlikçi öğrenme ve çoklu-duyusal yöntemler kullanarak ikili programlama gibi stratejilerle öğrendiklerini göstermiştir (Hui & Umar, 2011; Katai & Toth, 2010; Köse, Koç ve Yucesoy, 2013; Jonassen, 2011). Diğler yandan, 3 yükseköğretim çalışması, kontrol ve deney grupları arasında anlamlı bir farklılık olmadığını rapor etmiştir (Garner, 2009). Anlamlı olmayan sonuçlar, deney grubunda görevlere harcanan sürenin kontrol grubuna göre daha az olması, kısa deney süresi veya katılımcı sayısının az olması (Hsiao & Brusilovsky, 2011) gibi sebeplere dayandırılmaktadır. Deney ve kontrol grubu arasında anlamlı farklılık bulunmamasına rağmen, bu çalışmalar deneyin, zayıf öğrenciler üzerinde olumlu bir etkiye sahip olduğunu bildirmiştir. Öğrencilerin yarı-tamamlanmış çözümler üzerinde çalışması (Garner, 2009), programlama örneklerine ilişkin yorumların gözden geçirilmesi (Hsiao & Brusilovsky, 2011) gibi stratejiler, zayıf öğrenciler için yararlı olarak değerlendirilmektedir. Daha zayıf öğrenciler de, ikili programlamayı deneyimledikten sonra daha iyi öğrencilerle aynı düzeye gelmiştir (Goel & Kathuria, 2010) ve oyun stratejisi oluşturulmasıyla yüzde 80 oranında öğrencilerin başarı puanları artmıştır (Jiau, Chen, & Ssu, 2009). Seçilen programlama dili ile belirlenen sürelerde uygulamalı eğitimler verilmiş ve verilerin analizinde sonuçlar deney grupları lehine olumlu sonuçlanmıştır.

İki çalışma haricindeki (Yağcı, 2016; Özdiñç ve Altun, 2014) hem yükseköğrenim hem de ilk ve orta dereceli eğitim kurumları için yapılan çalışmaların ortak özelliđi, öğrencilere problem çözüme mantığını kavratmayı hedeflemeleridir. Hem yurtiçinde hem de yurtdışında yapılan programlama eğitimlerinde incelenen özellikler şu şekildedir (Çatlak vd 2015):

- Problem Çözüme [Sistemantik ve Mantıksal Düşünme, Karar Alma, Planlama, Yaratıcılık]
- Farklı Derslerde Kullanımı
- Programa Dair Öğrenci Görüşleri

- Duyuşsal Özellikler [Motivasyon, Tutum, Öz güven]
- Algoritma ve Programlama Öğrenimi

2.8.2. Bilgisayar Programlamaya Karşı Tutum İle İlgili Çalışmalar

Alanyazında öğrencilerin bilgisayar programlama öğrenmeye yönelik tutumlarına ilişkin pek çok çalışmaya erişilebilmiştir.

Fesakıs ve Sesafım'ın (2009), öğrencilerin bilgisayar programlama öğrenmelerine karşı tutumları ile ilgili yaptıkları çalışmada, görsel programlama aracı kullanarak yapılan programlama eğitiminde öğrencilerin stres ve kaygı yüzdelerinin azaldığı belirtilmektedir. Benzer bir çalışma olarak Bishop-Clark, Courte & Howard (2007) da 2,5 hafta boyunca görsel programlama aracı kullanarak yaptıkları çalışmada öğrencilerin programlama konusunda kendilerine olan inançlarının ve güvenlerinin arttığını, kaygılarının azaldığını ve programlama kavramlarını ve yapılarını daha iyi anladıklarını söylemişlerdir.

Erol ve Kurt'un (2017), yaptıkları bir çalışmada öğrenciler programlamaya olan olumsuz tutumun olumluya dönüşmesi için programlama öğretimi sürecinde daha fazla uygulama olmasını, algoritma öğretimine ayrılan sürenin artmasını ve hazırbulunuşluk düzeylerine dikkat edilerek yapılan öğretim tasarımının kullanılmasını belirtmişlerdir. Böylece programlama derslerinde daha fazla eğlenebileceklerini, bu işten zevk alacaklarını ve kaygı düzeylerinin azalacağını eklemişlerdir.

Karaci'nin (2016) yükseköğrenim öğrencileri ile yaptığı çalışmada öğrenciler programlamada kendine güven ve güdülenme ile programlamanın faydası hakkında tarafsız, programlamanın başarıya karşı tutumuna yönelik olumlu, programlamada başarının sosyal algısı hakkında olumsuz tutum sergilediklerini belirtmişlerdir. Ayrıca öğrenciler arasında cinsiyete göre programlamaya yönelik tutum bakımından anlamlı bir fark bulunmamıştır. Buna ek olarak, programlama amacıyla bilgisayar üzerinde harcanan günlük süre arttıkça olumlu tutumların da arttığı gözlemlenmiştir.

Korkmaz ve Altun'un (2013), bilgisayar mühendisliği ile bilgisayar ve öğretim teknolojileri öğretmenliği öğrencileriyle yaptığı çalışmada öğrencilerin programlama öğrenimini önemli ölçüde gerekli gördüklerini fakat yüzde elli oranında istek duyduklarını belirtmişlerdir. Araştırmanın çalışma grubunu 8 farklı üniversitede öğrenim gören 731 öğrenci oluşturmaktadır ve bilgisayar mühendisliği öğrencilerinin tutumlarının diğer bölümlere göre daha olumlu

olduğu sonucu elde edilmiştir. Ayrıca elde edilen sonuçta, öğrenim görülen üniversitelerin de öğrencilerin tutumları üzerinde etkili olduğu vurgulanmıştır.

Geva-May ve diğerleri (1993, akt. Kormaz ve Altun, 2013) tarafından yapılan bir çalışmada öğrencilerin bilgisayar programlamaya karşı tutumlarının orta seviyede ($X = 64,6$) olduğu sonucunu, Nurazian ve diğerleri (2007) ise 126 üniversite öğrencisi ile birlikte yaptıkları araştırmada, öğrencilerin bilgisayar programlama öğrenmeye yönelik pozitif tutumlarının negatif tutumlarından daha yüksek olduğu sonucunu elde etmişlerdir.

Bennedsen (2003) tarafından yapılan bir çalışmada ise programlama öğreniminin önündeki engelin motivasyon problemi olmadığı, öğrencilerin programlamaya dönük tutumlarının oldukça pozitif olduğu fakat esas problemin programlamanın öğrencilerin beklediğinden daha zor bir süreç olduğu ifade edilmektedir.

Özyurt ve Özyurt'un (2015) toplam 325 yükseköğrenim öğrencisi ile yaptıkları çalışmada, öğrencilerin programlamaya karşı tutumlarının olumlu olduğu görülmüştür. Bununla birlikte bu tutumun daha da olumlu hale getirilmesi için programlama derslerinde kullanılan öğretim teknik ve yöntemlerinin gözden geçirilmesi, derslerde öğrencilerin problem çözme ve eleştirel düşünme becerilerini geliştirecek etkinliklere yer verilmesi önerilmektedir.

Wang ve diğerleri (2017), öğrencilerin bilgisayar programlama kavramlarını ve becerilerini geliştirmek için çevrimiçi akran değerlendirme temelli bir sistem geliştirmiştir. Bu sistemde öğrenciler, akranlarına yorum yapıp, öğrenme etkinliği sırasında akranlardan gelen geribildirimleri ve puanları gözden geçirmektedir. Geliştirilmiş sistemin etkilerini incelemek için Güney Tayvan'da bulunan bir ortaokulun dört sınıfı üzerinde bir deney yapılmıştır. Toplam öğrenci sayısı 166 kişiden oluşmakta ve tüm öğrenciler 9. Sınıfta öğrenim görmektedir. Deney grubuna iki sınıf öğrencisi tahsis edilmiş ve bu gruba çevrimiçi akran değerlendirme temelli öğretim stratejisi ile öğrenim gerçekleştirilmiştir. Diğer iki sınıf ise geleneksel öğretim stratejisiyle öğrenen kontrol grubunu oluşturmuştur. Deneysel sonuçlar, deney grubundaki öğrencilerin kontrol grubundakilere göre daha fazla bilgi ve beceri geliştirmenin yanı sıra daha olumlu öğrenme tutumları sergilediklerini ortaya çıkarmıştır.

Alanyazında yapılan bu çalışmalar, düzenlenen öğrenme etkinliklerinin öğrenci tutumlarını nasıl etkilediğini göstermiştir. Yapılan çalışmalarda

öğrencilerin ilgi ve isteklerine hitap edebilecek etkinlikler belirlenmiş ve sonuçlarında tutuma yönelik değişimler ortaya konmuştur. Çalışmalarda etkinlik süresince öğrencilerin daha fazla uygulama yapmasına imkân verilmiş ve bu durumun tutumda olumlu etkiler yarattığı gözlemlenmiştir. Bu çalışmada da belirlenen etkinlikler sonucunda öğrenci tutumlarındaki değişim sunulacaktır.

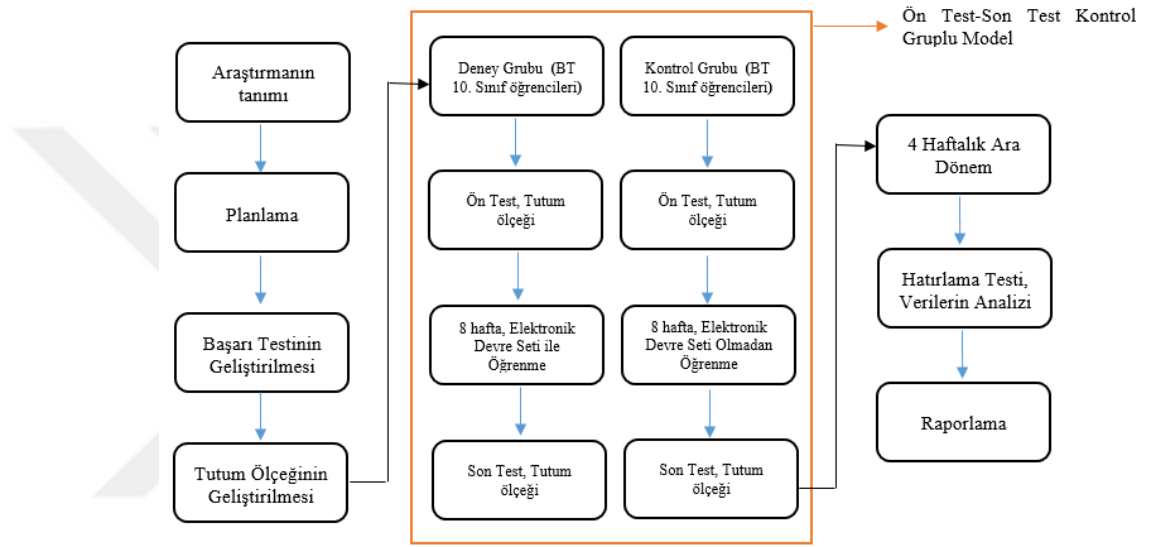


BÖLÜM III

YÖNTEM

Bu bölümde; araştırmanın tasarımı ve araştırma sorularına cevap bulmak için izlenen adımlar ele alınmıştır. Bu kapsamda, araştırmanın modeli, çalışma grubu, araştırma süreci, kullanılan veri toplama araçları ve toplanan verilerin analizinde kullanılan istatistiksel yöntem ve teknikler açıklanmıştır. Şekil 5' te tüm çalışma süreci verilmiştir.

Şekil 5.Çalışma süreci



3.1. Araştırmanın Modeli

Bu çalışma iki farklı ilde bulunan mesleki ve teknik Anadolu lisesi bilişim teknolojileri alanı 10. sınıf öğrencileri üzerinde, Programlama Temelleri dersi kapsamında gerçekleştirilmiştir. Yapılan çalışmada Arduino ile programlama dili öğretilmesi kapsamında dersin hem teori hem de uygulama bölümleriyle bütünleştirilmesinin başarıyı ve bilgisayar programlama yönelik tutumu nasıl etkilediğinin belirlenmesi amaçlanmıştır. Bu amaçla çalışmada araştırma alanı olarak da adlandırılan kontrol gruplu ön-test/son-test yarı deneysel model kullanılmıştır.

3.1.1. Araştırmanın Deneysel Modeli

Bu çalışmada yarı deneysel desenden yararlanılmıştır. Bu model ile nicel verilerin toplanmasında, değişkenler arasındaki neden-sonuç ilişkilerini ortaya çıkarmak için gözlenmek istenen veriler üretilmektedir (Büyüköztürk, 2001; Karasar, 2004). Bir deney grubunun bir de kontrol grubunun bulunduğu

çalışmada, yarı deneysel desenlerden kontrol gruplu ön-test son-test modeli kullanılmıştır. Bu modelde tek bir grup üzerinde uygulanan çalışma ile deneysel işlemin etkisi test edilmektedir. Çalışmaya katılan öğrencilerin bağımlı değişkene bağlı ölçümleri ön test yardımı ile uygulama öncesinde, son-test ile de uygulama sonrasında aynı ölçme araçları ile elde edilmektedir. Ön-test ve son-test uygulaması yapılan katılımcılar değişmemektedir. Kullanılan araştırma deseninde eşleştirme bulunmaması ve seçkisizlik olması sebebiyle tekrarlı ölçümler deseni olarak da isimlendirilir. Ayrıca bu desende sadece deney veya sadece kontrol grubuna ait ön-test ve son-test değerleri arasındaki farkın (D_1-D_2) anlamlılık boyutu da test edilmektedir (Büyüköztürk vd., 2008). Buna göre tablo 4'te araştırmanın akademik başarıya ilişkin deneysel desen tasarımı, tablo 5'te ise programlamaya yönelik tutuma ilişkin deneysel desen tasarımı verilmiştir.

Tablo 4. *Araştırmanın Akademik Başarıya İlişkin Deneysel Desen Tasarımı*

Gruplar	Ön test	İşlem	Son test	Ara dönem	İzleme Testi
Deney Grubu	D_1	Z	D_2	4 Hafta	D_3
Kontrol Grubu	D_1		D_2		D_3

Tablo 4'te gruplar, işlem yapılan grupları; D_1 , gruplardan alınan ön ölçümü, Z, deneysel işlemi, D_2 gruplardan alınan son ölçümü, D_3 ise 4 hafta ara dönem sonunda gruplardan alınan izleme testinin ölçümünü göstermektedir (Büyüköztürk vd., 2008).

Tablo 5. *Araştırmanın Programlamaya Yönelik Tutuma İlişkin Deneysel Desen Tasarımı*

Gruplar	Tutum Ölçeği	İşlem	Tutum Ölçeği
Deney Grubu	D_4	Z	D_5
Kontrol Grubu	D_4		D_5

Tablo 5'te gruplar, işlem yapılan grupları, D_4 , gruplardan alınan ön ölçümü, Z, deneysel işlemi, D_5 ise gruplardan alınan son ölçümü göstermektedir (Büyüköztürk vd., 2008).

Kontrol gruplu ön-test son-test yarı deneysel desenin kullanıldığı araştırmada her iki gruba önce ön test ve tutum ölçeği uygulanarak, öğrencilerin

programlama ile ilgili ön bilgileri ile programlamaya yönelik tutumları ölçülmüştür. Uygulama yapıldıktan sonra ise çalışmada Arduino'nun Programlama Temelleri dersine dâhil edilmesinin öğrencilerin akademik başarılarına ve programlamaya yönelik tutumlarına olan etkisini karşılaştırabilmek amacıyla son test ve tutum ölçeği uygulanarak başarılar ve ölçek puanları arasında anlamlı bir farkın olup olmadığı tespit edilmiştir. Yapılan deneysel işlemden sonra 4 hafta elektronik devre setleri kullanılmadan programlama eğitimi verilmiş ve bu süreç sonunda her iki gruba izleme testi uygulanarak konuların hatırlanma ve öğrenilen programlama becerilerinin devamlılık seviyesi ölçülmüştür. Tablo 6'da araştırma deseni verilmiştir.

Tablo 6. *Araştırma Deseni*

Deneysel İşlem Öncesi	
Etkinlik	Süre
Akademik Başarı Ön Testi	1 Ders Saati
Programlamaya Yönelik Tutum Ölçeği	1 Ders Saati
Arduino Devre Kartının Tanıtımı ve Kullanımı	2 Ders Saati
Toplam	4 Ders Saati
Deneysel İşlem	
Etkinlik	Süre
Uygulama 1	4 Ders Saati
Uygulama 2	4 Ders Saati
Uygulama 3	4 Ders Saati
Uygulama 4	4 Ders Saati
Uygulama 5	4 Ders Saati
Uygulama 6	4 Ders Saati
Uygulama 7	4 Ders Saati
Uygulama 8	4 Ders Saati
Toplam	32 Ders Saati
Deneysel İşlem Sonrası	

Etkinlik	Süre
Akademik Başarı Son Testi	1 Ders Saati
Programlamaya Yönelik Tutum Ölçeği	1 Ders Saati
Toplam	2 Ders Saati
4 Haftalık Deneysel İşlem Olmadan Öğretim	
Akademik Başarı İzleme Testi	1 Ders Saati

3.2. Çalışma Grubu

Çalışma gruplarından deney grubu, araştırmacının görev yaptığı, Manisa ilinde Mesleki ve Teknik Anadolu Lisesi Bilişim Teknolojileri Alanı'nda öğrenim gören ve Programlama Temelleri dersini alan 14 erkek ve 4 kız olmak üzere 18 öğrenciden oluşmaktadır.

Kontrol grubu ise Balıkesir ilinde Mesleki ve Teknik Anadolu Lisesi Bilişim Teknolojileri Alanı'nda öğrenim gören ve Programlama Temelleri dersini alan 20 erkek öğrenciden oluşmaktadır. Kontrol grubunun seçiminde iki grup öğrencilerinin birbirleri ile iletişimini minimumda tutmak amacıyla, araştırma sonuçlarının etkileyecek herhangi bir bilgi aktarımı olmasını önlemek için, farklı ilçe veya ilde öğrenim gören öğrencilerin olmasına karar verilmiştir. Ayrıca kontrol grubuna öğretim gerçekleştirecek öğretmenin, araştırmacının kendisi olamadığı için, araştırmacı ile aynı süre öğretmenlik deneyimine sahip ve yüksek lisans mezunu olması dolayısıyla da bilimsel araştırma süreçleri hakkında bilgi sahibi olan ve bu süreçlerdeki veri toplama gibi işlemlerde etik değerlere özen gösterecek niteliğe sahip olmasına dikkat edilmiştir. Bunun yanısıra her iki okuldaki bilişim teknolojileri alanlarının gerek atölyelerin fiziksel özellikleri bakımından gerekse araç gereç olarak aynı imkânlarla sahip olması da kontrol grubu öğrencilerinin belirlenmesinde etkili olmuştur.

Tablo 7. *Araştırma Grupları*

Gruplar	İşlem	Frekans
Deney Grubu	Elektronik devre setleri destekli öğrenim	18
Kontrol Grubu	Elektronik devre setleri desteği olmadan öğrenim	20

3.3. Araştırma Süreci

Bir öğrencinin, programın davranışını çıktısıyla ilişkilendirmek için programın nasıl yürüdüğünü anlaması gerekir. Bunun için bir tür fiziksel hesaplama modülleri kullanılarak fiziksel ortamda programlama yapmanın acemi bir programcı için faydalı olduğu belirtilir (Tan, Venema, & Gonzalez, 2017). Cihazın davranışı (örneğin; yüksek ses tonu veya motor hareketi), programcıya daha iyi görsel ve işitsel geri bildirim sağlayabilir. Bu yüzden bu çalışmada, düşük maliyetli gömülü bir platform olan Arduino Uno ve C# programlama dilinin birlikte kullanımına dayalı ders içeriği gerçekleştirilmiştir.

Bu çalışmada, Arduino Uno kullanılarak öğrencilerin günlük hayatta kullanabilecekleri nesnelere bütünleşik uygulamalar geliştirmelerini ve somut çıktı üretmeye odaklanmalarını sağlamak için mesleki ve teknik eğitim alan bilişim teknolojileri 10. sınıf Programlama Temelleri dersinde uygulanmak üzere 8 haftalık ders içeriği geliştirilmiştir. Bunun sonucu olarak da öğrencilerin, gerçek dünya problemlerine yaratıcı çözümler getirmeleri ve işbirlikli öğrenme ortamı sayesinde düşüncelerini özgür ve rahat bir şekilde ifade ederek araştırma meraklarının, programlama öğrenimine karşı ilgi ve motivasyonun artırılması hedeflenmektedir.

