



**T.C.**  
**ONDOKUZ MAYIS ÜNİVERSİTESİ**  
**LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ**  
**BİLGİSAYAR VE ÖĞRETİM TEKNOLOJİLERİ EĞİTİMİ ANA BİLİM**  
**DALI**

**MATEMATİK DESTEKLİ TEMEL PROGRAMLAMA**  
**EĞİTİMİNİN BİLGİ İŞLEMSEL DÜŞÜNME BECERİSİ**  
**ÜZERİNDEKİ ETKİSİ**

Yüksek Lisans Tezi

**Muradiye BOZAL**

Danışman

**Doç. Dr. Polat ŞENDURUR**

SAMSUN  
2022

**T.C.**  
**ONDOKUZ MAYIS ÜNİVERSİTESİ**  
**LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ**  
**BİLGİSAYAR VE ÖĞRETİM TEKNOLOJİLERİ EĞİTİMİ ANA BİLİM**  
**DALI**



**MATEMATİK DESTEKLİ TEMEL PROGRAMLAMA**  
**EĞİTİMİNİN BİLGİ İŞLEMSEL DÜŞÜNME BECERİSİ**  
**ÜZERİNDEKİ ETKİSİ**

Yüksek Lisans Tezi

**Muradiye BOZAL**

Danışman

**Doç. Dr. Polat ŞENDURUR**

SAMSUN  
2022

## TEZ KABUL VE ONAYI

**Muradiye BOZAL** tarafından, **Doç. Dr. Polat ŞENDURUR** danışmanlığında hazırlanan “**MATEMATİK DESTEKLİ TEMEL PROGRAMLAMA EĞİTİMİNİN BİLGİ İŞLEMSEL DÜŞÜNME BECERİSİ ÜZERİNDEKİ ETKİSİ**” başlıklı bu çalışma, jürimiz tarafından 5.9.2022 tarihinde yapılan sınav sonucunda oy birliği ile başarılı bulunarak Yüksek Lisans Tezi olarak kabul edilmiştir.

	<b>Unvanı Adı Soyadı</b> <b>Üniversitesi</b> <b>Ana Bilim/Ana Sanat Dalı</b>	<b>İmza</b>	<b>Sonuç</b>
<b>Başkan</b>	Dr. Öğr. Üyesi Nilüfer ATMAN USLU Manisa Celal Bayar Üniversitesi Bilgisayar ve Öğretim Teknolojileri Eğitimi Ana Bilim Dalı		<input checked="" type="checkbox"/> Kabul <input type="checkbox"/> Ret
<b>Üye</b>	Doç. Dr. Polat ŞENDURUR Ondokuz Mayıs Üniversitesi Bilgisayar ve Öğretim Teknolojileri Eğitimi Ana Bilim Dalı		<input checked="" type="checkbox"/> Kabul <input type="checkbox"/> Ret
<b>Üye</b>	Doç. Dr. Rezan YILMAZ Ondokuz Mayıs Üniversitesi Bilgisayar ve Öğretim Teknolojileri Eğitimi Ana Bilim Dalı		<input checked="" type="checkbox"/> Kabul <input type="checkbox"/> Ret

Bu tez, Enstitü Yönetim Kurulunca belirlenen ve yukarıda adları yazılı jüri üyeleri tarafından uygun görülmüştür.

ONAY

... / ... / ...

Prof. Dr. Ali BOLAT  
Enstitü Müdürü

## BİLİMSEL ETİĞE UYGUNLUK BEYANI

Hazırladığım Yüksek Lisans tezinin bütün aşamalarında bilimsel etiğe ve akademik kurallara riayet ettiğimi, çalışmada doğrudan veya dolaylı olarak kullandığım her alıntıya kaynak gösterdiğimi ve yararlandığım eserlerin Kaynaklar'da gösterilenlerden oluştuğunu, her unsurun enstitü yazım kılavuzuna uygun yazıldığını ve TÜBİTAK Araştırma ve Yayın Etiği Kurulu Yönetmeliği'nin 3. bölüm 9. maddesinde belirtilen durumlara aykırı davranılmadığını taahhüt ve beyan ederim.

Etik Kurul Gerekli mi ?

Evet  (Gerekli ise ekler kısmına ekleyiniz)

Hayır

İmza

21 /06/ 2022

Muradiye BOZAL

## TEZ ÇALIŞMASI ÖZGÜNLÜK RAPORU BEYANI

**Tez Başlığı :** MATEMATİK DESTEKLİ TEMEL PROGRAMLAMA EĞİTİMİNİN BİLGİ İŞLEMSEL DÜŞÜNME BECERİSİ ÜZERİNDEKİ ETKİSİ

Yukarıda başlığı belirtilen tez çalışması için şahsım tarafından 21.06.2022 tarihinde intihal tespit programından alınmış olan özgünlük raporu sonucunda;

Benzerlik oranı : % 16

Tek kaynak oranı : % 2 çıkmıştır.