Bir içeriğin sunulma sürecinin tasarımı kişilerin öğrenme sürecini kolaylaştırmakta, eğitimin etkisini ve niteliğini arttırmaktadır (Çakır, Calp ve Doğan, 2015). Bu yüzden öğrenme ve öğretim teorilerine dayalı, öğrenenlerin analizi, hedef, içerik, yöntem ve son olarak da değerlendirme basamaklarını içeren sistematik bir yöntem olan öğretim tasarımı (Özdemir ve Uyangör, 2011), içerik geliştirme çalışmalarında önemli bir yerdedir. Öğretim tasarımı planlı, amaçlı, sistemli olarak yürütülen öğrenme etkinliklerinin, öğretim materyallerinin değerlendirme metotlarının bir eğitim kurumu aracılığıyla gerçekleştirildiği yansıtıcı ve sistematik bir süreçtir (Dooley, 2005; Şimşek, 2011). Başka bir tanıma göre ise öğretim tasarımı, derslerin ve öğretim materyallerinin güvenilir ve tutarlı bir şekilde işlerlik kazanmasıdır. Böylece tasarımcılar daha verimli çalışmakta, daha çekici ve etkin öğrenme ortamları oluşturulmaktadır (Molenda, Reigeluth & Nelson, 2003).

Bir öğretim tasarımının temel yapısı “Analiz (Analyse)”, “Tasarım (Design)”, “Geliştirme (Development)”, “Uygulama (Implementation)” ve

“Değerlendirme (Evaluation)” olmak üzere beş evreden oluşmaktadır (Dooley, 2005; Şimşek, 2013; Zheng & Smaldino, 2003). Temel yapı olarak görülen bu model ADDIE öğretim tasarım modeli olarak adlandırılmaktadır. Bu çalışmada da programlama eğitimi için dersler ADDIE öğretim tasarımı modelinin analiz, tasarım, geliştirme, uygulama, değerlendirme basamaklarına göre düzenlemiştir.

3.3.1. Hazırlık Aşaması

Bu aşamada öncelikle öğrenen analizi yapılmış, öğretim amacı ve çalışma yapılacak programlama konuları belirlenmiştir. Belirlenen konuların hedeflerinin öğrencilere kazandırılması amacıyla da uygulama ağırlıklı bir öğretim programı tasarlanmıştır. Bu öğretim programında ise öğrencilerin eleştirel, yaratıcı düşünme ve problem çözme becerilerini, iletişim, araştırma ve sorgulama yeteneklerini geliştirip sergileyebilecekleri etkinlikler düzenlenmiştir. Daha sonra akademik başarı testi ve bilgisayar programlamaya karşı tutum ölçeği geliştirilmiştir. Deney ve kontrol grupları belirlenerek her iki ölçme aracı ön test olarak öğrencilere uygulanmıştır. Deney grubuna Arduino elektronik devre kartı ve kullanımı hakkında da detaylı bilgi verilmiştir.

Programlama konularının belirlenmesinde MEB müfredatı dâhilinde C# programlama dilinin kullanılmasına karar verilmiş ve Bloom (Bloom, 1956) taksonomisinin bilişsel düzey tasarım basamakları dikkate alınarak, öğrencilerin başarı durumlarını belirlemek için akademik başarı testi hazırlanmıştır (Ek 1). Bu aşamada yapılan diğer bir çalışma ise öğrencilerin programlamaya yönelik tutumlarını belirlemek için programlama tutum ölçeğinin geliştirilmesidir (Altay ve Kışla, 2018).

Çalışma yapılacak grupların seçiminde öğrenci sayıları, okul türü, ders öğretmeninin niteliği ve deney- kontrol grubu öğrencilerinin birbirlerini tanıma olasılıklarının olmaması gerektiği göz önünde bulundurulmuştur. Her iki grubun birbirine denk özelliklerde olmasına dikkat edilmiştir. Gruplar belirlendikten sonra deney grubu öğrencilerine 1 hafta (4 ders saati) boyunca Arduino Uno elektronik devre kartının tanıtımı ve kullanım şekli hakkında bilgi verilmiştir. Öğrenciler haftada 2 ders saati Temel elektronik ve ölçme dersi aldıkları ve bu derste elektronik devre elemanlarının ne olduğu ve kullanım şekillerini öğrendikleri ve breadboard üzerine devre kurma işlemlerini uygulamalı olarak yaptıkları için bu konular çalışmanın hazırlık aşamasında hatırlatıcı bilgi olarak verilmiştir.

Tablo 8’de tasarlanan öğretim programına ait sekiz hafta boyunca uygulanacak olan etkinlikler verilmiştir.

Tablo 8. *Sekiz Haftalık Uygulama Süreci*

Kazanımlar	Süre	Uygulamalar
1. Değişken nedir? Nasıl kullanılır? Sorularını cevaplayabilmek 2. Değişkenleri İsimlendirme Kuralları 3. Veri tiplerini tanımlayabilmek 4. Giriş - Çıkış İşlemleri	4 Ders Saati	3 Adet Led ile trafik lambası yapımı Bir metin ifadesini LCD ekrana yazdırma
1. Aritmetiksel operatörleri kullanabilmek 2. İlişkisel operatörleri kullanabilmek 3. Mantıksal operatörleri kullanabilmek 4. İşlem önceliğine göre problem çözebilmek	4 Ders Saati	Klavyeden girilen sayı ve operatöre göre işlem yapılıp sonucu LCD ekranda yazdırma. Seçilen her operatör için ayrı renkteki Led ışığını yakma. 5 adet Led ile yürüyen ışık devresinin yapımı
Karar kontrol deyimlerini kullanabilmek. (if yapısı)	4 Ders Saati	Ortamda bulunan ışık değerinin şiddetine göre LDR sensör kullanarak led ışığını yakma veya söndürme 5 Led ile hazırlanan Karaşimşek devresinin başlamasının klavye ile kontrolü ve yine klavye kontrolü ile devreyi durdurma.
Karar kontrol deyimlerini ve operatörleri birlikte kullanabilmek. (iç -içe if yapısı)	4 Ders Saati	10 Led ile hazırlanan Karaşimşek veya Johnson sayıcı devresinin başlaması için seçim yapma, yanlış değer seçilmesi halinde alarm devresinin çalışması ve klavye kontrolü ile çalışan devrenin durdurulması. Klavyeden girilen 1-7 arası sayıların ikilik sayı sistemindeki karşılığına göre led ışığını yakma
Switch-case yapısı	4 Ders Saati	Klavyeden girilen not değeri kadar göre buzzer ile ses üreten devre yapımı
For Döngüsü	4 Ders Saati	Kullanıcı adı ve şifre doğrusu ise 3 defa ileri geri, yanlışsa 3 defa flaşör şeklinde led yakan devre yapımı Girilen ismin harf sayısı kadar led yanacak ve harf sayısı kadar tekrar edecek devre yapımı
1. While Döngüsü 2. Do -while Döngüsü	4 Ders Saati	Karaşimşek ve Johnson sayıcı devrelerini döngü değerleri kadar çalıştıran devre yapımı Park sensörü yapımı ve mesafe sensöründen aldığımız uzaklığı LCD ekrana yazdıran uygulama
İç içe döngüler	4 Ders Saati	Şifreli bir kapının açılması için kapıda bulunan sensör, harekvdgıladıđı zaman şifre değeri isteyen ve kişiye 5 defa şifre girme hakkı veren, şifre doğru ise kapıyı açan, şifre yanlış ise alarm devresini çalıştıran devre yapımı.

3.3.2.Çalışma Konularının Belirlenmesi Ve Ders İşlenişi

Çalışma konuları Milli Eğitim Bakanlığının Programlama Temelleri ders müfredatına göre giriş çıkış kavramları, değişken ve sabit işlemleri, operatörler, koşul yapısı ve döngü yapısı kavramları olarak belirlenmiştir. Alanyazında yapılan çalışmalar da öğrencilerin programlama öğrenirken bu konularda zorluklar yaşadıklarını ortaya koymuştur (Çetin, 2012). Bu araştırmalarda öğrencilerin döngü yapısındaki döngü içinde kalan veya döngü dışında kalan gibi ifadeleri kavrayamadığı, kontrol yapıları ve değişkenlerinin döngü yapısı içinde aldıkları değerleri takip edemediklerini ortaya koymuştur (DuBoulay, 1986; Pea, 1986; Sleeman, Putnam, Baxter & Kuspa, 1986).

Ders tasarımındaki ders planı 8 hafta boyunca ve haftada 2 saat uygulama ve 2 saat teori olmak üzere 4 ders saatinden oluşmaktadır. Derslerin tasarlanmasında aşağıdaki programlama aşamaları izlenmiştir (MEGEP, 2011).

1. **Problemi Tanımlama:** Öncelikle çözülecek problem iyi anlaşılmalıdır. Bu adımda yapılacak en ufak bir hata daha sonraki adımların yeni baştan yapılmasını gerektirebilir. Problemin tanımı yapılırken var olan bilgiler, anlamları ve birbirleri ile ilişkileri tanımlanmalıdır. Daha sonra istenilenler belirlenmeli ve bunların var olan bilgiler ile ilişkileri öğrenilmelidir. Son olarak yapılacak işlemler belirlenmelidir.

2. **Problemi Geliştirme:** Problem tanımını yapıldıktan sonra çözüm için yol aramak gerekir. Genellikle bir problemin birden fazla çözüm yolu olabilir. Bunlardan en uygunu seçilmeye çalışılır. Problem ne kadar karışık olursa olsun, alt birimlere bölünür. Her birimin çözümü ayrı ayrı yapılır. Bu yapılırken birimler arası ilişki sürekli olarak korunur.

3. **Sisteme Uyumluluğunu Tespit Etme (Girdi-Çıktı Belirleme):** Programcı program çıktısı olarak almak istediği dökümün biçimini tasarlar. Bir döküm biçimi tasarlanırken anlaşılır ve kullanılabilir olmasına özen gösterilmelidir.

4. **Çözümü Kâğıt Üzerinde Gösterme (Prototip Oluşturma):** Bu adımda, sorunun çözüm basamakları, birbirleri ile ilişkileri ve bilgi akışı daha kolay görülebilir ve yanlışlıklar düzeltilebilir.

5. **Çözümü Deneme:** Program yazıldıktan sonra sonuçları daha önceden bilinen veriler girilerek eldeki sonuçlarla çıkan sonuçlar karşılaştırılır. Algoritmanın doğruluğu kontrol edilir.

6. **Çözümü Geliştirme:** Bu aşamada doğruluğu kontrol edilmiş çözümün tekrar edilen blokların daha kısa yoldan sonuca ulaşılması durumları gözden geçirilir.

Ders tasarımı bu aşamalar çerçevesinde ADDIE öğretim tasarım yöntemine göre düzenlenerek Şekil 6'daki ders tasarımı oluşturulmuştur (Altay ve Kışla, 2018).

Çalışmada her ders için içerik oluşturulurken,

- *Analiz basamağında*, hedef kitlenin özellikleri belirlenmiş, daha sonra bu özellikler dikkate alınarak eğitim ihtiyaçları ve problem durumu ortaya koyularak Programlama Temelleri dersinin “değişkenler ve sabitler, operatörler, karar yapıları ve döngüler” başlıkları seçilmiştir.

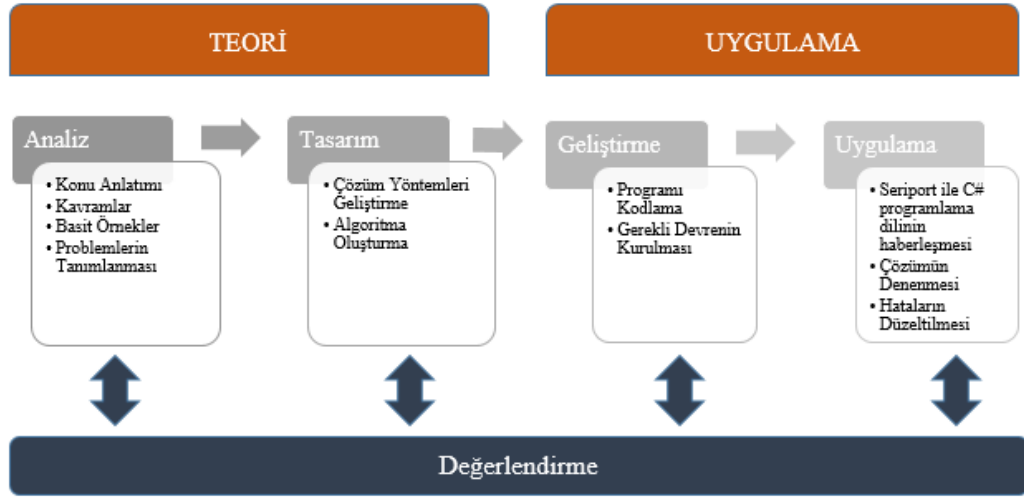
- *Tasarım basamağında*, analiz basamağından gelen veriler doğrultusunda programlama öğretiminin amaç ve hedefleri ile sunulacak içerikle beraber öğretim sürecinde kullanılacak yöntem ve araçlar belirlenmiştir. Bu aşamada MEB müfredatı dâhilinde C# programlama dilinin kullanılmasına ve Arduino ile yapılacak uygulamalar için gerekli devre elemanlarına karar verilmiştir. Problemler için çözüm yolları geliştirilmiş ve gerekli kodlama için algoritmalar üretilmiştir.

- *Geliştirme basamağında*, öğretim sürecinde kullanılacak materyaller tasarlanıp geliştirilmiştir. Algoritmalar kâğıt üzerinde gösterildikten sonra program kodlanmış ve çözüme yönelik devreler elektronik devre tahtası (breadboard) üzerine kurulmuştur.

- *Uygulama basamağında* ise, geliştirilen ürün uygulamaya konmuştur. Öğrenciler yazdıkları programı derleyip Arduino devre kartına yükledikten sonra seri port ile C# programlama dili arasında bir köprü kurulmuştur. Oluşan hataları gidermek için çözüm yolları üretilerek istenilen çıktı elde edilmiştir.

- Son olarak *değerlendirme basamağında*, belirlenen öğrenme hedefleri ve ihtiyaçları kontrol edilmiştir. Yapılan gözlemler ile süreç içinde sınırlılıklar belirlenmiş ve bir sonraki uygulamada bu sınırlılıklar giderilmeye çalışılmıştır.

Şekil 6. Haftalık Ders Tasarımı



3.3.3. Materyal Temini

Çalışmanın hedefleri ve konuları (giriş çıkış deyimleri, değişken ve sabitler, operatörler, kontrol ve döngü yapısı) belirlendikten sonra programlama öğrenmeyi kolaylaştıran, kolay temin edilen, kullanması ve öğrenmesi basit, düşük maliyetli materyaller seçilmesine özen gösterilmiştir. Bu materyallerin seçiminde de ilgi çekici olması, işbirlikli çalışmaya uygun, yaratıcı düşünmeyi ve sürekli öğrenmeyi desteklemesi belirleyici etkenler olmuştur (Arduino, 2019).

Bu çalışma Ege Üniversitesi tarafından desteklenen 17-EĞF-002 nolu Bilimsel Araştırma Projesidir. Çalışmanın tüm materyalleri proje desteğiyle alınmıştır.

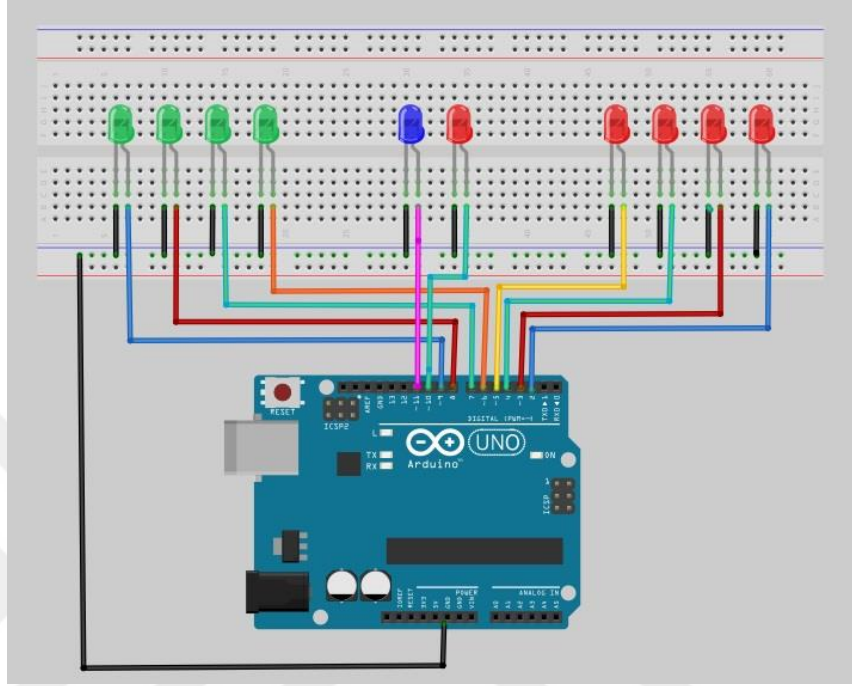
3.3.4. Uygulama Aşaması

Bu aşamada 8 Haftalık uygulama süreci boyunca haftada 4 saat olmak üzere toplam 32 ders saati için hazırlanan ders planı uygulanmıştır. Öncelikle hem deney hem de kontrol grubuna akademik başarı ön testi ile programlamaya yönelik tutum ölçeği uygulanmıştır. Deney grubuna 8 haftalık planlanan ders programı uygulanırken kontrol grubu geleneksel öğretime devam etmiştir. Çalışma konuları Milli Eğitim Bakanlığı müfredatında belirtilen C# programlama dili ile anlatılmıştır. Her iki grup öğrencileri de Programlama Temelleri dersini atölye ortamında almaktadır. Öncelikle öğrencilere konularla ilgili teorik bilgi verilmiş daha sonra uygulama aşamasına geçilmiştir. Kontrol grubunda öğretmen tarafından teorik bilgi verildikten sonra uygulama aşamasında göster yap tekniği kullanılmıştır. Deney grubunda öğrencilere araştırmacı tarafından teorik bilgi anlatıldıktan sonra araştırma kapsamında kendilerine verilen görevleri yerine

getirmiştiler. Oluşturulacak devre şemaları hazırlanarak, öğrencilere gösterilmiş ve atölyede işbirlikli çalışma ortamları oluşturulmuştur.

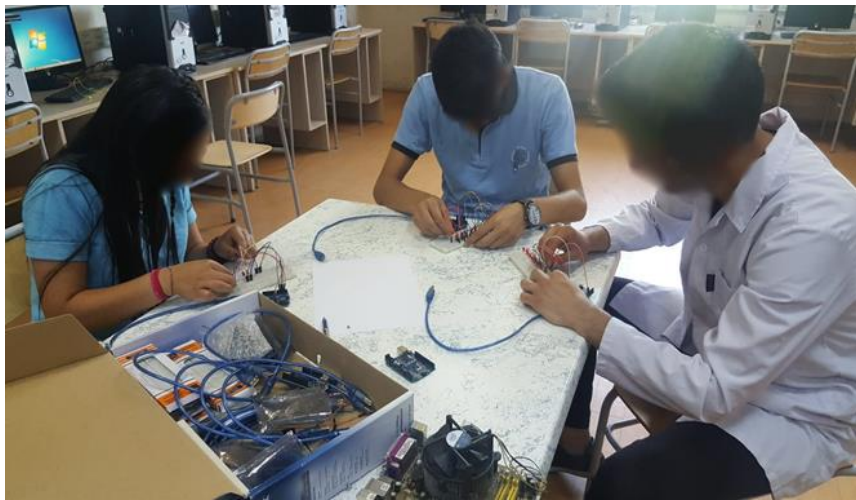
Şekil 7' de derste kullanılan devre şemalarından biri yer almaktadır.

Şekil 7. Devre Şeması Örneği



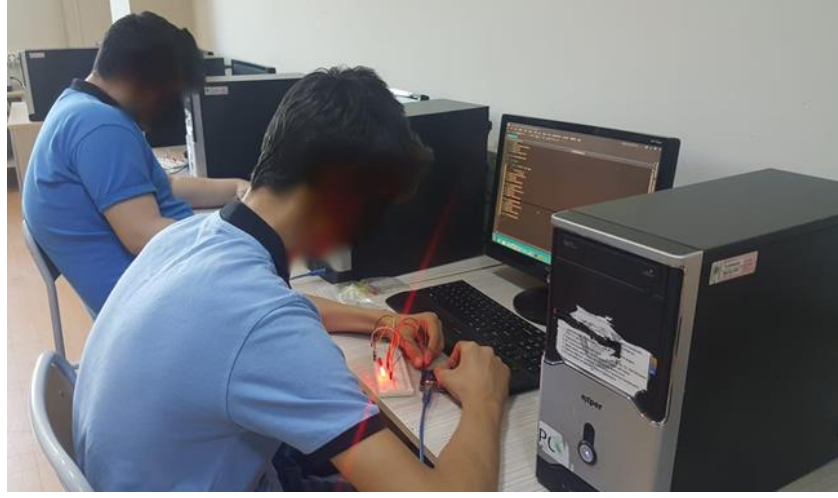
Bu aşamada öğrenciler gerekli fiziksel malzemeleri bir araya getirerek işbirlikli olarak devreleri oluştururlar.

Şekil 8. Öğrenenlerin İşbirlikli Öğrenme Yaklaşımı İle Devre Tasarımları



Oluşturulan devrelerin istenilen şekilde çalışabilmesi için bireysel olarak, C# programlama dili ile kodlama işlemi yapılmıştır.

Şekil 9. Öğrenenlerin Bireysel Kodlama Ortamı



Şekil 10. Ders Sırasında Yazılan Bir Kodlama Örneği

```
static void Main(string[] args)
{
    SerialPort port = new SerialPort("COM9", 9600);
    port.Open();
    Console.WriteLine("1-4 arası bir sayı giriniz");
    int cevap = Convert.ToInt16(Console.ReadLine());
    switch (cevap)
    {
        case 1:
            port.Write("a");
            break;

        case 2:
            port.Write("b");
            break;

        case 3:
            port.Write("c");
            break;

        case 4:
            port.Write("d");
            break;

        default:
            port.Write("e");
            break;
    }
    port.Close();
    Console.ReadKey();
}
```

8 hafta sonunda ise ön testleri uygulanan akademik başarı testinin ve programlamaya yönelik tutum ölçeğinin son testleri uygulanmıştır. Gerçekleştirilen deneysel işlemin akademik başarıdaki kalıcılığına etkisini görmek amacı ile her iki grup öğrencileri 4 haftalık elektronik devre setleri

olmadan bir öğretim süreci geçirdikten sonra, öğrencilere izleme testi uygulanmıştır. Son olarak tüm testlerin analizleri gerçekleştirilmiştir.

3.4. Veri Toplama Araçları

Çalışmanın verilerini elde etmek için nicel veri toplama araçları kullanılmıştır. Nicel veri toplama aracı olarak; programlamaya yönelik tutum ölçeği ve akademik başarı testi kullanılmıştır.

3.4.1. Akademik Başarı Testi

Bu çalışmanın amacı, Mesleki ve Teknik Eğitim Ortaöğretim Programı Bilişim Teknolojileri Alanı 10. Sınıf Programlama temelleri dersi için geçerli ve güvenilir bir başarı testi geliştirmektir. İlk olarak Mesleki ve Teknik Anadolu Lisesi Bilişim Teknolojileri Alanı 10. Sınıf Programlama Temelleri dersi öğretim programı dikkate alınarak değişkenler ve sabitler, operatörler, karar yapıları ve döngüler konularında kazanımlar belirlenmiş ve Bloom Taksonomisinin bilişsel düzey basamaklarına göre bu kazanımlara uygun 75 adet soru hazırlanmıştır.

Testin kapsam geçerliliği için 3 uzmandan görüş alınmış ve bu görüşler doğrultusunda, aynı kazanımın birden fazla soru ile ölçülmesi, bazı soruların kazanım ile örtüşmemesi gibi sebeplerden dolayı testteki bazı sorular düzeltilerek, bazı sorular da çıkartılarak soru sayısı 36'ya düşürülmüştür. Oluşturulan bu test, Mesleki Ve Teknik Anadolu Lisesi Bilişim Teknolojileri Alanı 11. Sınıfta öğrenim gören 145 öğrenciye uygulanmıştır. Test sonuçlarına göre her bir maddenin ayırt edicilik ve güçlük indeksleri ayrı ayrı hesaplanmıştır.

Madde analizi için bütün öğrencilerin cevap kâğıtları puanlanmış ve en yüksek puanlıdan başlanarak en düşük puanlıya doğru sıralanmıştır. En yüksek puanlıdan başlanarak %27'lik üst grup, en düşük puanlıdan başlanarak %27'lik alt grup olmak üzere iki grup belirlenmiştir. Arada kalan diğer cevap kâğıtları madde analizinde dikkate alınmamıştır. Bu durumda 38 öğrenci üst grupta, 37 öğrenci alt grupta yer almıştır. Bütün sorular için “madde güçlük indeksi (P)” ve “madde ayırt edicilik indeksi (D)” hesaplanmıştır. Madde güçlük indeksi (P), her bir maddenin doğru cevaplanma oranını göstermektedir ve “0” ile “1” arasında değerler alabilmektedir. Bulunan değer sıfıra yaklaştıkça maddenin zor olduğu, bire yaklaştıkça maddenin kolay olduğu söylenebilir. Genellikle madde güçlük indeksinin başarı testlerinde .50 civarında olması arzu edilir. Orta zorluk düzeyindeki maddelerden oluşan testlerin güvenilirlik düzeyleri daha yüksek bulunmaktadır (Gönen, Kocakaya ve Kocakaya, 2011). Madde ayırt edicilik

indeksi (D) ise, bir maddenin başarı düzeyi yüksek öğrencilerle başarı düzeyi düşük öğrencileri ayırt etme derecesidir. Madde ayırt edicilik indeksi “-1” ile “+1” arasında değerler alabilmektedir. Madde ayırt edicilik indeksinin 0’ a yaklaşması, maddenin üst ve alt grubu ayırt ediciliğinin düşük, +1’ e yaklaşması ayırt ediciliğinin yüksek olması demektir. Madde analizi sonucunda ayırt edicilik kriterini değerlendirirken şu hususlara dikkat edilir (Turgut, 1992, akt. Ayvacı ve Durmuş, 2015):

$\underline{D} \geq 0,40$: Madde çok iyi, düzeltilmesi gerekmez.

$0,40 > \underline{D} \geq 0,30$: Madde iyi, yine de geliştirilebilir.

$0,30 > \underline{D} \geq 0,20$: Maddenin düzeltilmesi ve geliştirilmesi gerekir.

$\underline{D} < 0,20$: Madde kullanılmamalıdır

$\underline{D} \leq 0$: Maddeler teste dâhil edilemez.

Madde ayırt edicilik ve madde güçlük indeksleri aşağıdaki formüller ile hesaplanmaktadır.

$$P = \frac{(Dü+Da)}{(Nü+Na)} \quad D = \frac{(Dü-Da)}{(Nü \text{ veya } Na)}$$

D: Madde ayırt edicilik indeksi

P: Madde güçlük indeksi

Dü: Maddeyi üst grupta doğru cevaplayan öğrencilerin sayısı

Da: Maddeyi alt grupta doğru cevaplayan öğrencilerin sayısı

Nü: Üst gruptaki öğrencilerin mevcudu

Na: Alt gruptaki öğrencilerin mevcudu

Tablo 9. Başarı Testinin Madde Güçlük Ve Ayırt Edicilik İndeksleri

Soru	Gruplar	Seçenekler				P Ve D Değerleri
		A	B	C	D	
S1	Üst Grup	0	38	0	0	P= 0,85
	Alt Grup	6	26	4	3	D= 0,32
S2	Üst Grup	0	0	0	38	P= 0,72
	Alt Grup	5	6	12	16	D= 0,59
S3	Üst Grup	0	31	0	7	P= 0,63
	Alt Grup	5	16	8	10	D= 0,41
S4	Üst Grup	25	9	1	3	P= 0,49
	Alt Grup	12	6	9	12	D= 0,35
S5	Üst Grup	1	3	34	0	P= 0,55
	Alt Grup	8	12	7	12	D= 0,73
S6	Üst Grup	25	0	3	10	P= 0,45
	Alt Grup	9	4	21	5	D= 0,43
S7	Üst Grup	2	23	12	1	P= 0,47
	Alt Grup	9	12	10	8	D= 0,30
S8	Üst Grup	4	33	0	1	P= 0,52

	Alt Grup	14	6	15	4	D=	0,73
S9	Üst Grup	0	0	36	2	P=	0,67
	Alt Grup	12	10	14	3	D=	0,59
S10	Üst Grup	5	0	32	1	P=	0,55
	Alt Grup	7	15	9	8	D=	0,62
S11	Üst Grup	3	31	4	0	P=	0,48
	Alt Grup	11	5	12	11	D=	0,70
S12	Üst Grup	1	0	0	37	P=	0,59
	Alt Grup	9	12	11	7	D=	0,81
S13	Üst Grup	25	1	11	1	P=	0,45
	Alt Grup	9	11	11	8	D=	0,43
S14	Üst Grup	3	5	1	29	P=	0,43
	Alt Grup	9	17	10	3	D=	0,70
S15	Üst Grup	5	1	0	32	P=	0,61
	Alt Grup	8	8	9	14	D=	0,49
S16	Üst Grup	1	7	25	5	P=	0,51
	Alt Grup	5	11	13	10	D=	0,32
S17	Üst Grup	0	1	35	2	P=	0,53
	Alt Grup	10	14	5	10	D=	0,81
S18	Üst Grup	10	11	6	11	P=	0,23
	Alt Grup	7	11	12	9	D=	0,08
S19	Üst Grup	4	0	33	1	P=	0,61
	Alt Grup	7	11	13	8	D=	0,54
S20	Üst Grup	28	8	1	1	P=	0,55
	Alt Grup	13	7	13	6	D=	0,41
S21	Üst Grup	33	5	0	0	P=	0,59
	Alt Grup	11	8	13	7	D=	0,59
S22	Üst Grup	9	22	4	3	P=	0,36
	Alt Grup	6	5	12	16	D=	0,46
S23	Üst Grup	1	1	35	1	P=	0,63
	Alt Grup	4	13	12	10	D=	0,62
S24	Üst Grup	0	0	0	38	P=	0,59
	Alt Grup	12	8	13	6	D=	0,86
S25	Üst Grup	35	2	0	1	P=	0,57
	Alt Grup	8	14	13	4	D=	0,73
S26	Üst Grup	2	0	32	4	P=	0,51
	Alt Grup	7	16	6	10	D=	0,70
S27	Üst Grup	0	1	1	36	P=	0,49
	Alt Grup	16	13	9	1	D=	0,95
S28	Üst Grup	29	5	3	1	P=	0,45
	Alt Grup	5	15	12	7	D=	0,65
S29	Üst Grup	21	15	2	0	P=	0,27
	Alt Grup	6	5	18	10	D=	0,27
S30	Üst Grup	30	5	2	1	P=	0,59
	Alt Grup	14	11	7	7	D=	0,43
S31	Üst Grup	36	1	1	0	P=	0,65
	Alt Grup	13	10	10	6	D=	0,62
S32	Üst Grup	34	1	1	2	P=	0,61
	Alt Grup	12	8	9	10	D=	0,59
S33	Üst Grup	21	4	13	0	P=	0,41
	Alt Grup	10	10	10	9	D=	0,30
S34	Üst Grup	35	0	2	1	P=	0,56
	Alt Grup	7	12	12	8	D=	0,76

S35	Üst Grup	22	4	6	6	P=	0,32
	Alt Grup	2	12	10	15	D=	0,54
S36	Üst Grup	27	5	5	1	P=	0,41
	Alt Grup	4	10	14	11	D=	0,62

Madde analizi sonucunda 16 madde testten çıkarılmış ve 20 adet çoktan seçmeli sorulardan oluşan Programlama Temelleri Başarı Testi oluşturulmuştur. Geliştirilen testte her soru 5 puan üzerinden değerlendirilmekte ve testten alınacak maksimum puan 100 olarak hesaplanmaktadır. Buna göre testin ortalama madde güçlüğü .53, ortalama madde ayırt edicilik indeksi ise .66 olarak elde edilmiştir.

Sonuç olarak, hem müfredata uygun hem de diğer bilimsel çalışmalarda öğretmenlerin bu çalışmada yer alan konularda sınıf içi değerlendirme süreçlerinde kullanabilecekleri geçerli ve güvenilir bir ölçüm aracı geliştirilmiştir.

3.4.2. Programlamaya Yönelik Tutum Ölçeği

Yapılan alanyazın incelemesi sonucunda birçok programlamaya yönelik tutum ölçeği çalışmasının olduğu görülmüştür (Çetin ve Özden, 2017; Korkmaz ve Altun, 2013; Özyurt ve Özyurt, 2015), fakat yapılan çalışmaların tümünde örneklem üniversite öğrencileridir. Şimşek'e (2002) göre öğrencilerin bir öğrenme ortamında sergiledikleri tutumda öğrencilerin yaşı ve öğrenim düzeyi gibi değişkenler önemli etmenlerdir. Bu sebeple bu çalışmada lise öğrencilerine yönelik bir ölçme aracı geliştirilmiştir (Altay ve Kışla, 2018).

Çalışmada ilk olarak problem durumu belirlenmiş ve programlama içeriğini kapsayacak şekilde 64 maddeden oluşan bir soru havuzu hazırlanmıştır. Daha sonra ilgili alanlarda dört uzmandan görüş alınarak tüm görüşler tek bir formda birleştirilmiştir. Maddelerin her biri için kapsam geçerliliği hesaplanmış ve geçerlik oranı .80 altında çıkan tüm maddeler atılmıştır. Bu işlemde sonra oluşan 21 maddelik form 214 öğrenci tarafından doldurulmuştur. Toplanan veriler analiz edildikten sonra ölçeğin Cronbach Alfa iç tutarlık katsayısı .80 olarak hesaplanmıştır.

Ölçeğin yapı geçerliği için açıklayıcı faktör analizi (AFA) yapılmıştır. Fakat yapı geçerliliğinin belirlenmesinde faktör analizi tek başına yeterli olmadığı (Demir ve Yurdugül, 2014) için Fornell ve Larcker'ın (1981) geliştirdiği iraksak ve yakınsak geçerliği belirlemek amacıyla daha basit çözümlenebilen bir sına yöntemi kullanılmıştır. Bu yöntem ile yapı geçerliği AVE değerleri aracılığıyla ve çıkan yapının doğruluğunun ölçülmesi için paylaşılan varyans değerleri DFA

aracılığıyla elde edilmiştir. Analiz sonucunda faktörlerin yapısal güvenirliği .81 ile .97, faktörlerin AVE değerleri .59 ile .68, faktörlerin AVE değerlerinin karekökleri ise .83 ile .77 arasında bulunmuştur. Böylece analiz sonuçları geliştirilen ölçeğin yapı geçerliğini ve yapısal güvenirliğini destekler niteliktedir. Oluşan üç faktörlü ölçekte birinci faktör *Programlamaya Yönelik İstek*, ikinci faktör *Programlamanın Yararına Yönelik İnanç*, üçüncü faktör ise *Programlamaya Yönelik İlgi* olarak isimlendirilmiştir.

3.5.Verilerin Analizi

Ön test-son test-kontrol gruplu yarı deneysel tasarım modeli (Campbell & Stanley, 1963) günümüzde tercih edilen araştırma tasarımlarından biridir. Bu tasarımın araştırmacılar tarafından neden bu kadar sık kullanıldığını açıklayan iki sebep vardır. Birincisi bu tasarım modeli çalışmayı, iç geçerliliğe karşı tehditler üzerinde sıkı bir bilimsel kontrol sağlayarak “gerçek bir deneysel tasarım” olarak sınıflandırılmasını sağlar. İkincisi ise, çok yönlü bir tasarım modeli olmasıdır. Bu model genellikle sadece iki grubu (deney ve kontrol grubu) karşılaştırmak için kullanılsa da ek karşılaştırma gruplarına uyması için kolayca genişletilebilir (Huck, & McLean, 1975).

Bu tasarım modeli tarafından sağlanan verilerin istatistiksel analizine alınabilecek birkaç yaklaşım vardır. Bazı araştırmacılar kovaryans analizini kullanırken, diğerleri her konu için bir kazanç (değişim) puanı hesaplar ve ardından grupları ortalama kazanç puanları açısından karşılaştırır. Fakat eğitim alanında yayınlanmış olan alanyazın, birçok araştırmacının kovaryansı veya puan kazanma yaklaşımını atladığını, bunun yerine verilerini tekrarlı ölçülen varyans analizine (ANOVA) maruz bıraktığını göstermektedir (Huck & McLean,1975).

Bu çalışmada verilerin analizi için IBM SPSS 24 programı kullanılmıştır.

3.6.Araştırmacının Rolü

Deneysel işlem bilişim teknolojileri alanı 10. sınıf Programlama Temelleri dersinde 18 öğrenci ile yapılmış ve araştırmacı bu sürecin içerisinde dersin hem teorik hem de uygulamalı kısmında rol almıştır. Hem kontrol grubu öğretmeninin hem de araştırmacının 10 yıllık öğretmenlik deneyimi olması sebebiyle programlama yeterliliğine sahiptirler.

3.7.Çalışmanın Güvenirliđi ve Geçerlilik

Bir araştırmanın güvenirliđi ve geçerliliđi arařtırmaların deđerini belirten en önemli unsurdur (Büyüköztürk, Çakmak, Akgün, Karadeniz ve Demirel, 2008). Burada bahsedilen güvenirlik ve geçerlilik hem arařtırma sürecini hem de arařtırmada kullanılan veri toplama araçlarını kapsamaktadır. Çalışmada kullanılacak veri toplama araçlarının geçerlilik ve güvenirliđi için gerekli önlemler alınmış ve analizler yapılmıştır. Nicel veri toplama araçları ile yapılan çalışmalarda ise güvenirlik, elde edilen sonuçların genellenebilirliđiyle, geçerlilik ise hatasız elde edilen sonuçlar ile ilişkili olmaktadır (McMillan & Schumacher, 2010). Buna göre bu çalışmada alınan güvenirlik ve geçerlilik önlemleri řu şekildedir.

Güvenirlik;

- Uygun veri toplama araçları kullanılmıştır.
- Uygulama süresi uzun tutulmuştur (8 Hafta).

Geçerlilik;

- Çalışmanın amacı detaylı olarak açıklanmıştır.
- Seçilen örneklem grubunun seçim şekli ve grupların özellikleri detaylı olarak açıklanmıştır.
- Uygulama süreci detaylı olarak açıklanmıştır.
- Çalışmanın sınırlılıkları ve varsayımları belirtilmiştir.
- Arařtırmacının rolü detaylı olarak açıklanmıştır.

BÖLÜM IV

BULGULAR

Araştırmanın bu aşamasında, yöntem bölümünde belirtilen istatistiksel analizlerle elde edilen verilere ve bu verilerin yorumlanmasına yer verilmiştir.

4.1.Akademik Başarı Testi ANOVA Analizleri

Deney ve kontrol grubu öğrencilerine 3 farklı zamanda akademik başarı testi uygulanmıştır. Tablo 10'a göre deneysel işlem öncesi uygulanan test *ön test*, deneysel işlem sonrası uygulanan test *son test* ve kalıcılığı ölçmek için 4 hafta ara verildikten sonra uygulanan test *izleme testi* olarak adlandırılmıştır.

Tablo 10. Akademik Başarı Testi Uygulama İsimleri

Zaman	Bağımlı Değişkenler
1	Ön test
2	Son test
3	İzleme testi

Tablo 11'de deney ve kontrol gruplarının frekansları verilmiştir. İki faktörlü tekrarlı ANOVA ölçümlerinde gruplar arası faktörünün her bir düzeyi için bağımlı değişkene ait puanlar, evrende normal bir dağılım gösterir. Eğer ilişkili örneklem büyüklüğü en az 15 kişiden oluşuyorsa bu varsayım sağlanmış demektir. Araştırmada kontrol grubu 20, deney grubu ise 18 kişiden oluştuğu için bu varsayım gerçekleşmiştir.

Tablo 11. Grup Frekansları

Gruplar	Frekans
Deney	18
Kontrol	20

Tablo 12.'de her iki grup için de test ortalamaları verilmiştir. Tabloya göre programlama temelleri dersinde elektronik devre setleri (Arduino) kullanılmadan yapılan öğretim yöntemi ile kontrol grubunun ön test ortalaması 30, son test ortalaması 28.50 ve izleme testi ortalaması ise 27.75 olarak elde edilmiştir. Dersin Arduino kullanılarak anlatıldığı deney grubunun ise ön test ortalaması 39.17, son test ortalaması 61.94 ve izleme testi ortalaması 61.11 olarak elde edilmiştir.

Tablo 12. *Test Ortalamaları*

	Grup	Ortalama	Standart Sapma	Frekans
Ön test	Deney	39.17	16.019	18
	Kontrol	30.00	8.584	20
	Toplam	34.34	13.314	38
Son test	Deney	61.94	10.729	18
	Kontrol	28.50	9.333	20
	Toplam	44.34	19.596	38
İzleme testi	Deney	61.11	14.303	18
	Kontrol	27.75	12.298	20
	Toplam	43.55	21.369	38

İki faktörlü tekrarlı ölçümlerde Sphericity Varsayımı, sonuçlara güvenilip güvenilmeyeceği konusunda önemlidir. Sphericity Varsayımının sağlanıp sağlanmadığı Mauchly's Test of Sphericity ile araştırılır (Mauchly, 1940). Bu test, n değişkenli bir popülasyondan alınan bir örneğin aslında varyansların hepsinin eşit olduğu ve korelasyonların sıfır olduğu bir popülasyondan olduğu hipotezinin test edilmesiyle ilgilidir. Bu simetriye sahip bir popülasyon "küresel" olarak adlandırılır. Değişkenlerin doğrusal ortogonal dönüşümü altında, küresel bir popülasyon değişmeden kalır. Sonuç olarak bu hipotezle ilgili bilgi veren bir örneğin özellikleri bu dönüşümler altında değişmez olmalıdır. Diğer bir ifadeyle bu testte, bütün tekrarlanan ölçümlerin eşit varyansa sahip olduğunu ve birbirleriyle eşit korelasyon içinde olduğu varsayılmaktadır. Küresellik varsayımını ihlal eden durumlarda, konu içi istatistikler anlamsızdır. Sphericity okurları, göreceli olarak karmaşık bir cebirsel kavramla karşılaşabilirler. Bu yüzden çeşitli istatistik programları ile hesaplanan Mauchly's Test of Sphericity analiz sonuçlarında eğer $p > .05$ ise varsayım sağlanmış demektir (Mauchly, 1940).

Tablo 13. *Mauchly's Test of Sphericity analiz sonuçları*

Grup içi etki	Mauchly's W	Ki kare	df	p	Epsilon		
					Greenhouse-Geisser	Greenhouse-Geisser	Greenhouse-Geisser
Zaman	.930	2.536	2	.281	.935	1.000	.500

Tablo 13.'e göre p değeri $0.281 > .05$ olduğu için Sphericity Varsayımı anlamlıdır. Gruplar arası hesaplanan varyanslar homojendir. Bu da ölçümlerin birbirinden bağımsız yapıldığını göstermektedir.

Analizde varsayım sağlandığı için grup içi etki testi tablosundaki Sphericity Assumed değerleri incelenir.

Tablo 14. *Grup içi etki testi*

Kaynak	Tip III Kareler toplamı	df	Ortalama kare	F	p.	Kısmi eta kare	Yüzde olmayan parametre	Gözlemlenen güç	
Zaman	Sphericity Assumed	2662.490	2	1331.245	11.822	.000	.247	23.644	.993
	Greenhouse-Geisser	2662.490	1.869	1424.289	11.822	.000	.247	22.100	.990
	Huynh-Feldt	2662.490	2.000	1331.245	11.822	.000	.247	23.644	.993
	Lower-bound	2662.490	1.000	2662.490	11.822	.001	.247	11.822	.917
Zaman * Grup	Sphericity Assumed	3709.859	2	1854.929	16.473	.000	.314	32.945	1.000
	Greenhouse-Geisser	3709.859	1.869	1984.575	16.473	.000	.314	30.793	.999
	Huynh-Feldt	3709.859	2.000	1854.929	16.473	.000	.314	32.945	1.000
	Lower-bound	3709.859	1.000	3709.859	16.473	.000	.314	16.473	.977
Hata (Zaman)	Sphericity Assumed	8107.685	72	112.607					
	Greenhouse-Geisser	8107.685	67.296	120.477					
	Huynh-Feldt	8107.685	72.000	112.607					
	Lower-bound	8107.685	36.000	225.213					

Tablo 14'teki Sphericity Assumed satırındaki p değeri $< .05$ olduğu için de yapılan deneysel işlemin gruplar arasında farklı zamanlarda yapılan ölçümlere göre istatistiksel olarak anlamlı bir fark olduğunu göstermektedir. Analizden elde edilen bu anlamsal büyüklüğün incelenmesi de sonuçların genelleştirilmesine katkı sağlayacaktır.

Araştırmacılar etki büyüklüğü istatistiklerini sunmak ve tartışmak için giderek daha fazla bir şekilde talepte bulunmaktadır (American Psychological Association, 2001). *Büyük* veya *küçük*, dereceyi ifade eden bu iki kelime, araştırma yazılarında etki büyüklüğü tahminlerini bildirme şartını savunan birçok istatistikçinin önerilerini yansıtmaktadır (Bezeau & Graves, 2001; Fan, 2001;

Hyde, 2001; Kirk, 1996; Shaver, 1993; Vacha-Haase, 2001). Neredeyse bütün nicelikler gibi, arařtırmacılar (a) belirli bir popülasyondaki etki büyüklüğünün deęerini gerçekten bilmek ve (b) bu popülasyondan seçilen bir örneklem ile de bu deęeri tahmin etmek isterler. Sonuç olarak, popülasyon deęerleri (yani, parametreler) hakkındaki açıklamalar, deney yoluyla elde edilen bu deęerlere (yani istatistiklerin) karşılık gelen tahminlerle ortaya çıkmaktadır (Robey vd., 1999). Arařtırmacılar, tekrarlanan ölçümleri içeren varyans analizlerinden elde edilen etki büyüklüklerini genelleştirilmiş eta karesi istatistięi ile rapor etmektedirler (η_G^2 ; Olejnik & Algina, 2003).

ANOVA alanyazınında, etki büyüklüğü istatistięi genellikle eta karesi (η^2) olarak adlandırılır ve etkinin toplam varyansa oranı ile hesaplanmaktadır (Kerlinger,1964).

$$\eta^2 = \frac{SS_{\text{effect}}}{SS_{\text{total}}},$$

η^2 , birçok hiyerarşik regresyon durumunda veya tek yönlü bir ANOVA'da iyi olabilir, ancak bir ANOVA'da birden fazla varyans kaynağı olduęunda sorun oluşturabilir (Bakeman, 2005).

İki yönlü AxB faktörlü ANOVA hesaplarında, faktör B, SS_{total} 'a ratgele olmayan farklarla katkıda bulunabilir; bu nedenle de, iki η^2 deęerinin karşılaştırılmasında sorun olacaktır. Genellikle η_p^2 (kısmi eta kare), bu karşılaştırılabilirlik problemine bir çözüm olarak sunulmaktadır (Keppel, 1991; Tabachnick & Fidell, 2001). Olejnik ve Algina' a (2003) göre, faktoriyel tasarım durumlarında η_p^2 , gruplar arasında ve gruplar arası tasarımlarla yapılan çalışmalar arasında doğrudan karşılaştırılmaz olan dięer bireysel farklılık kaynaklarını ortadan kaldırmaktadır (Bakeman, 2005).

$$\eta_p^2 = \frac{SS_{\text{effect}}}{SS_{\text{effect}} + SS_{\text{s/cells}}}.$$

AxB faktörlü ANOVA tasarımlarında kareler toplamı dört bileşene ayrılır: SS_A , SS_B , SS_{AB} ve hata terimi veya $SS_{s/AB}$. Böylece, A etkisi, B etkisi ve AB

etkileşimi için η_p^2 paydaları sırasıyla $SS_A + SS_{s/AB}$, $SS_B + SS_{s/AB}$ ve $SS_{AB} + SS_{s/AB}$ olacaktır. Analiz için SPSS'nin güncel sürümleri kullanıldığında ise, etki büyüklüğü istatistikleri η_p^2 sağlamakta ve η^2 sağlamamaktadır (Bakeman, 2005).

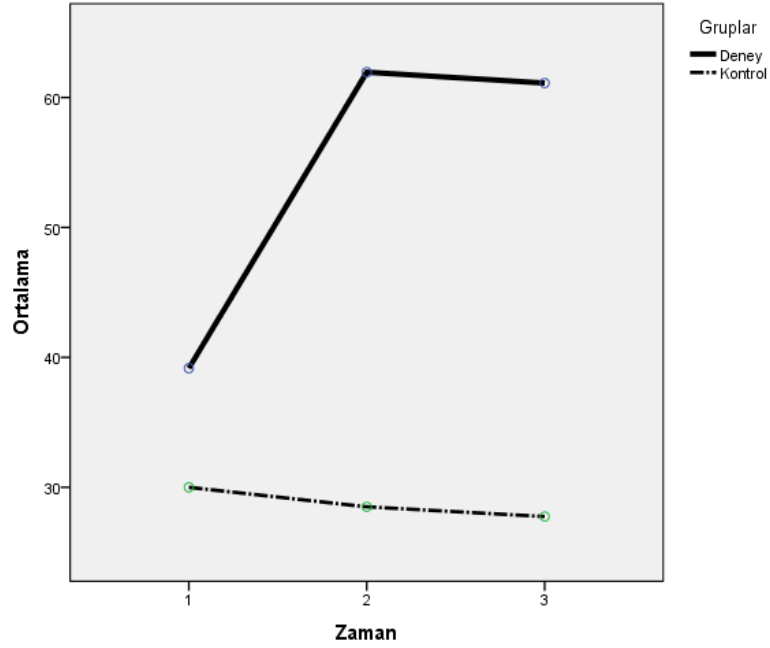
Partial eta-square etki büyüklüğü tahmin değerleri, analiz sonucu oluşan Tests of Within-Subjects Effects tablosunda yer almaktadır ve η_p^2 değerleri .01–.03 arasında ise etki büyüklüğü küçük, .06–.09 arasında ise etki büyüklüğü orta, >.14 ise etki büyüklüğü yüksek olarak yorumlanmaktadır (Cohen 1988). Buna göre tablo 14 incelendiğinde Zaman*Grup etkileşiminin kısmi eta-kare .314 olduğu ve yapılan deneysel işlemin etki büyüklüğünün yüksek olduğu görülmektedir.

Tablo 15'te gruplar arası ön test - son test arasındaki ilişki ile son test - izleme testi arasındaki ilişki verilmiştir. Buna göre gruplar arası ön test ve son test arasında anlamlı bir ilişki bulunmuş, son test ve izleme testi arasında ise anlamlı bir ilişki bulunamamıştır.

Tablo 15. Test sonuçlarının karşılaştırılması

Kaynak	Ölçümler	Tip III kareler toplamı	df	Ortalama kare	F	p
Zaman	Ön test ve Son test	4289,152	1	4289,152	19,383	,000
	Son test ve İzleme testi	23,750	1	23,750	,135	,715
Zaman* Grupo	Ön test ve Son test	5583,889	1	5583,889	25,234	,000
	Son test ve İzleme testi	,066	1	,066	,000	,985
Hata (Zaman)	Ön test ve Son test	7966,111	36	221,281		
	Son test ve İzleme testi	6326,250	36	175,729		

Şekil 11. Test Sonuçlarının Grafikselleştirilmesi



Şekil 11’de verilen grafik ise tablo 12’deki analiz sonuçlarının grafikselleştirilmesidir. Görüldüğü üzere kesik çizgilerle belirtilen kontrol grubunda deneysel işlem uygulanmadan yapılan öğretimde öğrenci test puanlarının ortalamasının üç ölçüm sonuçlarında giderek bir azalma meydana gelmiştir. Birinci ölçüm (ön-test) ortalaması 30 puan olan kontrol grubunda 8 hafta sonra uygulanan ikinci ölçümde (son-test) ortalama kısmen azalmış ve 4 hafta sonra uygulanan üçüncü ölçümde (izleme-testi) ise ortalama puanının biraz daha düşmüş olduğu görülmektedir. Düz çizgi ile ifade edilen deney grubunda ise birinci ölçüm (ön-test) ortalaması yaklaşık 40 puandır. 8 hafta sonra uygulanan ikinci ölçümde (son-test) ortalama 60 puan üzerine çıkmış ve 4 hafta sonra uygulanan üçüncü ölçümde (izleme-testi) ise ortalama puanın yine 60 puan seviyelerinde olduğu görülmektedir. Yapılan analizde, öğrencilerin öntest ve sontest puanları arasında bulunan bu anlamlı farklılığın programlama temelleri dersinde Arduino kullanılmasından kaynaklandığı söylenebilir.

Alanyazında programlama dili, elektronik cihazların yapmasını istediğimiz işlemler için kullanılan, özel sembol ve kelimelerle ifade edilen komutlar bütünü olarak tanımlanır ve programlama dili derslerinde, bu diller öğretilirken genellikle gösterip yaptırma tekniğinin kullanıldığı söylenmektedir (Balcı vd., 2017). Fakat yapıdaki soyut ve karmaşıklık sebebiyle yeni yöntemler geliştirilmesi gerektiği de savunulur (Rubio vd., 2013; Ersoy vd., 2011). Bu doğrultuda öğrencilere üst

düzyer bilişsel beceriler kazandırmak ve programlama öğretimindeki zorlukları giderebilmek için robot programlamanın kullanılması önerilir (Arslan ve Tanel, 2017). Böylelikle programlamanın zorlukları sebebiyle düşük olan akademik başarının da arttırılabileceđi belirtilmiştir. Çünkü robotik programlama ile öğrenciler hem işbirlikli çalışma fırsatı bularak birbirleriyle etkileşim halinde problemlere birden fazla çözümler geliştirebilecekler hem de özgün tasarımlar ile yaratıcılıkları gelişecektir. Dolayısıyla öğrencilerin derslerdeki performansları, ilgi, istek ve başarıları artacaktır (Junior vd., 2013). Gezici ve diğerlerinin (2017) üniversite öğrencileri üzerinde yaptığı bir çalışmada, 14 haftalık Arduino ile verilen güz dönemindeki programlama eğitimi sonucunda, dersi alan 47 öğrencinin % 85'i derste başarılı olmuştur. Aynı dersi bir önceki sene alan 38 öğrencinin derste başarı oranı %26'dır. İlgili dersi bahar döneminde alan 47 öğrencinin başarı oranı ise %72 olmuştur. Bir önceki sene bahar döneminde dersi alan 38 öğrencinin ders başarı oranının % 48 olduğu vurgulanan çalışmada Arduino ile verilen programlama eğitiminin başarıyı arttırdığı belirtilmiştir.

Hertzog ve Swart (2016) tarafından 64 mühendislik öğrencileri ile yaptığı bir çalışmada, öğrencilere bir sömestir boyunca Arduino ile programlama eğitimi verilmiştir. Eğitim sonunda öğrencilerin %91'i Arduino ile etkileşimin dersi daha iyi anlamalarına ve temel tasarım ilkelerini kavramalarına yardımcı olduğunu, Arduino' yu proje tasarımlarında başarıyla kullanabileceklerini belirtmişlerdir. Ayrıca bu çalışma, ilgili mühendislik öğrencileri için akademik başarıya da katkıda bulunmuştur. Öğrencilerin % 90'ından fazlası diğer öğrencileri Arduino' yu almaya teşvik edeceğini, % 93'ü Arduino' nun kendileri için değerli bir öğrenme deneyimi olduğunu belirtmişlerdir. Memnuniyeti ifade eden bu sonuçlar, öğrencilerin derse karşı motive olduklarını, motive olmuş öğrencilerin ise akademik olarak daha iyi performans gösterdiğini ve diğer öğrencilerin de yararlanabilmesi için daha iyi bir öğrenme ortamı oluşturulmasına katkıda bulunduğunu göstermektedir.

Ardışık üç dönem boyunca yapılan bir başka çalışma, öğrencilerin % 94'ünün ders projeleri için Arduino'yu seçtiklerini ortaya koymuş ve böylece bu öğrencilerin ders performanslarının iyileştiđi belirtilmiştir (El-Abd, 2016). Rodriguez-Sanchez ve diğerleri (2016) tarafından yapılan bir başka çalışmada ise, öğrenme ortamında işbirlikli öğrenme yaklaşımının benimsenmesi ve öğretim aracı olarak da Arduino kullanılmasıyla, öğrencilerin akademik başarıları % 93,5

puan ile % 85'in üzerine çıkmıştır (Dasig, 2014). Ayrıca Arduino ile yapılan projelerde kod karmaşıklığının ve proje geliştirme süresinin azaldığı vurgulanmıştır (Dasig, 2014; El-Abd, 2016). Çünkü Arduino kullanımı, öğrencilerin daha kısa sürede uygun dokümantasyon ile fonksiyonel sistemler geliştirmelerine izin verdiği söylenmektedir (El-Abd, 2016).

Omar ve diğerleri (2017) yedi s6mestr boyunca üniversite öğrencilerine Arduino ile verdiği programlama eğitimi sonucunda, öğrencilerin bu yedi yılda aldıkları final notlarını karşılaştırmıştır. Çalışma sonucunda yapılan analizlerde A ve A+ dereceye sahip öğrencilerin oranlarında %6'dan %23'e çıkan bir artış gözlemlenmiştir. Benzer şekilde Haughery (2018) de üniversite düzeyindeki 84 öğrenci ile mekanik ve bilgisayar sistemleri ile ilgili derslerdeki başarı oranlarını inceleyen bir çalışma gerçekleştirmiştir. Çalışmada öğrencilere Arduino kullanımı ile ilgili ödevler verilmiş ve deney-kontrol grupları arasındaki akademik fark gözlemlenmiştir. Analiz sonucunda Arduino ile çalışan deney grubunun akademik başarılarında olumlu gelişme elde edilmiştir.

Pong ve diğerleri (2015) tarafından, Elektronik, bilgisayar ve makine mühendisliği alanlarında öğrencilerin akademik başarı için ihtiyaç duydukları becerileri geliştirmelerine yardım etmek için Arduino kullanılarak çeşitli aktiviteler gerçekleştirilmiştir. İşbirlikli öğrenme yaklaşımı benimsenen çalışmada, 16 öğrenciden elde edilen sonuçlara göre öğrencilerin laboratuvar, ilgili disiplinleri anlama, problem çözme ve eleştirel düşünme becerileriyle akademik performanslarında artış gözlemlenmiştir.

Lise öğrencilerinin Arduino kullanarak programlama derslerinde göstermiş oldukları akademik performansı inceleyen bu çalışmadan elde edilen sonuçların da alanyazında yapılan çalışmaları desteklediği görölmektedir. Ayrıca öğrenme ortamının işbirlikli öğrenme yaklaşımı benimsenerek tasarlanması, bu başarıyı olumlu yönde etkileyen bir unsur olarak da söylenebilir.

4.2. Programlamaya Yönelik Tutum Ölçeği ANOVA Analizleri

Araştırmacı tarafından geliştirilen 3 faktörlü tutum ölçeği deney ve kontrol gruplarına iki farklı zamanda uygulanmıştır. Birincisi deneysel işlem öncesi, ikinci ise 8 haftalık süren deneysel işlem sonrasıdır. Bu iki ölçümden elde edilen analiz sonuçları ise aşağıda yer almaktadır.

Tablo 16. *Bağımlı Değişken İsimleri*

Zaman	Bağımlı Değişkenler
1	Ön test
2	Son test

Tablo 16.'da ölçüm isimleri verilmiştir. Buna göre birinci ölçüm için verilen isim Tutum_ontest, ikinci ölçüm için verilen isim ise Tutum_sontest' tir.

Tablo 17. *Grup İsimleri*

Gruplar	Frekans
Deney	18
Kontrol	20

Tablo 17.'de grup isimleri ve frekanslar verilmiştir. Buna deney grubu frekansı 18, kontrol grubu frekansı 20'dir.

Tablo 17. *Grupların Test Ortalamaları*

	Grup	Ortalama	Standart Sapma	Frekans
Tutum_ontest	Deney	54.67	9.235	18
	Kontrol	41.20	10.196	20
	Toplam	47.58	11.790	38
Tutum_sontest	Deney	52.67	6.126	18
	Kontrol	46.15	11.184	20
	Toplam	49.24	9.610	38

Tablo 18'de grupların ön-test ve son-test ortalamaları yer almaktadır. Buna göre deney grubunun ön-test tutum ortalamaları 54.67, son test tutum ortalamaları ise 52.67 olarak bulunmuştur. Kontrol grubunun ise ön-test tutum ortalamaları 41.20, son-test tutum ortalamaları ise 46.15 olarak hesaplanmıştır.

Tablo 19. *Mauchly's Test Of Sphericity*

Grup içi etki	Mauchly's W	Ki kare	df	p	Epsilon		
					Greenhouse-Geisser	Greenhouse-Geisser	Greenhouse-Geisser
Zaman	1.000	.000	0	.	1.000	1.000	1.000

Sphericity varsayımı için tablo 19'da verilen Mauchly's Test of Sphericity sonuçlarına baktığımızda p değerinde herhangi bir sayısal değer bulunmamaktadır. Çünkü iki faktörlü tekrarlı ölçümlerde ölçüm sayısı iki ise Sphericity varsayımı zaten sağlanmamış demektir (Howell, 2002). Bu yüzden ilgili tabloda Greenhouse-Geisser değerine bakılmalıdır. Greenhouse-Geisser değeri bu tabloda $>.75$ olduğu için de tablo 20'de verilen grup içi etki testi tablosundaki Huynh-Feldt istatistikleri incelenmelidir (Field, 2013; Howell, 2002).

Tablo 20. *Grup İçi Etki Testi*

	Kaynak	Tip III kareler toplamı	df	Ortalama kare	F	p	Kısmi eta kare
Zaman	Sphericity Assumed	41.222	1	41.222	.636	.430	.017
	Greenhouse-Geisser	41.222	1.000	41.222	.636	.430	.017
	Huynh-Feldt	41.222	1.000	41.222	.636	.430	.017
	Lower-bound	41.222	1.000	41.222	.636	.430	.017
Zaman * Grup	Sphericity Assumed	228.801	1	228.801	3.531	.068	.089
	Greenhouse-Geisser	228.801	1.000	228.801	3.531	.068	.089
	Huynh-Feldt	228.801	1.000	228.801	3.531	.068	.089
	Lower-bound	228.801	1.000	228.801	3.531	.068	.089
Hata (Zaman)	Sphericity Assumed	2332.475	36	64.791			
	Greenhouse-Geisser	2332.475	36.000	64.791			
	Huynh-Feldt	2332.475	36.000	64.791			
	Lower-bound	2332.475	36.000	64.791			

Tablo 20.'de verilen sonuçlardaki Huynh-Feldt satırındaki p değeri ise .068 bulunmuş ve bu sonuç $>.05$ olduğu için iki grup arasında tutum puanları açısından anlamlı bir fark olmadığını göstermektedir.

Alanyazında yapılan çalışmalar incelendiğinde deney grubundaki öğrencilerin tutumlarındaki azalmaya sebep olarak, programlama yaparken karşılaşılan zorluklar ve programlama amacıyla bilgisayarda harcanan günlük süre olarak gösterilmiştir (Karacı, 2016;Woodrow, 1992). Karacı'ya (2016) göre, bilgisayarda programlama amacıyla günde 4 saatten fazla zaman harcayan öğrencilerin, bilgisayarda programlama amacıyla günde 2 ila 4 saat harcayanlara kıyasla daha olumlu tutumlara sahip olduğunu göstermektedir. Ayrıca öğrencilerin program yazarken kendilerini daha kaygılı ve stresli hissetmeleri ve programlamadaki karmaşıklık, öğrencilerde tutuma yönelik olumlu bir artışın olmasına engel olmaktadır (Balcı vd., 2017).

Korkmaz ve Altun'un (2013) tarafından üniversite düzeyindeki öğrencilerle yaptıkları bir çalışmada, öğrencilerin programlama sürecinde kod editörlerinden aldıkları hatalara ilişkin problemler ve dikkatsizlik gibi durumların, öğrencilerde programlamaya yönelik olumsuz bir tutum oluşmasına sebep olduğu belirtilmiştir. Ayrıca öğrenciler programlama bilgisine sahip olsalar dahi verilen bir problemi nasıl tasarlayacakları konusunda da zorluk yaşamışlardır.

Tan, Ting ve Yang (2009) yaptığı bir başka çalışmada öğrencilerin bir problemi işlem basamaklarına ayırarak çözme işlemlerinde sorunlar yaşadıkları bilgisi desteklenmektedir. Ayrıca çalışmada öğrencilerin programlardaki hataları bulma konusunda da zorluk çektikleri belirtilmiştir. Buna göre programlarda oluşan hataları bulma işi programlama deneyimi ile ilişkilendirilmiştir ve programlama deneyimi yüksek olan kişilerin acemi programcılara göre hata bulma işinde daha başarılı oldukları belirtilmiştir (Bednarik ve Tukiainen, 2004).

Futschek ve Moschitz (2010), tarafından yapılan bir çalışmada da öğrencilerin problem çözümüne yönelik algoritma geliştirme sürecinin zor ve vakit alıcı olarak algıladıkları, bu yüzden de programlama derslerinde karşılaşılan en büyük problemlerden biri (Ziatdinov ve Musa, 2012) olduğu vurgulanmıştır.

Erol ve Kurt (2016) bilgisayar ve öğretim teknolojileri alanında öğrenim gören 128 öğrenci ile programlamaya yönelik tutumun belirlenmesi amacıyla öğrencilerle odak görüşmesi yaptığı çalışmada, öğrencilerin olumsuz tutum sergilemelerinin birçok sebebe dayandığını göstermişlerdir. Çalışmaya göre

öğrencilerin yaşadığı sınav kaygısı, programlamadaki zor yapı, problemin çözümünde yaşadıkları kararsızlık gibi durumların tutumda olumlu yönde bir gelişme gösterilmemesine sebep olmuştur.

Bu çalışmadan elde edilen sonuçlar da alanyazındaki çalışmaları destekler niteliktedir. Elektronik setler kullanarak verilen eğitim öncesi yapılan ön test puanlarına göre öğrencilerin, programlamaya yönelik 54,67 ortalama puanında olumlu bir tutum içinde oldukları görülmektedir. Fakat süreç içinde yaşadıkları başarısız denemeler, problem çözümüne yönelik algoritma geliştirme konusunda yaşadıkları zorluklar, programlarda oluşan hataları ayıklamada yetersiz olmaları gibi durumların eğitim sonunda yapılan son test puanlarında programlama dersine yönelik tutumda olumlu bir artış olmamasına sebep olduğu düşünülmektedir. Bunun yanında programlama yapısındaki soyut kavramlar öğrencilerin zihinsel yükünü arttırmaktadır (Arabacıoğlu, Bülbül ve Filiz, 2007). Programlama üst düzey bilişsel beceriler gerektiren bir alan olduğu için de program yazma sürecinde öğrencilerde kaygı ve stres oranları artmaktadır. Bu da derse karşı olan tutumu olumsuz yönde etkilemektedir.

BÖLÜM V

SONUÇ, TARTIŞMA VE ÖNERİLER

Teknolojideki gelişmeler dünyanın geleceği için özellikle yazılım alanında olmak üzere yaratıcı ve üretkenlik becerilerine sahip bireyler yetiştirilmesini gerektirmektedir. Bu nedenle de tüm dünyada küçük yaşlardan itibaren verilmesi önerilen programlama eğitiminin önemi üzerinde durulmaktadır.

Birçok araştırmacı programlamayı her biri farklı bir bilişsel etkinlik gerektiren bir dizi aşamadan oluştuğunu söylemektedir.(Aron, 1974; Weinberg, 1971) Yapılan analizler sonrasında programlamanın problem belirleme, algoritma tasarımı, kodlama ve hata ayıklama kısımlarından oluştuğu konusunda hem fikir olmuşlardır (Dalbey & Linn, 1985). Bir problemin tüm gerekliliklerini tespit etmek ve programın hangi işlevleri yerine getirmesi gerektiğini netleştirmek için problemin analiz edildiği, problem belirleme aşaması, öğrenciler için genellikle çok karmaşık, fazlaca bilgi ve deneyim gerektiren bir işittir. Aron'a (1974) göre problem analizi, ilgili bilgiyi tanımlamayı, ihtiyaçları değerlendirmeyi, mevcut girdiyi belirlemeyi, istenen hedefleri, sonuçları belirlemeyi ve hesaplanabilir fonksiyonları hesaplanamayanlardan ayırmayı içermektedir. Algoritma tasarımı aşamasında Jones'a (1970) göre öğrencilerin sahip olması gereken iki önemli beceri vardır. Birincisi, yaratıcılık kullanılarak problem için potansiyel çözüm yöntemini aramak, ikincisi ise belli ölçütler dâhilinde uygulama için uygun bir yaklaşım seçmektir. Böylelikle kişiler çözüme yönelik bilinçli bir karar verirler. Kodlama ise, bir tasarımın belirli bir programlama dilinde uygulanmasıdır. Programlama dilleri biçimsel dillerdir ve doğru kullanım için birçok kurala sahiptir. Bu da programlama dillerinin öğrenilmesinin zor olmasının önemli bir nedenidir.

Bir bilgisayarı programlamak için, öğrencilerin problem çözme becerileri hakkında iyi bir bilgiye sahip olmaları, programlama dili sözdizimi ve anlam bilgilerinin yanı sıra mevcut kodu anlayabilmeleri, hataları test edip düzeltmeleri gerekmektedir (Linn ve Clancy, 1992). Bu sebeple programlama öğrenmek zor bir süreçtir. Sloane ve Linn (1988), programlamanın sadece bir beceri değil, aynı zamanda bir beceri hiyerarşisi olduğunu ve programcının kariyeri boyunca çoğuna ihtiyaç duyacağını, Dijkstra (1989), programlamayı öğrenmenin, uygulamaya dayalı bir çalışma gerektiren, yavaş ve kademeli bir süreçtir olduğunu ve çok daha fazla teoriye dayanan diğer derslerden oldukça farklı olduğunu söylemişlerdir. Bu

yüzden programlamayı öğrenmeye başlayan pek çok öğrenci zorluklarla karşılaşmaktadır. Alanyazına göre bu zorluklar problemi nasıl çözeceklerini tasarlayamama, problemi alt bileşenlerine ayıramama, hataları kavrayamama ve motivasyon eksikliğidir (Dalbey & Linn,1985; Erol ve Kurt, 2017; Miliszewska ve Tan, 2007). Bu noktada programlama derslerinde kullanılan öğretim metotları önem kazanmaktadır. İlgili programlama dilinin sözdizimi üzerine yoğunlaşarak çok az pratik yapmaya ve diğer derslerdeki gibi sadece teorik konuları okumaya dayalı bir öğretim yönetiminde öğrenciler, öğrenmeye katılmamakta ve kavramları anlayamamaktadırlar (Almeida vd., 2002). Bu yüzden programlama eğitiminde yeni yöntemlere yer vermek ve bir problemi çözmek için kullanılacak kavram ve stratejiler hakkındaki bilgileri öğrenme sürecine dâhil etmek önemlidir. Ayrıca Lahtinen ve diğerleri (2005) de, öğrencilerin programlamadaki değişkenler, veri türleri, hafıza adresleri gibi soyut ve temel kavramları öğrenirken yaşadıkları zorluklar sebebiyle programlamaya karşı olumsuz bir tutum içine girdiklerini belirtmişlerdir. Tutum, birey davranışlarını saptayan psikolojik bir parametredir. Davranışlar bilgiyi ne şekilde elde edeceğimizi ve hareketlerimize etki etmektedir. Bu nedenle bir dersin öğreniminde kullanılacak yeni yöntemlerin öğrenci tutumlarına nasıl bir etki yapacağına da incelenmesi önerilir.

Bu çalışmada alanyazından elde edilen bilgilere göre lise öğrencilerinin programlama öğrenimine yönelik, arduino kartları kullanılarak sekiz haftalık etkinlikler düzenlenmiş ve bu etkinliklerin öğrenci tutumlarına ve akademik başarılarına nasıl etki edeceği incelenmiştir.

Derste uygulanan etkinlikler robotik programlama temeline dayandırılarak hazırlanmıştır. Çünkü robotik programlama öğrencilerin yaratıcılıklarını, eleştirel düşünme becerilerini işbirlikli çalışma ve problem çözme becerilerini geliştirmektedir (Alimisis & Kynigos, 2009). Soyut olan programlama kavramlarının somutlaştırılmasına da katkı sağlamaktadır (Ersoy vd., 2011; Witherspoon, 2018).

Robotik programlamada kullanılacak malzemelerin kolay kullanım, kolay programlanabilme, kolay temin edilebilirliği, ucuz olması ve her platformda çalışabilmesi gibi özellikleri, etkinliklerin tasarlanmasında dikkate alınmıştır. Bu sebeple sekiz hafta süren robotik etkinliklerde ana malzeme olarak Arduino devre kartı seçilmiştir. Bunun yanında birçok temel elektronik malzemeler kullanılmıştır. Bu çalışma Ege Üniversitesi tarafından desteklenen 17-EĞF-002

nolu Bilimsel Araştırma Projesidir ve çalışmanın tüm materyalleri proje desteğiyle alınmıştır.

Araştırmada deneysel desenlerden kontrol grublu ön-test son-test modeli kullanılmıştır. Bu modelde tek bir grup üzerinde uygulanan çalışma ile deneysel işlemin etkisi test edilmiştir. Çalışmaya katılan öğrencilerin bağımlı değişkene bağlı ölçümleri ön test yardımı ile uygulama öncesinde, son-test ile de uygulama sonrasında aynı ölçme araçları ile elde edilmiştir. Ön test ve son test uygulaması yapılan katılımcılar değişmemiştir. Çalışmada deney grubu ve kontrol grubu, farklı illerde öğrenim göre meslek lisesi bilişim teknolojileri alanı 10. sınıf öğrencilerinden oluşmaktadır.

Çalışmaya hazırlık aşamasında öncelikle öğrenen analizi yapılmış, öğretim amacı ve çalışma yapılacak programlama konuları belirlenmiştir. Belirlenen konuların hedeflerinin öğrencilere kazandırılması amacıyla da uygulama ağırlıklı bir öğretim programı tasarlanmıştır. Bu öğretim programında ise öğrencilerin eleştirel, yaratıcı düşünme ve problem çözme becerilerini, iletişim, araştırma ve sorgulama yeteneklerini geliştirip sergileyebilecekleri etkinlikler düzenlenmiştir. Daha sonra programlama konularının belirlenmesinde MEB müfredatı dâhilinde C# programlama dilinin kullanılmasına karar verilmiş ve Bloom (Bloom, 1956) taksonomisinin bilişsel düzey tasarım basamakları dikkate alınarak, öğrencilerin başarı durumlarını belirlemek için akademik başarı testi hazırlanmıştır (Ek 1). Yapılan etkinliklerin öğrenci tutumlarına etkisinin incelenmesi amacıyla da bilgisayar programlamaya karşı tutum ölçeği geliştirilmiştir (Altay ve Kışla, 2018). Her iki ölçme aracı ön test olarak öğrencilere uygulanmıştır. Deney grubu öğrencilerine Arduino elektronik devre kartının tanıtımı ve kullanım şekli hakkında bilgi verilmiş ve sekiz hafta boyunca tasarlanan etkinlikler uygulanmıştır. Bu süreç sonunda her iki gruba son test olarak iki ölçme aracı tekrar uygulanmıştır. Dört hafta boyunca elektronik malzemeler kullanılmadan verilen eğitimden sonra öğrencilerin öğrenilen bilgileri hatırlama düzeylerini incelemek amacıyla her iki gruba izleme testi uygulanmıştır.

Toplanan veriler iki yönlü tekrarlı ölçümler ANOVA yöntemiyle analiz edilmiş ve akademik başarı açısından yüksek etki düzeyinde anlamlı bir fark elde edilmiştir. Deney grubu öğrencilerinin ön test ve son test puan ortalamaları arasında 20 puanlık, son test ve izleme testi puan ortalamaları arasında ise 1 puanlık bir fark gözlemlenmiştir. Kontrol grubundaki ölçüm sonuçlarında ise

birkaç puanlık bir azalma gözlemlenmiştir. Bu durumda, Arduino ile verilen programlama eğitiminin öğrencilerin akademik başarılarında olumlu düzeyde yüksek bir etki yaptığı ve öğrenilen konuların kalıcılığına katkı sağladığı söylenebilir.

Tutum ölçeği analizlerinde ise iki grup arasında anlamlı bir fark bulunamamış ve robotik etkinlikler sonucunda programlamaya yönelik olumsuz bir tutum oluşmamıştır.

Sonuç olarak bu çalışmada, giderek artan şekilde bir öneme sahip olan programlama eğitiminde öğretilen programlama dilleri için arduino destekli öğretim tasarımı ile öğrenci başarıları ve tutumları incelenmiştir. Çalışma sırasında yapılan gözlemlerde, düzenlenen ders etkinliklerinin öğrencilerin ilgisini çektiği fark edilmiştir. Öğrenme sürecinde öğrencilerin ilgisini çekebilecek etkinliklerin düzenlenmesi hem öğrencilerin sürece daha aktif olarak katılmasını sağlamakta hem de öğrenilen bilginin hatırlanma düzeyini arttırmaktadır.

Öğrencilerin süreçte aktif olabilmesini sağlamak için öğrenme ortamında kullanılan öğretim yaklaşımının önemi de çalışma sonuçlarına yansımaktadır. Çalışmada öğrenme ortamının işbirlikli çalışmaya elverişli olarak tasarlanması ile öğrenciler arası etkileşim gerçekleştirilmiştir. Alanyazında öğrenme etkinliklerinde öğrenciler arası gerçekleşen etkileşimin akran öğrenmesini desteklediği, öğrencilerin problem çözme yeteneklerini geliştirdiği ve iletişim becerilerini artırdığı belirtilmiştir (Açıkgöz, 1992). Buna ek olarak günümüzde programlama eğitiminde sıkça tercih edilen robotik programlamanın sağladığı somutlaştırma ile de öğrencilerin soyut olarak algıladıkları kavramların öğrenilmesi kolaylaşmıştır.

Araştırmada öğrenciler ürün geliştirmeye odaklı bir eğitim almışlardır. Böylelikle öğrencilerin yaparak ve yaşayarak öğrenmelerini destekleyecek bir ortam sağlanarak onların robotik programlama ile küçük projeler üretmeleri ve daha büyük projelerdeki karmaşık problemleri çözmeleri (Akpınar ve Altun, 2014; Çakıroğlu vd., 2011; Fesakis & Serafeim, 2009) için destek sağlanmıştır. Robotik programlamada kullanılacak materyal olarak tüm dünyada sıklıkla tercih edilen Arduino platformunun kullanılması da çok disiplinli öğrenmeyi sağlamıştır. “Fizik”, “Matematik”, “Temel Elektronik” gibi derslerdeki bilgilerin de kullanılmasını sağlayan Arduino devre kartları ile geliştirilen elektronik devreler, bu derslerde öğrenilen bilgilerin de pekiştirilmesini sağlamaktadır. Bu derslerde

ve özellikle programlama derslerinde yeterli başarıyı elde etmek için üst düzey bilişsel becerilere sahip olunması gerektiği için öğrencilerde bu derslere yönelik olumsuz tutum sergilenme olasılığı meydana gelebilmektedir. Fakat araştırma sonucunda görülmüştür ki öğrencilerin programlama dersine karşı tutumlarında olumsuz bir durum oluşmamaktadır. Yapılan çalışma hem programlamaya yönelik tutumda bir olumsuzluk oluşturmamış hem de akademik başarı yönünde olumlu anlamda yüksek bir etki elde etmiştir.

- Yapılan bu çalışma çeşitli şekilde düzenlenebilir, farklı yönlerden geliştirilebilir. Bu kapsamda araştırmacılara önerilerim şu şekildedir:
- Bu araştırma programlama konusunda bilgi sahibi olmayan kişilerle yapılmıştır. Bunun yerine programlama konusunda ön bilgisi olan öğrenciler üzerindeki etkiler incelenebilir.
- Nitel araştırma yöntemleri kullanılarak öğrencilerin görüşleri incelenebilir.
- Öğrencilerin programlamaya yönelik tutumlarının yanısıra kaygı, motivasyon vb. farklı özellikler de incelenebilir.
- Programlama öğreniminde daha çok uygulama yapılması öğrenmeyi pekiştireceği için deneysel işlem süresi daha uzun tutulabilir.

KAYNAKÇA

- Açıkgöz, K. Ü. (1992). İşbirlikli öğrenme, kuram, araştırma, uygulama, Malatya: Uğurel Matbaası.
- Adıgüzel, O. C., ve Berk, Ş. (2009). Mesleki ve teknik ortaöğretimde yeni arayışlar: yeterliğe dayalı modüler sistemin değerlendirilmesi, *Yüzüncü Yıl Üniversitesi Eğitim Fakültesi Dergisi*, Haziran. Cilt:V1, Sayı: I, 220-236.
- Akçay, A., ve Çoklar, A. N. (2018). Bilişim Teknolojileri ve Yazılım Dersi Öğretmen Adaylarının Programlamaya İlişkin Algılanan Öz Yeterliklerinin Farklı Değişkenler Açısından İncelenmesi, *Kastamonu Eğitim Dergisi*, 26(6), 2163-2176.
- Akçay, A., ve Çoklar, A. N. (2016). Bilişsel becerilerin gelişimine yönelik bir öneri: Programlama eğitimi, *Eğitim Teknolojileri Okumaları*, 121-139.
- Akpınar, Y., ve Altun, A. (2014). Bilgi toplumu okullarında programlama eğitimi gereksinimi, *İlköğretim Online*, 13(1), 1-4.
- Akpınar, E., ve Ergin, Ö. (2005). Yapılandırmacı kuramda fen öğretmenin rolü, *İlköğretim Online*, 4(2), 55-64.
- Aldridge, R., Brandt, T., & Parikh, C. (2016). Autonomous robot design and Build: Novel hands-on experience for undergraduate students. In *Proceedings of the 2016 ASEE North central section conference*, 18-19.
- Alimisis, D., & Kynigos, C. (2009). Constructionism and robotics in education. *Teacher Education on Robotic-Enhanced Constructivist Pedagogical Methods*, 11-26.
- Almeida, E. S., Costa, E. B., Braga, J. D. H., Silva, K. S., Paes, R. B., & Almeida, A. M. (2002). Um ambiente de apoio ao aprendizado de programação, In X Workshop sobre Educação em Computação, Florianópolis, Anais do WEI 2002/ SBC2002, 79-88.
- Altay, G., ve Kışla, T. (2018). Computer programming with arduino: example course desing, Sustainable Development and Actual Problems of Humanitarian Sciences International Conference, Azerbaijan University, 623-626.

- Altay, G., ve Kışla, T. (2018). Programlamaya yönelik tutum ölçeği ve psikometrik özellikleri, *Ege Eğitim Dergisi*, 19(2), 559-574.
- Altın, Dr. R., Yalçın, O., Çetin, Ş., Aksoy, M., Kaya, Ö., Alam, G., Yıldırım, O. (2007). Gelişmiş ülkelerin mesleki eğitim sistemleri ve türkiye, Millî Eğitim Bakanlığı Projeler Koordinasyon Merkezi Başkanlığı, Ankara.
- Altun, A., ve Mazman, S. G. (2012). Programlamaya ilişkin öz yeterlilik algısı ölçeğinin türkçe formunun güvenilirlik ve geçerlik çalışması, *Eğitimde ve Psikolojide Ölçme ve Değerlendirme Dergisi*, 3(2), 297-308.
- Arabacıoğlu, T., Bülbül, H. İ., ve Filiz, A. (2007). Bilgisayar programlama öğretiminde yeni bir yaklaşım, *Akademik bilişim*, 193-197.
- Arduino. (2019). <https://www.arduino.cc/en/Main/AboutUs>. (Erişim Tarihi: 22 Mayıs 2019)
- Aron, J. D. (1974). The program development process, Addison-Wesley, Reading, Massachusetts.
- Arslan, K., ve Tanel, Z. (2017). Arduino ile dinleyen öğrencilerden yapan öğrencilere geçiş, *Uluslararası Bilgisayar ve Öğretim Teknolojileri Sempozyumu*, Malatya, Türkiye, 664-674.
- Assaf, D. (2014). Enabling rapid prototyping in K-12 engineering education with BotSpeak, a universal robotics programming language, *Proceedings of 4th international workshop teaching robotics, teaching with robotics & 5th international conference robotics in education*, 19-26.
- Bakeman, R. (2005). Recommended effect size statistics for repeated measures designs, *Behavior research methods*, 37(3), 379-384.
- Balcı, H., Korkmaz, Ö., Çakır, R., ve Erdoğan, F. U. (2017) Görsel Programlama Ortamlarında Yapılan Oyun Geliştirme Etkinliklerinin Öğrencilerin Programlamaya Dönük Tutumları ve Öğrencilerin Kodlamaya Yönelik Öz-Yeterlilik Algılarına Etkisi, *1. Uluslararası Çağdaş Eğitim Ve Sosyal Bilimler Sempozyumu*, 62.
- Balogh, R. (2010). Educational robotic platform based on arduino, *Proceedings of the 1st international conference on Robotics in Education, RiE2010. FEI STU, Slovakia*, pp. 119-122.

- Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12: what is Involved and what is the role of the computer science education community?, *Inroads*, 2(1), 48-54.
- Baş, H. (2016). “Kizlar-kodluyor”, <http://www.milliyet.com.tr/kizlar-kodluyor-ekonomi-2352237/> (Erişim tarihi: 14 Ekim 2018)
- Battal, A. (2017). “Teknoloji devi Güney Kore’de kodlama eğitimi nasıl”, <https://www.teknolugat.com/teknoloji-devi-guney-korede-kodlama-egitimi-nasil> (Erişim tarihi: 14 Ekim 2018)
- Beck, A. T., Emery, G., & Greenberg, R. L. (2005). Anxiety disorders and phobias: A cognitive perspective, Basic Books.
- Bednarik, R., & Tukiainen, M. (2004). Visual attention and representation switching in java program debugging: a study using eye movement tracking, In Proceedings of 16th Annual Psychology of Programming Interest Group Workshop (PIIG'04), Institute of Technology Carlow, Ireland, April 5-7, 2004, 159-169.
- Bennedsen, J. (2003). Teaching java programming to media students with a liberal arts background. *Proceedings for the 7th Java & the Internet in the Computing Curriculum Conference (JICC 7)*, London, United Kingdom.
- Bennedsen, J., & Caspersen, M. E. (2007). Failure rates in introductory programming, *AcM SIGcSE Bulletin*, 39(2), 32-36.
- Benzer, A. İ., ve Erümit, A. K. (2017). Programlama öğretimine yönelik lisansüstü tezlerin incelenmesi, *Journal of Instructional Technologies & Teacher Education Vol*, 6(3), 99-110.
- Bergin, S., & Reilly, R. (2005). Programming: Factors that influence success, *Proceedings of the 36th SIGCSE Technical Symposium on Computer Science Education*, February 23 – 27, St. Louis, Missouri, USA.
- Bezeau, S., & Graves, R. (2001). Statistical power and effect sizes of clinical neuropsychology research, *Journal of clinical and experimental neuropsychology*, 23(3), 399-406.

- Bilgin, İ., ve Karaduman, A. (2005). İşbirlikli öğrenmenin 8. sınıf öğrencilerinin fen dersine karşı tutumlarına etkisinin incelenmesi, *İlköğretim Online*, 4,32-46.
- Bishop- Clark, C., Courte, J., & Howard, E. V. (2007). A Quantitative and Qualitative Investigation of Using Alice Programming to Improve Confidence, Enjoyment and Achievement among Non-Majors. *Journal of Educational Computing Research*, 37(2) 193-207.
- Bloom, B. S. (1956). Taxonomy of educational objectives - Handbook 1 cognitive domain, New York: David McKay Co Inc.
- Buechley, L., Eisenberg, M., Catchen, J., & Crockett, A. (2008). The LilyPad Arduino: using computational textiles to investigate engagement, aesthetics, and diversity in computer science education, In *Proceedings of the SIGCHI conference on Human factors in computing systems*, 423-432.
- Burke, Q. (2012). The markings of a new pencil: Introducing programming-as-writing in the middle school classroom, *Journal of Media Literacy Education*, 4(2), 121-135.
- Büyüköztürk, Ş., Çakmak, E. K., Akgün, Ö. E., Karadeniz, Ş. ve Demirel, F. (2008). *Bilimsel Araştırma Yöntemleri*, Ankara: Pegem A Yayıncılık.
- Byrne, P., & Lyons, G. (2001). The effect of student attributes on success in programming, In *ACM SIGCSE Bulletin*,33(3), 49-52.
- Campbell, D. T., & Stanley, J. C. (2015). *Experimental and quasi-experimental designs for research*, Ravenio Books.
- CEDEFOP. (2013). Labour market outcomes of vocational education in Europe, evidence from the European Union labour force survey (Erişim tarihi: 28 Ekim 2018).
- Cevahir, H., ve Özdemir, M. (2017). Programlama öğretiminde karşılaşılan zorluklara yönelik öğretmen görüşleri ve çözüm önerileri, Uluslararası Bilgisayar ve Öğretim Teknolojileri Sempozyumu, İnönü Üniversitesi, Malatya, 24-26.

- Chan, J., Pondicherry, T., & Blikstein, P. (2013). LightUp: an augmented, learning platform for electronics, In *Proceedings of the 12th International Conference on Interaction Design and Children*, pp. 491-494.
- Cohen, J. (1988). *Statistical power analysis for the behavioral sciences*, 2nd ed., Hillsdale, NJ: Erlbaum.
- Çakır, Ö., Calp, M. H., ve Doğan, A. (2015). Uzaktan eğitimde içerik geliştirme süreci: Gazi Üniversitesi Bilişim Enstitüsü örneği, *Cumhuriyet International Journal of Education*, 4(1),1-20.
- Çakıroğlu, Ü., Sarı, E., ve Akkan, Y. (2011). The view of the teachers about the contribution of teaching programming to the gifted students in the problem solving, In *5th International Computer & Instructional Technologies Symposium (22-24 September)*, 1096-1102.
- Çatlak, Ş., Tekdal, M., ve Fatih, B. A. Z. (2015). Scratch yazılımı ile programlama öğretiminin durumu: Bir doküman inceleme çalışması, *Öğretim Teknolojileri & Öğretmen Eğitimi Dergisi*,4(3), 13-25.
- Çavas, B., Kesercioglu, T., Holbrook, J., Rannikmae, M., Ozdogru, E., ve Gokler, F. (2012). The effects of robotics club on the students' performance on science process & scientific creativity skills and perceptions on robots, human and society, *Proceedings of 3rd International Workshop Teaching Robotics, Teaching with Robotics Integrating Robotics in School Curriculum*, 40-50.
- Çetin, E. (2012). Bilgisayar programlama eğitiminin çocukların problem çözme becerileri üzerine etkisi, Yayınlanmamış yüksek lisans tezi, Gazi Üniversitesi, Ankara.
- Çetin, İ., ve Özden, M. Y. (2015). Development of computer programming attitude scale for university students, *Computer Applications in Engineering Education*, 23(5), 667-672.
- Çölkesen, R. (2002). Veri yapıları ve algoritmalar, Papatya Yayıncılık, Mayıs.
- Dalbey, J., & Linn, M. C. (1985). The demands and requirements of computer programming, A literature review, *Journal of Educational Computing Research*, 1(3), 253-274.

- Dasig Jr. D. (2014). User experience of embedded system students on arduino and field programmable gate array (FPGA), In the Second International Conference in Applied science and Environmental Engineering, 124-128.
- Demir, F. (2015). Programlama öğretiminde eğitsel programlama dilinin farklı kullanımlarının programlama başarısı ve kaygısına etkisi, Doktora Tezi, Atatürk Üniversitesi.
- Demirer, V. ve Sak, N. (2016). Dünyada ve Türkiye'de programlama eğitimi ve yeni yaklaşımlar, *Eğitimde Kuram ve Uygulama*, 12(3), 521-546.
- Demir, Ö., ve Yurdugül, H. (2014). Ortaokul ve lise öğrencileri için bilgisayara yönelik tutum ölçeğinin Türkçe'ye uyarlanması, *Eğitim ve Bilim*, 39(176), 247-256.
- Denner, J., & Werner, L. (2007). Computer programming in middle school: How pairs respond to challenges, *Journal of Educational Computing Research*, 37(2), 131-150.
- Dijital Eğitim. (2018). "Kod-haftasi-2018", <http://dijitalbenlik.com/kod-haftasi-2018/> (Erişim tarihi: 14 Ekim 2018)
- Dijkstra, E. W. (1989). On the cruelty of really teaching computing science. *Communications of the ACM* 32(12), 1398–1404.
- Dooley, K. E. (2005). Advanced methods in distance education: Applications and practices for educators, administrators and learners, Hersley, PA: Information Science Publishing, IGI Global.
- Doymuş, K., Şimşek, Ü., ve Bayrakçeken, S. (2004). İşbirlikçi öğrenme yönteminin fen bilgisi dersinde akademik başarı ve tutuma etkisi, *Türk Fen Eğitimi Dergisi*, 1(2), 103-115.
- Dönmez, F., ve Azizoğlu, N. (2010). Investigation of the students science process skill levels in vocational schools: a case of Balıkesir, *Necatibey Faculty of Education Electronic Journal of Science and Mathematics Education*, 4(2), 79-109.
- Druin, A., ve Hendler, J. (2000). Robots for kids: exploring new technologies for learning, Morgan Kaufmann/Academic Press, San Francisco.

- Du Boulay, B. (1986). Some difficulties of learning to program, *Journal of Educational Computing Research*, 2(1), 57-73.
- Eguchi, A. (2014). Robotics as a learning tool for educational transformation, *Proceeding of 4th international workshop teaching robotics, teaching with robotics & 5th international conference robotics in education Padova, Italy*, 27-34.
- El-Abd, M. (2016). How course projects can successfully prepare engineering students for capstone design projects, *2016 IEEE Global Engineering Education Conference (EDUCON)*, 746-750.
- Elkner, J., Downey, A. B., & Meyers, C. (2010). Bilgisayar bilimcisi gibi düşünmek, Python ile Öğrenme (2. baskı, Çeviren: TE Kalaycı).
- Erdem, E., ve Demirel, Ö. (2002). Program geliştirmede yapılandırmacılık yaklaşımı, *Hacettepe Üniversitesi Eğitim Fakültesi Dergisi*, 23(23), 81-87.
- Erdoğan, B. (2005). Programlama başarısı ile akademik başarı, genel yetenek, bilgisayara karşı tutum, cinsiyet ve lise türü arasındaki ilişkilerin incelenmesi, Marmara Üniversitesi, İstanbul.
- Erdoğan, F. U., ve Çağıltay, K. (2009). Türkiye’de eğitim teknolojileri alanında yapılan master ve doktora tezlerinde genel eğilimler, Akademik Bilişim. Akademik Bilişim Konferansı Bildirileri, Harran Üniversitesi, Şanlıurfa, 389-393.
- Ersoy, H., Madran, R. O., ve Gülbahar, Y. (2011). A model proposed for teaching programming languages: Robotic programming, In *XIII. Academic Information Conference, Malatya, Türkiye*, pp. 731-736.
- Erol, O., ve Kurt, A. A. (2017). BÖTE bölümü öğrencilerinin programlamaya karşı tutumlarının incelenmesi. *Mehmet Akif Ersoy Üniversitesi Eğitim Fakültesi Dergisi*, 1(41), 314-325.
- Esteves, M., & Mendes, A. J. (2004). A simulation tool to help learning of object oriented programming basics, In *Frontiers in Education, 2004, FIE 2004, 34th Annual*, F4C-7.
- Euronews. (2015). ”Avrupa’da bilgisayar programlama dersleri ilköğretim mufredatına giriyor”, <https://tr.euronews.com/2015/09/03/avrupa-da->

[bilgisayar-programlama-dersleri-ilkogretim-mufredatina-giriyor](#) (Erişim tarihi: 14 Ekim 2018)

- Fan, X. (2001). Statistical significance and effect size in education research: Two sides of a coin. *The Journal of Educational Research*, 94(5), 275-282.
- Fessakis, G., Gouli, E., & Mavroudi, E. (2013). Problem solving by 5–6 years old kindergarten children in a computer programming environment: A case study, *Computers & Education*, 63, 87-97.
- Field, A. (2013). *Discovering statistics with IBM SPSS*, Newbury Park, CA: Sage.
- Futschek, G., & Moschitz, J. (2010). Developing algorithmic thinking by inventing and playing algorithms, *Proceedings of the 2010 Constructionist Approaches to Creative Learning, Thinking and Education: Lessons for the 21st Century (Constructionism 2010)*, 1-10.
- Garner, S. (2009). Learning to program from Scratch. 9th IEEE Int. Conference on Advanced Learning Technologies, 451-452, Doi: 10.1109/ICALT.2009.50.
- Gezici, H., Kocaoğlu, S., Coşgun, E., Yılmazlar, E., ve Tuna, M. (2017). Mekatronik programlarında arduino ile gömülü programlama dersinin robot proje uygulamalı planlanması, *Colleges*, 1.
- Goel, S., & Kathuria, V. (2010). A novel approach for collaborative pair programming, *Journal of Information Technology Education: Research*, 9(1), 183-196.
- Gomes, A., & Mendes, A. J. (2007). An environment to improve programming education, In B. Rachev, A. Smrikarov, and D. Dimov (Eds.), *Proceedings of the 2007 International Conference on Computer Systems and Technologies*, Bulgaria, June 14–15, 88.
- González, R.M., González, P.J. C., & Jiménez-Fernández, C. (2017). Which cognitive abilities underlie computational thinking? Criterion validity of the Computational Thinking Test, *Computers in Human Behavior*, 72, 678-691.
- Grover, S., & Pea, R. (2013). Computational thinking in K–12: A review of the state of the field, *Educational researcher*, 42(1), 38-43.

- Gökoğlu, S. (2017). Programlama eğitiminde algoritma algısı: bir metafor analizi, *Cumhuriyet International Journal of Education*, 6(1), 1-14.
- Gömlüksiz, M. N., ve Erten, P. (2010). Mesleki ve teknik ortaöğretimde modüler öğretim programının uygulanmasında karşılaşılan güçlükler: nitel bir çalışma, *Yüzüncü Yıl Üniversitesi Eğitim Fakültesi Dergisi*, 7(1), 174-198.
- Gupta, K., Stuart, J. A., and Owens, J. D. (2012). A study of persistent threads style GPU programming for GPGPU workloads, In *2012 Innovative Parallel Computing*, 1-14.
- Habitat. (2018). “minik parmaklar geleceği programlıyor”, <https://habitatderneği.org/sona-eren-projeler/minik-parmaklar-gelecegi-programliyor/> (Erişim tarihi: 14 Ekim 2018)
- Haughery, J. R. (2018). Mechatronics and academic success: towards understanding the impacts of age, major, and technical experience, *Agricultural and Biosystems Engineering*, 1-14.
- Herger, L. M., & Bodarky, M. (2015). Engaging students with open source technologies and Arduino, *2015 IEEE Integrated STEM Education Conference*, 27-32.
- Hertzog, P. E., & Swart, A. J. (2016). Arduino—Enabling engineering students to obtain academic success in a design-based module, *2016 IEEE Global Engineering Education Conference (EDUCON)*, 66-73.
- Howell, D.C. (2002). *Statistical Methods for Psychology* (5th ed.), Pacific Grove CA: Duxbury.
- Hsiao, I. H., & Brusilovsky, P. (2011). The role of community feedback in the student example authoring process: An evaluation of annotex, *British Journal of Educational Technology*, 42(3), 482-499.
- Hubwieser, P., Giannakos, M. N., Berges, M., Brinda, T., Diethelm, I., Magenheimer, J., & Jasute, E. (2015). A global snapshot of computer science education in K-12 schools. *Proceedings of the 2015 ITiCSE on working group reports*, 65-83.

- Huck, S. W., & McLean, R. A. (1975). Using a repeated measures ANOVA to analyze the data from a pretest-posttest design: A potentially confusing task, *Psychological Bulletin*, 82(4), 511.
- Hui, T. H., & Umar, I. N. (2011). Pair programming and lss in computing education: its impact on students' performances, *Online Submission*, US-China Education Review B5, 613-626.
- Hyde, J. S. (2001). Reporting effect sizes: The roles of editors, textbook authors, and publication manuals, *Educational and Psychological Measurement*, 61(2), 225-228.
- Jang, Y., Lee, W., & Kim, J. (2015). Assessing the usefulness of object-based programming education using Arduino, *Indian Journal of Science and Technology*, 8(S1), 89-96.
- Jenkins, T. (2002). On the difficulty of learning to program, *Proceedings of the 3rd Annual Conference of the LTSN Centre for Information and Computer Sciences*, Vol. 4, 53-58.
- Jiau, H. C., Chen, J. C., & Ssu, K. F. (2009). Enhancing self-motivation in learning programming using game-based simulation and metrics, *IEEE Transactions on Education*, 52(4), 555-562.
- Jones, J. C. (1970). Design methods: seeds of human futures, *Wiley-Interscience*, London.
- Jonassen, D. H. (2011). Design Problems for Secondary Students, *National Center for Engineering and Technology Education*, 1-6.
- Junior, L. A., Neto, O. T., Hernandez, M. F., Martins, P. S., Roger, L. L., & Guerra, F. A. (2013). A low-cost and simple arduino-based educational robotics kit. Cyber Journals: Multidisciplinary Journals in Science and Technology, *Journal of Selected Areas in Robotics and Control (JSRC)*, December edition, 3(12), 1-7.
- Kafai, Y. B., & Burke, Q. (2015). Constructionist gaming: Understanding the benefits of making games for learning, *Educational psychologist*, 50(4), 313-334.

- Kafai, Y. B., & Burke, Q. (2013). The social turn in K-12 programming: moving from computational thinking to computational participation, *Proceeding of the 44th ACM technical symposium on computer science education*, 603-608.
- Kafai, Y. B., Fields, D. A., & Giang, M. T. (2016). Coding by choice: A transitional analysis of social participation patterns and programming contributions in the online Scratch community, In *Mass collaboration and education*, 209-240.
- Kahn, K., Sendova, E., Sacristán, A. I., & Noss, R. (2011). Young students exploring cardinality by constructing infinite processes, *Technology, Knowledge and Learning*, 16(1), 3–34.
- Kahraman, B. (2015). “Avustralya’da ilkokullarda programcılık eğitimi veriliyor”, <https://www.webtekno.com/sektorel/avustralya-da-ilkokullarda-programcilik-egitimi-veriliyor-h10859.html> (Erişim tarihi: 14 Ekim 2018)
- Kalelioğlu, F. (2015). A new way of teaching programming skills to K-12 students: Code. org. *Computers in Human Behavior*, 52, 200-210.
- Kalelioglu, F., Gülbahar, Y., ve Kukul, V. (2016). A framework for computational thinking based on a systematic research review, *Baltic Journal of Modern Computing*, 4(3), 583.
- Kalkınma. (2014). Mesleki eğitimin yeniden yapılandırılması çalışma grubu raporu, Kalkınma Bakanlığı, Ankara.
- Karabak, D., ve Güneş, A. (2013). Ortaokul birinci sınıf öğrencileri için yazılım geliştirme alanında müfredat önerisi, *Eğitim ve Öğretim Araştırmaları Dergisi*, 21(2-3), 163-169.
- Karacı, A. (2016). Investigation Of Attitudes Towards Computer Programming In Terms Of Various Variables. *International Journal of Programming Languages and Applications (IJPLA)*, Vol.6, No.1/2,1-9.
- Karim, M. E., Lemaignan, S., & Mondada, F. (2015). A review: Can robots reshape K-12 STEM education?, *2015 IEEE International Workshop on Advanced Robotics and its Social Impacts (ARSO)*, 1-8.

- Katai, Z., & Toth, L. (2010). Technologically and artistically enhanced multi-sensory computer-programming education, *Teaching and teacher education*, 26(2), 244-251.
- Keppel, G. (1991). Design and analysis: A researcher's handbook, Prentice-Hall, Inc.
- Kerlinger, F. H. (1964). Foundations of behavioural research: educational and psychological inquiry, New York: Holt, Rinehart & Winston.
- Kert, S. B., ve Uğraş, T. (2009). Programlama eğitiminde sadelik ve eğlence: Scratch örneğın, In *The First International Congress of Educational Research*, Çanakkale, Turkey.
- Kesici, T., ve Kocabaş, Z. (2007). Bilgisayar 2 Ders Kitabı (2. Baskı).
- Keskin, S., ve Kılıç, D. (2016). Ortaokul 7. sınıf matematik dersinde olasılık konusunun işbirlikli öğrenme yöntemiyle öğretiminin öğrencilerin akademik başarılarına etkisi/the effect of teaching probability in mathematics through, *Atatürk Üniversitesi Sosyal Bilimler Enstitüsü Dergisi*, 20(3), 1173-1183.
- Kim, S., Chung, K., & 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.
- Kinnunen, P., McCartney, R., Murphy, L., & Thomas, L. (2007). Through the eyes of instructors: a phenomenographic investigation of student success, *Proceedings of the third international workshop on Computing education research*, 61-72.
- Kirk, R. E. (1996). Practical significance: A concept whose time has come, *Educational and psychological measurement*, 56(5), 746-759.
- Kobsiripat, W. (2015). Effects of the media to promote the scratch programming capabilities creativity of elementary school students, *Procedia-Social and Behavioral Sciences*, 174, 227-232.
- Koç, A., ve Büyük, U. (2013) Fen ve teknoloji eğitiminde teknoloji tabanlı öğrenme: Robotik uygulamaları, *Türk Fen Eğitimi Dergisi*, 10(1), 139-155.

- Konecki, M. (2014). Problems in programming education and means of their improvement, *DAAAM international scientific book, 2014*, 459-470.
- Korkmaz, Ö., ve Altun, H. (2013). Engineering and ceit student's attitude towards learning computer programming. *The Journal of Academic Social Science Studies International Journal of Social Science*, 6(2), 1169-1185.
- Köse, U., Koç, D., ve Yücesoy, S. A. (2013). An augmented reality based mobile software to support learning experiences in computer science courses, *Procedia Computer Science*, 25, 370-374.
- Köse, U., ve Tüfekçi, A. (2015). Algoritma ve akis semasi kavramlarının öğretiminde akıllı bir yazılım sistemi kullanımı, *Pegem Eğitim ve Öğretim Dergisi= Pegem Journal of Education and Instruction*, 5(5), 569.
- Lahtinen, E., Ala-Mutka, K., & Järvinen, H.M. (2005). A study of the difficulties of novice programmers, *Annual Joint Conference Integrating Technology into Computer Science Education: Proceedings of the 10th annual SIGCSE conference on Innovation and technology in computer science education*, Monte de Caparica, Portugal, 14-18.
- Lin, J. M. C., & Liu, S. F. (2012). An investigation into parent-child collaboration in learning computer programming, *Educational Technology & Society*, 15(1), 162–173.
- Linn, M. C. & Clancy, M. J. (1992). Can experts' explanations helps students develop program design skills?, *International Journal of Man-Machine Studies*, 36, 4 (Apr. 1992), 511- 551.
- Lo, J., & Paulos, E. (2014). ShrinkyCircuits: sketching, shrinking, and formgiving for electronic circuits, In *Proceedings of the 27th annual ACM symposium on User interface software and technology*, 291-299.
- Lye, S. Y., & Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12?, *Computers in Human Behavior*, 41, 51-61.
- Maloney, J. H., Peppler, K., Kafai, Y., Resnick, M., & Rusk, N. (2008). Programming by choice: Urban youth learning programming with scratch. In *Proceedings of the 39th SIGCSE Technical Symposium on Computer*

Science Education, Portland, OR, USA, March 12-15, 2008, SIGCSE '08, 367–371.

Margolis, P. A., DeWalt, D. A., Simon, J. E., Horowitz, S., Scoville, R., Kahn, N., & Miles, P. (2010). Designing a large-scale multilevel improvement initiative: The improving performance in practice program, *Journal of Continuing Education in the Health Professions*, 30(3), 187-196.

Mauch, E. (2001). Using technological innovation to improve the problem-solving skills of middle school students: educators' experiences with the lego mindstorms robotic invention system, *The Clearing House: A Journal of Educational Strategies, Issues and Ideas*, 74(4), 211–213.

Mauchly, J. W. (1940). Significance test for sphericity of a normal n-variate distribution, *The Annals of Mathematical Statistics*, 11(2), 204-209.

McMillan, J. H., & Schumacher, S. (2010). Research in education: evidence-based inquiry, MyEducationLab Series, Pearson.

MEB. (2011). <http://mtegm.meb.gov.tr/tr/dokumanlar/calistay/files/assets/basic-html/page51.html> (Erişim tarihi: 28 Ekim 2018)

MEB. (2012a). Ortaokul ve imam hatip ortaokulu bilişim teknolojileri ve yazılı dersi (5.6.7.ve 8. Sınıflar) öğretim programı, Talim ve Terbiye Kurulu Başkanlığı, Ankara.

MEB. (2012b). Mesleki ve teknik eğitim çalıştayı, Milli Eğitim Bakanlığı, Antalya.

MEB. (2014). Türkiye mesleki ve teknik eğitim strateji belgesi ve eylem planı 2014-2018, <http://www.resmigazete.gov.tr/eskiler/2016/12/20161209-5.htm>, (Erişim tarihi: 28 Ekim 2018)

MEB. (2018a). Bilişim teknolojileri ve yazılım dersi öğretim programı (Ortaokul 5 ve 6. Sınıflar), Talim ve Terbiye Kurulu Başkanlığı, Ankara.

MEB. (2018b). Bilişim teknolojileri ve yazılım dersi öğretim programı (ortaokul ve imam hatip ortaokulu 7 ve 8. sınıflar), Talim ve Terbiye Kurulu Başkanlığı, Ankara.

- MEB. (2018c) Bilgisayar bilimi dersi öğretim programı (kur 1 ve kur 2), Talim ve Terbiye Kurulu Başkanlığı, Ankara.
- MEB. (2018d). Bilişim teknolojileri alanı alan ortak yeterlikleri, Talim ve Terbiye Kurulu Başkanlığı, Ankara.
- MEB. (2018e). Bilişim teknolojileri ve yazılım dersi öğretim programı (ilkokul 1, 2, 3 ve 4. sınıflar), Talim ve Terbiye Kurulu Başkanlığı, Ankara.
- MEGEP. (2011). Kodlamaya hazırlık, Milli Eğitim Bakanlığı, Ankara.
- Mellis, D., Banzi, M., Cuartielles, D., & Igoe, T. (2007). Arduino: An open electronic prototyping platform, In *Proc. Chi*, Vol. 2007, 1-11.
- Memurlar. (2018). Teknik eğitim fakültesi kapatıldı, <https://www.memurlar.net/haber/153308/16-teknik-egitim-fakultesi-kapatildi.html> (Erişim tarihi: 28 Ekim 2018)
- Merino, P. P., Ruiz, E. S., Fernandez, G. C., & Gil, M. C. (2016). A Wireless robotic educational platform approach, *2016 13th International Conference on Remote Engineering and Virtual Instrumentation (REV)*, 145-152.
- Miliszewska, I., & Tan, G. (2007). Befriending computer programming: A proposed approach to teaching introductory programming, *Informing Science: International Journal of an Emerging Transdiscipline*,4(1), 277-289.
- Molenda, M., Reigeluth, C. M., & Nelson, L. M. (2003). Instructional design.,In (L. Nadel, Ed.) *Encyclopedia of cognitive science*.
- Monroy-Hernández, A. & Resnick, M. (2008). Empowering kids to create and share programmable media, *ACM interactions*, vol. 15, no. 2, 50-53.
- Naps, T., Rößling, G., Almstrum, V., Dann, W., Fleischer, R., Hunhausen, C., MsNally, M., Hundhausen, C., Rodger, S., Korhonen, A., Velazquez-Iturbide, J. A. & Malmi, L. (2003). Exploring the role of visualization and engagement in computer science education, *SIGCSE Bulletin*,35(2),131–152.

- Nowaczyk, R. H. (1984). The relationship of problem-solving ability and course performance among novice programmers, *International Journal of Man-Machine Studies*, 21(2), 149-160.
- Numanoğlu, M., ve Keser, H. (2017). Programlama öğretiminde robot kullanımı- mbot örneği, *Bartın Üniversitesi Eğitim Fakültesi Dergisi*,6(2), 497.
- Nurazian M.D., Suzana B., Haslizatul F.M.H., & Ismassabah I. (2007). *Development of Instruments for Measuring Learning Attitudes Towards Computer Programming*. In: National Conference on Programming Science (ATUR 07), 5 December 2007, Pacific KLIA, Selangor.
- Olejnik, S., & Algina, J. (2003). Generalized eta and omega squared statistics: measures of effect size for some common research designs, *Psychological methods*,8(4), 434.
- Omar, A. M., Mahzan, N. N., Rimon, L., Noor, S. M., & Rosselan, M. Z. (2017). Design and development of an Arduino based data logger for photovoltaic monitoring system, *Int. J. Simul. Syst. Sci. Technol*,17(41), 15-1.
- Öndeş, Ö. (2016). “İngiltere ve ABD’de kodlama eğitimi”, <http://www.hurriyet.com.tr/egitim/ingiltere-ve-abdde-kodlama-egitimi-40061604> (Erişim tarihi: 14 Ekim 2018)
- Özdemir, A. (2015). “Kodlama dersi gelişmiş ülkelerin okullarında en önemli ders- oldu”, <http://www.kpsscafe.com.tr/egitim-haberleri/kodlama-dersi-gelistmis-ulkelerin-okullarinda-en-onemli-ders-oldu-h50358.html> (Erişim tarihi: 14 Ekim 2018)
- Özdemir, E., ve Uyangör, S. M. (2011). Matematik Eğitimi İçin Bir Öğretim Tasarımı Modeli, *Education Sciences*, 6(2), 1786-1796.
- Özdemirci, E., Ersin, Ç., ve Canal, M. R. (2019). Arduino Uno Uygulama Setinin Gerçekleştirilmesi, *Mehmet Akif Ersoy Üniversitesi Fen Bilimleri Enstitüsü Dergisi*, 8(Özel 1), 127-133.
- Özdiñç, F., ve Altun, A. (2014). Bilişim teknolojileri öğretmeni adaylarının programlama sürecini etkileyen faktörler, *İlköğretim Online*, 13(4), 1531-1541.

- Özyurt Ö., ve Özyurt, H. (2015). A study for determining computer programming students' attitudes towards programming and their programming self-efficacy, *Journal of Theory and Practice in Education*, 11(1), 51-67.
- Pea, R. D. (1986). Language-independent conceptual “bugs” in novice programming, *Journal of educational computing research*, 2(1), 25-36.
- Przybylla, M., & Romeike, R. (2014). Overcoming issues with students' perceptions of informatics in everyday life and education with physical computing, *Informatics in School: Situation, Evolution and Perspectives, ISSEP, 2014*, 9-20.
- Pong, W. (2015). Assessing the impact of research experiences on the success of underrepresented community college engineering students, *age*, 26, 1.
- Qian, Y., & Lehman, J. D. (2016). Correlates of success in introductory programming: a study with middle school students, *Journal of Education and Learning*, 5(2), 73-83.
- Qiu, K., Buechley, L., Baafi, E., & Dubow, W. (2013). A curriculum for teaching computer science through computational textiles, In *Proceedings of the 12th International Conference on Interaction Design and Children*, 20-27.
- Reas, C., Fry, B., & Maeda, J. (2014). *Processing: a programming handbook for visual designers*, MIT Press, Cambridge, Massachusetts, USA.
- Rees, A., García-Peñalvo, F. J., Jormanainen, I., Tuul, M., & Reimann, D. (2016). An overview of the most relevant literature on coding and computational thinking with emphasis on the relevant issues for teachers, KA2 project “TACCLE 3 – Coding” (2015-1-BE02-KA201-012307).
- Resinovic, B. (2015). The use of Nao, a humanoid robot, in teaching computer programming, *The Proceedings of International Conference on Informatics in Schools: Situation, Evolution and Perspectives—ISSEP 2015*, 63.
- Robey, R. R., Schultz, M. C., Crawford, A. B., & Sinner, C. A. (1999) Single-subject clinical-outcome research: Designs, data, effect sizes, and analyses, *Aphasiology*, 13(6), 445-473.

- Robins, A., Rountree, J., & Rountree, N. (2003). Learning and teaching programming: A review and discussion. *Computer Science Education*, 13(2), 137-172.
- Rodriguez-Sánchez, M. C., Torrado-Carvajal, A., Vaquero, J., Borromeo, S., & Hernandez-Tamames, J. A. (2016). An embedded systems course for engineering students using open-source platforms in wireless scenarios, *IEEE Transactions on Education*, 59(4), 248-254.
- Rogerson, C., & Scott, E. (2010). The fear factor: How it affects students learning to program in a tertiary environment, *Journal of Information Technology Education: Research*, 9, 147-171.
- Roth, W., & Roychoudhury, A. (1994). Science discourse through collaborative concept mapping: new perspectives for the teacher, *International Journal of Science Education*, 6 (4), 437-455.
- Rubio, M. A., Hierro, C. M., & Pablo, A. P. D. M. (2013). July, Using arduino to enhance computer programming courses in science and engineering, *Proceedings of EDULEARN13 conference*, IATED Barcelona, Spain, 1-3.
- Rusk, N., Resnick, M., Berg, R., & Pezalla-Granlund, M. (2008). New pathways into robotics: Strategies for broadening participation, *Journal of Science Education and Technology*, 17(1), 59-69.
- Şabanoviç, A., ve Yannier, S. (2003). Robotlar: Sosyal etkileşimli makineler, *TÜBİTAK Bilim Teknik Dergisi*, 1-9.
- Sadler, J., Durfee, K., Shluzas, L., & Blikstein, P. (2015). Bloctopus: a novice modular sensor system for playful prototyping, In *Proceedings of the ninth international conference on tangible, embedded, and embodied interaction*, 347-354.
- Shaver, J. P. (1993). What statistical significance testing is, and what it is not, *The Journal of Experimental Education*, 61(4), 293-316.
- Saleiro, M., Carmo, B., Rodrigues, J. M., & Buf, J. H. (2013). A low-cost classroom-oriented educational robotics system, *International Conference on social robotics*, 74-83.

- Saygıner, Ş., ve Tüzün, H. (2017). Programlama Eğitiminde Yaşanan Zorluklar Ve Çözüm Önerileri, Uluslararası Bilgisayar Ve Öğretim Teknolojileri Sempozyumu, Malatya.
- Sezer, A., ve Tokcan, H. (2003). İş birliğine dayalı öğrenmenin coğrafya dersinde akademik başarı üzerine etkisi, *Gazi Üniversitesi Gazi Eğitim Fakültesi Dergisi*, 23(3), 227-242.
- Simon, S., Fincher, S., Robins, A., Baker, B, Box, I., Cutts, Q., de Raadt., M., Haden, P., Hamer, J., Hamilton, M., Lister, R., Petre, M., Sutton, K., Tolhurst, D., & Tutty, J. (2006). *Predictors of success in a first programming course, Proceedings of the 8th Australian conference on Computing education*, January 16 – 19, Hobart, Tasmania, Australia, 189-196.
- Sleeman, D., Putnam, R. T., Baxter, J. A., & Kuspa, L. K. (1986). A summary of misconceptions of high school Basic programmers. *Journal of Educational Computing Research*, 2(4), 459-472.
- Sloane, K. D. & Linn, M. C. (1988). Instructional conditions in Pascal programming classes, R. E. Mayer (Ed.), *Teaching And Learning Computer Programming*, 207–235.
- Smith, M. B. (1968). Attitude change. *International encyclopedia of the social sciences*, 1, 458–67.
- Sohn, W. (2014). Design and Evaluation of Computer Programming Education Strategy using Arduino. *Advanced Science and Technology Letters*, 66(1), 73-77.
- SSRVM. (2007). “Model curriculum and teaching material for K-12 Indian schools.”, <http://www.it.iitb.ac.in/~sri/papers/SSRVM-CS-March07.pdf> (Erişim tarihi: 14 Ekim 2018)
- Stage-Co. (2017). “Coderjojo-Türkiye”, <https://www.stage-co.com/coderjojo-turkiye/>. (Erişim tarihi: 14 Ekim 2018)
- Sterritt, R., Hanna, P., and Campbell, J. (2015). Reintroducing programming to the school environment.
- Şimşek, A. (2011). Öğretim tasarımı. Ankara: Nobel Akademik Yayıncılık.

- Şimşek, A. (2013). Öğretim tasarımı ve modelleri. Çağıltay, K. ve Göktaş, Y. (Ed.), Öğretim teknolojilerinin temelleri: Teoriler, araştırmalar, eğilimler içinde (99-116), Ankara: Pegem Akademi.
- Şimşek, N. (2002). BIG 16 öğrenme biçimleri envanteri, *Eğitim Bilimleri ve Uygulama*, 1(1), 33-47.
- Tabachnick, B. G., & Fidell, L. S. (2001). Using multivariate statistics (4th ed.). Needham Heights, MA: Allyn and Bacon.
- Tan, P. H., Ting, C. Y., & Ling, S. W. (2009). Learning difficulties in programming courses: Undergraduates' perspective and perception, 2009 *International Conference on Computer Technology and Development*, Kota Kinabalu, Malaysia, 42-46.
- Tan, W. L., Venema, S., & Gonzalez, R. (2017). Using Arduino to Teach Programming to First-Year Computer Science Students, *International Association for Development of the Information Society*, 98-104.
- 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.
- Uçar, C., ve Özerbaş, M. A. (2013). Mesleki ve teknik eğitimin dünyadaki ve Türkiye'deki konumu, *Eğitim ve Öğretim Araştırmaları Dergisi*, 2(2), 242-253.
- Uzgun, B. Ç., ve Aykaç, N. (2016). Bilişim Teknolojileri ve Yazılım Dersi Öğretim Programının Öğretmen Görüşlerine Göre Değerlendirilmesi (Ege Bölgesi Örneği)/The Evaluation of Information Technologies and Software Course's Curriculum According to the Teacher's Ideas, *Mustafa Kemal Üniversitesi Sosyal Bilimler Enstitüsü Dergisi*, 13(34), 273-297.
- Ünal, E., ve Çakır, H. (2017). Students' Views about the Problem Based Collaborative Learning Environment Supported by Dynamic Web Technologies, *Malaysian Online Journal of Educational Technology*, 5(2), 1-19.

- Vacha-Haase, T. (2001). Statistical significance should not be considered one of life's guarantees: Effect sizes are needed, *Educational and Psychological Measurement*, 61(2), 219-224.
- Wang, X. M., Hwang, G. J., Liang, Z. Y., & Wang, H. Y. (2017). Enhancing students' computer programming performances, critical thinking awareness and attitudes towards programming: An online peer-assessment attempt, *Journal of Educational Technology & Society*, 20(4), 58-68.
- Weinberg, G. M. (1971). *The psychology of computer programming*, Van Nostrand Reinhold Company, New York.
- Wilson, B. C., & Shrock, S. (2001). Contributing to success in an introductory computer science course: A study of twelve factors, Proceedings of the Thirty-Second SIGCSE Technical Symposium on Computer Science Education, March, Charlotte, North Carolina, USA.
- Witherspoon, E. B., Schunn, C. D., Higashi, R. M., & Shoop, R. (2018). Attending to structural programming features predicts differences in learning and motivation, *Journal of Computer Assisted Learning*, 115-128.
- Woodrow, J. E. (1992). The influence of programming training on the computer literacy and attitudes of preservice teachers, *Journal of research on Computing in Education*, 25(2), 200-219.
- Yadagiri, R. G., Krishnamoorthy, S. P., & Kapila, V. (2015). A blocks-based visual environment to teach robot-programming to K-12 students, *2015 ASEE Annual Conference & Exposition*, 26-17.
- Yağcı, M. (2016). Bilişim teknolojileri (BT) öğretmen adaylarının ve bilgisayar programcılığı (BP) öğrencilerinin programlamaya karşı tutumlarının programlama öz yeterlik algılarına etkisi, *Journal of Human Sciences*, 13(1), 1418-1432.
- Yıldız, A. (2018). “Okullarda kodlama eğitimine yönelik bir uygulama örneği progetiger mufredati Estonya”, <http://anahtar.sanayi.gov.tr/tr/news/okullarda-kodlama-egitimine-yonelik-bir-uygulama-ornegi-progetiger-mufredati-estonya/9702> (Erişim tarihi: 14 Ekim 2018)

- Yılmaz, E. (2014). Arduino nedir, özellikleri nelerdir ve nerelerde kullanılır, <https://www.hataverdi.com/arduino-nedir-ozellikleri-nelerdir-ve-nelerelerde-kullanilir> (Erişim tarihi: 28 Ekim 2018).
- YÖK. (1998). Eğitim fakültesi öğretmen yetiştirme lisans programları, Ankara.
- YÖK. (2011). Türkiye Yükseköğretim Yeterlilikler Çerçevesi Temel Alan Yeterlilikleri Bilgisayar Temel Alan Kodu: 48, Ankara.
- Yörük, S., Dikici, A., ve Uysal, A. (2002). Bilgi toplumu ve Türkiye’de mesleki eğitim, *Fırat Üniversitesi Sosyal Bilimler Dergisi*, 12(2), 299-312.
- Yükseltürk, E., ve Altıok, S. (2016). Investigation of pre-service information technology teachers' game projects prepared with Scratch, *SDU International Journal of Educational Studies*, 3(1), 59-66.
- Yünkül, E., Durak, G., Çankaya, S., ve Abidin, Z. (2017). The effects of scratch software on students’ computational thinking skills, *Necatibey Eğitim Fakültesi Elektronik Fen ve Matematik Eğitimi Dergisi*, 11(2), 502-517.
- Ziatdinov, R., & Musa, S. (2012). Rapid mental computation system as a tool for algorithmic thinking of elementary school students development, *European Researcher*, 25(7), 1105-1110.
- Zheng, L. & Smaldino, S. (2003). Key instructional design elements for distance Education, *The Quarterly Review of Distance Education*, 4(2), 153-166.

EKLER



Ek 1. Akademik Başarı Testi

1. Aşağıdaki değişken isimlerinden hangisi hatalıdır?

- a) ogrNo b) OgrAd c) ogr_ad d) ogr soyad

2. Aşağıdaki yazımlardan hangisi doğrudur?

- a) Console.WriteLine("Boyunuz : {0} Kilonuz : {1}", boy, kilo);
b) Console.WriteLine("Boyunuz : boy Kilonuz : kilo");
c) Console.WriteLine("Boyunuz : [0] Kilonuz : [1]", boy, kilo);
d) Console.WriteLine("Boyunuz : {0} Kilonuz : {1} " + boy + kilo);

3. Mod alma işlemini aşağıdaki operatörlerden hangisi gerçekleştirir?

- a) * b) # c) % d) //

4. Aşağıdakilerden hangisi ilişkisel operatörler arasında yer almaz?

- a) && b) == c) != d) <

5. Veya(OR) mantık operatörü için aşağıdakilerden hangisi kullanılır?

- a) % b) || c) \$ d) ^

6. Bir burs başvurusunda yaşı 20 ile 30 arasında olan kişiler başvuru yapabilir cümlesinin programlamada (söz dizimi) yazılışı, aşağıdaki seçeneklerden hangisidir?

- a) yas <= 20 AND yas <= 30
b) yas <= 30 OR yas <= 20
c) yas <=30 AND yas >= 20
d) yas <= 20 OR yas >= 30

7. A=1;
 C=A+2;
 B=C+8;
 if (B>C) Console.WriteLine(B + " B değeri büyük");
 else if (C<A)
 Console.WriteLine(C + " C değeri büyük");

Yukarıdaki programın ekran çıktısı aşağıdaki seçeneklerden hangisidir?

- a) 10 B değeri büyük b) 11 B değeri büyük
c) 11 C değeri büyük d) 10 C değeri büyük

8. Console.Write("Birinci sayıyı giriniz: ");
 int sayi1 = Convert.ToInt32(Console.ReadLine());
 Console.Write("İkinci sayıyı giriniz: ");
 int sayi2 = Convert.ToInt32(Console.ReadLine());
 int toplam= sayi1+sayi2;
 Console.WriteLine(toplam);

Yukardaki programın açıklaması aşağıdakilerden hangisidir?

- a) Sabit iki sayının toplamını bulup yazdıran program
- b) Klavyeden girilen iki sayının ortalamasını bulup yazdıran program
- c) Sabit iki sayının ortalamasını bulup yazdıran program
- d) Klavyeden girilen iki sayının toplamını bulup yazdıran program

```
Console.Write("Bir sayı giriniz:");
int sayi = Convert.ToInt32(Console.ReadLine());
.....
Console.WriteLine("{0} sayısı hem 3'e hem de 5'e kalansız bölünebilir",sayi);
Else Console.WriteLine("{0} sayısı hem 3'e hem de 5'e kalansız bölünemez",
sayi);
```

Klavyeden girilen sayının hem 3'e hem de 5'e kalansız bölünüp bölünemediğini ekrana yazan programı yapabilmek için yukarıdaki programda boş bırakılan yere aşağıdakilerden hangisi yazılmalıdır?

- a) `if((sayi % 3 == 0) && (sayi / 5 ==0))`
- b) `if((sayi / 3 == 0) && (sayi / 5 ==0))`
- c) `if((sayi / 3 == 0) && (sayi % 5 ==0))`
- d) `if((sayi % 3 == 0) && (sayi % 5 ==0))`

10. Aşağıda verilen operatörlerden hangisinin işlem önceliği en yüksektir?

- a) *
- b) /
- c) %
- d) ()

11. `int puan = Convert.ToInt32(Console.ReadLine());`

`if (puan >= 0 && puan < 25)`

`Console.WriteLine("Puanınızın 5'lik sistemdeki karşılığı 0'dır");`

`else if (puan >= 25 && puan < 45)`

`Console.WriteLine("Puanınızın 5'lik sistemdeki karşılığı 1'dir");`

`else if (puan >= 45 && puan < 55)`

`Console.WriteLine("Puanınızın 5'lik sistemdeki karşılığı 2'dir");`

`else if (puan >= 55 && puan < 70)`

`Console.WriteLine("Puanınızın 5'lik sistemdeki karşılığı 3'tür");`

`else if (puan >= 70 && puan < 85)`

`Console.WriteLine("Puanınızın 5'lik sistemdeki karşılığı 4'tür");`

`else if (puan >= 85 && puan <= 100)`

`Console.WriteLine("Puanınızın 5'lik sistemdeki karşılığı 5'tir");`

`else Console.WriteLine("Hatalı puan girdiniz. Puanınız 0 ile 100 arasında olmalıdır.");`

Yukarıdaki programa göre tabloda verilen değerler için yazılan ekran çıktılarından doğru olanların verildiği seçenek aşağıdakilerden hangisidir?

	Girilecek Değerler	Ekran Çıktısı
I	12	Puanınızın 5'lik sistemdeki karşılığı 1'dir
II	55	Puanınızın 5'lik sistemdeki karşılığı 3'tür
III	78	Puanınızın 5'lik sistemdeki karşılığı 3'tür
IV	96	Puanınızın 5'lik sistemdeki karşılığı 5'tir

- a) Yalnız I
- b) I ve II
- c) II ve IV
- d) I, II ve IV

12. Veri tipi "short" olarak tanımlanan bir değişkene aşağıda verilen değerlerden hangisi atanamaz?

- a) 250 b) 3008 c) 40523 d) 31200

13. Aşağıda verilen for döngüsü tanımlamalarından hangisinde döngü sonsuz bir döngüye girer?

- a) for(int i=0;i<100;i++)
b) for(int i=0;i<100;i--)
c) for(int i=100;i>0;i--)
d) for(int i=0;i<=100;i=i+5)

```
14. int renk = Convert.ToInt32(Console.ReadLine());  
    switch (renk)  
    {  
        case 1: Console.WriteLine("Kırmızı"); break;  
        case 2: Console.WriteLine("Yeşil"); break;  
        case 3: Console.WriteLine("Mavi"); break;  
        default : Console.WriteLine("Geçerli bir değer giriniz"); break;  
    }
```

Yukarıda verilen programda klavyeden girilen "4" değerine göre ekran çıktısı aşağıdakilerden hangisi olur?

- a) Kırmızı b) Yeşil c) Mavi d) Geçerli bir değer giriniz

```
15. int i=1;  
    do{  
        Console.WriteLine(i);  
        i = i + 2;  
    } while (i < 20);
```

Yukarıda verilen program sonunda elde edilen sonuç aşağıdakilerden hangisidir?

- a) 1' de 10' a kadar olan tek sayıları ekrana yazdırır
b) 1' de 10' a kadar olan çift sayıları ekrana yazdırır
c) 1' den 20' ye kadar olan tek sayıları ekrana yazdırır
d) 1' den 20' ye kadar olan çift sayıları ekrana yazdırır

```
16. int i=0;  
    while (i <= 20)  
    {  
        Console.WriteLine(i);  
        i = i + 2;  
    }
```

Yukarıda verilen program sonunda elde edilen sonuç aşağıdakilerden hangisidir?

- a) 0' de 10' a kadar olan tek sayıları ekrana yazdırır
b) 0' de 10' a kadar olan çift sayıları ekrana yazdırır
c) 0' den 20' ye kadar olan tek sayıları ekrana yazdırır
d) 0' den 20' ye kadar olan çift sayıları ekrana yazdırır

```

17. int i,k;
    for (i = 1; i <=10; i++)
    { Console.WriteLine("-{0} ve Katları-",i);
      Console.WriteLine("-----");
      for (k = 1; k <= 10; k++)
      { int carpim = i * k;
        Console.WriteLine("{0} x {1} = {2}", i, k, carpim);
      }
      Console.WriteLine("-----");
    }

```

Yukarda verilen program sonunda elde edilen sonuç aşağıdakilerden hangisidir?

- a) 1'den 10'a kadar olan sayılar için çarpım tablosunu ekrana yazdırır
- b) 0'dan 10'a kadar olan sayılar için çarpım tablosunu ekrana yazdırır
- c) 0'dan 10'a kadar olan çift sayılar için çarpım tablosunu ekrana yazdırır
- d) 0'dan 10'a kadar olan tek sayılar için çarpım tablosunu ekrana yazdırır

18.Ekrana 10 kere "Merhaba Dünya" yazdıran program aşağıdakilerden hangisidir?

- a)

```
int i = 1;
while (i<=10)
{ Console.WriteLine("Merhaba Dünya");
  i++;
}
```
- b)

```
int i = 1;
while (i<=20)
{ Console.WriteLine("Merhaba Dünya");
  i++;
}
```
- c)

```
int i = 1;
while (i<=30)
{ Console.WriteLine("Merhaba Dünya");
  i++;
}
```
- d)

```
int i = 1;
while (i<=40)
{ Console.WriteLine("Merhaba Dünya");
  i++;
}
```

```

19. char i='a';
    do {
        Console.WriteLine(i);
        .....
    } while (i <= 'z')

```

a' dan z' ye kadar olan sayıları ekrana yazdırmak için boş bırakılan yere aşağıda verilenlerden hangisi getirilmelidir?

- a) i++;
- b)i--;
- c) i**;
- d)i==;

```
20. for (.....X döngüsü.....)
    {
        for (.....Y döngüsü.....)
        {
            Console.Write("* ");
        }
        Console.WriteLine();
    }
```



Yandaki ekran çıktısını elde etmek için yukardaki programda boş bırakılan yerlere yazılması gereken kodlar aşağıdakilerden hangisinde doğru verilmiştir?

- | | X döngüsü | Y döngüsü |
|----|------------------------|------------------------|
| a) | int k = 1; k <= 4; k++ | int m=1; m <= k; m++ |
| b) | int k = 3; k <= 4; k++ | int m=2; m <= k; m++ |
| c) | int k = 1; k <= 5; k++ | int m=1; m <= k; m++ |
| d) | int m=1;m <= k; m++ | int k = 1; k <= 4; k++ |

Ek 2. Programlamaya Yönelik Tutum Ölçeği

Sevgili öğrenciler, aşağıda hazırlanmış ölçek maddeleri “Öğrencilerin Programlamaya Yönelik Tutumu” ‘nu ölçmeyi hedeflemektedir. Sizden istenilen maddelere içtenlikle cevap vermenizdir. Elde edilecek bilgiler araştırmanın amacı dışında kullanılmayacak ve araştırmacılar dışında kimse bu bilgilere ulaşamayacaktır. Yardımlarınız ve katkılarınız için teşekkür ederiz.

Madde No	Madde	Kesinlikle Katılmıyorum	Katılmıyorum	Karasızım	Katılıyorum	Kesinlikle Katılıyorum
1	Bir programlama dilinde komut yazarken yazım (sintaks) hatalarıyla uğraşmayı sevmiyorum.	1	2	3	4	5
2	Programlamanın, karar verme becerilerimi geliştireceğini düşünüyorum.	1	2	3	4	5
3	Programlama öğrenmek çok zevklidir.	1	2	3	4	5
4	Programlama derslerinde eğlenceli zaman geçiririm	1	2	3	4	5
5	Programlama diline özgü kavramlar (değişken, döngü, dizi, fonksiyon vb) soyut olduğu için bana karmaşık görünüyor.	1	2	3	4	5
6	Programlamanın, problemlere farklı yönlerden bakabilme yeteneğimi geliştireceğini düşünüyorum.	1	2	3	4	5
7	Programlama yaparken çok eğlenirim.	1	2	3	4	5
8	Programlama sorunlarını çözmekten zevk alıyorum.	1	2	3	4	5
9	Programlama ile yeni bir ürün geliştirmek hoşuma gider	1	2	3	4	5
10	Programlamanın, matematiksel düşünme becerilerimi geliştireceğine inanıyorum.	1	2	3	4	5
11	Programlama yaparken olaylar arasındaki ilişkileri görebilme bana zevk verir.	1	2	3	4	5
12	Programlamadaki hataları ayıklama işlemi çok sıkıcıdır.	1	2	3	4	5
13	Programlama yaparken kendimi iyi hissediyorum.	1	2	3	4	5
14	Bir programlama problemini çözmek için gerekli olan bilgiyi araştırırken zevk alırım.	1	2	3	4	5

Ek 3. Etik Kurulu Karar Belgesi



**EGE ÜNİVERSİTESİ FEN ve MÜHENDİSLİK BİLİMLERİ
BİLİMSEL ARAŞTIRMA VE YAYIN ETİĞİ KURULU
KARAR BELGESİ
(Araştırma Dosyası)**

ARAŞTIRMACININ ADI SOYADI / KURUMU	Gönül ALTAY / Polinas Mesleki ve Teknik Anadolu Lisesi
DANIŞMANIN ADI SOYADI / KURUMU	Yrd.Doç.Dr. Tank KIŞLA / Eğitim Fakültesi
ARAŞTIRMANIN TÜRÜ	<input type="checkbox"/> Lisans Bitirme Tezi <input checked="" type="checkbox"/> Yüksek Lisans Tezi <input type="checkbox"/> Doktora Tezi <input type="checkbox"/> Özgün Araştırma <input type="checkbox"/> Diğer (belirtiniz) _____
ARAŞTIRMANIN BAŞLIĞI	Arduino Kullanımının Lise Öğrencilerinin Akademik Başarılarına ve Programlamaya Yönelik Tutumlarına Olan Etkisi
BİLİRKİŞİ GÖRÜŞÜ	YOK
KARARIN ALINDIĞI TOPLANTI TARİHİ	05.07.2017
TOPLANTI / KARAR SAYISI	02 / 01
PROTOKOL NO:	191-2017
KARAR	Araştırma, OYBİRLİĞİ ile etik açıdan uygun bulunmuştur.

Prof.Dr. N. ÜRKÜ KARABAY YAVAŞOĞLU
Kurul Başkanı

Prof.Dr. Şebnem TAYMAN
Kurul Başkan Yardımcısı

Prof.Dr. Günnur KOÇAR
Kurul Üyesi

Prof.Dr. Aynur LÖK
Kurul Üyesi

Prof.Dr. Burçin ÇOKBYAL
Kurul Üyesi

(Toplantıda bulunmadı)
Prof.Dr. Güven ÖZDEMİR
Kurul Üyesi

Prof.Dr. Sena TIMUR
Kurul Üyesi

Ek 4. Kaymakamlık Oluru



T.C.
YUNUSEMRE KAYMAKAMLIĞI
İlçe Milli Eğitim Müdürlüğü

Sayı : 77868257-300-E.5667054
Konu : Programlama Temelleri Tezi

24/04/2017

KAYMAKAMLIK MAKAMINA

İlgi : Bilişim Teknolojileri Öğretmeni Gönül ALTAY'ın 19.04.2017 tarihli dilekçesi.

İlçemiz Polinas Mesleki ve Teknik Anadolu Lisesi Bilişim Teknolojileri Öğretmeni Gönül ALTAY Müdürlüğümüze vermiş olduğu dilekçesinde; *Ege Üniversitesi Eğitim Fakültesi Bilgisayar ve Öğretim Teknolojileri Eğitim Anabilim Dalı'nda devam etmekte olan Yüksek Lisans eğitimini devam ettirdiğini, lise öğrencilerinin "Programlama Temelleri dersindeki Arduino kullanımının Akademik Başarılarına ve Programlamaya Yönelik Tutumlarına Olan Etkisi" başlıklı tezinin uygulamalarını Polinas Mesleki ve Teknik Anadolu Lisesi 10.sınıf programlama temelleri dersine kayıtlı öğrenciler ile birlikte yürütmek istediğini* belirtmektedir.

Adı geçen öğretmenin ekli listedeki program kapsamında söz konusu tez eğitimini yapması sorumluluğu okul idaresinde olması kaydıyla Şubemizce uygun görülmektedir.

Makamlarınızca da uygun görüldüğü takdirde Olurlarınıza arz ederim.

Recep ŞENGÜL
İlçe Milli Eğitim Müdürü

Ek: İlgi Dilekçe ve Eki (9 sayfa)

OLUR
24/04/2017

Ahmet ERDOĞDU
Kaymakam

T.C.
YUNUSEMRE KAYMAKAMLIĞI
İlçe Milli Eğitim Müdürlüğü

Sayın Gönül ALTAY
Polinas Mesleki ve Teknik Anadolu Lisesi Bilişim Teknolojileri Öğretmeni

Müdürlüğümüze vermiş olduğunuz ilgi dilekçenize istinaden **Kaymakamlık Makamından alınan Olur** yukarıya çıkarılmıştır.

Bilgilerinizi rica ederim.

Recep ŞENGÜL
İlçe Milli Eğitim Müdürü

Merkez Efendi Mah. 3910 sokak No:4 Kat:3 Yunusemre/MANİSA
Elektronik Ağ: www.yunusemre.meb.gov.tr
e-posta: http://yunusemre45@meb.gov.tr

Ayrıntılı bilgi için: Abdullah ŞANLI- Sef
Tel: (0 236) 233 37 96 Dahili 125
Faks: (0 236) 236 08 17

Bu evrak güvenli elektronik imza ile imzalanmıştır. http://evraksorgu.meb.gov.tr adresinden 7dca-602b-3ea7-a7df-b2f0 koda ile teyit edilebilir.