İmza

21 / 06 / 2022

Doç. Dr. Polat ŞENDURUR

## ÖZET

### MATEMATİK DESTEKLİ TEMEL PROGRAMLAMA EĞİTİMİNİN BİLGİ İŞLEMSEL DÜŞÜNME ÜZERİNDEKİ ETKİSİ

Muradiye BOZAL

Ondokuz Mayıs Üniversitesi

Lisansüstü Eğitim Enstitüsü

Bilgisayar ve Öğretim Teknolojileri Eğitimi Ana Bilim Dalı

Yüksek Lisans, Temmuz/2022

Danışman: Doç. Dr. Polat ŞENDURUR

Bu araştırmanın amacı ortaokul 6.sınıf düzeyinde matematik destekli temel programlama eğitiminin bilgi işlemsel düşünme becerisi üzerindeki etkisinin araştırılmasıdır. Araştırmada yarı deneysel tasarımlardan statik gruplar ön-test-son test tasarımı kullanılmıştır. Çalışmaya bir devlet okulunda 6. Sınıfta öğrenim görmekte olan 176 öğrenci katılmıştır. Bu öğrencilerden 89'u deney ve 87'si kontrol grubunda yer almıştır. Deney grubundaki öğrenciler matematik destekli etkinliklerle temel programlama eğitimi alırken, kontrol grubundaki öğrenciler geleneksel içeriklerle programlama eğitimi almıştır. İki grupta da eşdeğer programlama etkinlikleri gerçekleştirilmiştir. Çalışma öncesinde ve sonrasında olmak üzere iki ölçek ile veri toplanmıştır. Bu ölçeklerden ilki Bilgi İşlemsel Düşünme Testidir ve yedi alt boyuttan oluşmaktadır. Diğer ölçek ise Bilgi İşlemsel Düşünme Becerisine Yönelik Öz Yeterlik Algısı Ölçeğidir. Bu ölçek de beş boyut içermektedir. Çalışma tamamlandıktan sonra ön test puanları kontrol edilerek grupların son test puanları üzerindeki etkisi ANCOVA analizi incelenmiştir. Elde edilen bulgulara göre bilgi işlemsel düşünme test performansı açısından iki grup arasında herhangi bir boyutta farklılığa rastlanmamıştır. Benzer şekilde, bilgi işlemsel düşünme becerileri öz yeterlilik algısında gruplar arası fark olduğuna işaret eden herhangi bir bulguya ulaşılamamıştır. Bu sonuçlara göre çalışma sonunda Bilgi İşlemsel Düşünme Becerisinin ilişkili olduğu düşünülen matematik ile programlamanın temel programlama eğitimde birlikte kullanılmasının olumlu ve olumsuz sonuçlarına yönelik değerlendirmelere ulaşılmıştır.

**Anahtar Sözcükler:** Ortaokul öğrencileri, BİD becerisi, Scratch ile matematik dersi

## ABSTRACT

### EFFECT OF MATH SUPPORTED INTRODUCTORY PROGRAMING EDUCATION ON COMPUTATIONAL THINKING

Muradiye BOZAL

Ondokuz Mayıs University

Institute of Graduate Studies

Department of Computer Education and Instructional Technology

Master, July/2022

Supervisor: Assoc. Prof. Dr. Polat ŞENDURUR

The aim of this research is to investigate the effect of mathematics supported basic programming education at the 6th grade level on computational thinking skills. Static groups pre-test-post-test design, which is one of the quasi-experimental designs, was used in the research. 176 6th grade students studying at a public school participated in the study. 89 of these students were in the experimental group and 87 of them were in the control group. While the students in the experimental group received basic programming education with math-supported activities, the students in the control group received the same education with traditional content. Both groups were exposed to equivalent programming content. Data were collected with two scales, before and after the study. The first of these scales is the Computational Thinking Test and consists of seven sub-dimensions. The other scale is the Self-Efficacy Perception Scale for Computational Thinking Skills. This scale also includes five dimensions. After the study was completed, the difference between the groups was examined by ANCOVA analysis, where the pretest results were the control variable and the posttest results were the dependent variable. According to the findings, no difference was found between the two groups in terms of computational thinking skills. The same situation arose in the self-efficacy perception of computational thinking skills, and no finding indicating a difference between the groups could be reached. According to these results, at the end of the study, evaluations regarding the positive and negative results of using mathematics and programming together in basic programming education, which are thought to be related to Computational Thinking Skill, were reached.

**Keywords:** Middle school students, Computational thinking skills, Math lesson with Scratch

## ÖN SÖZ VE TEŞEKKÜR

Bu araştırma ile matematik destekli temel programlama eğitiminin öğrencilerin bilgi işlemsel düşünme becerileri üzerindeki etkisi araştırılmıştır. Araştırmanın, Bilişim Teknolojileri ve Yazılım dersinin disiplinler arası çalışmalarla desteklenmesine örnek teşkil etmesi ve Bilgi İşlemsel Düşünme konusunda yapılan deneysel bir çalışma olması bakımından alanyazına katkı sağlayacağı düşünülmektedir. Araştırma ve tez yazım süreci, bir bilişim teknolojileri öğretmeni olarak mesleki hayatıma çok büyük katkılar sağlamıştır.

Yüksek Lisans eğitimim ve tez yazma sürecimde bilgisi ve tecrübesi ile bana yol gösteren, sabrı ve hoşgörüsüyle daima güven veren değerli hocam Doç. Dr. Polat Şendurur'a teşekkürü bir borç bilirim.

Bu süreçte beni her zaman destekleyen, sorumlulukları paylaşan, sonsuz bir sabır gösteren sevgili eşim Tarık BOZAL'a ve canım çocuklarım Arda ile Aras'a sonsuz teşekkürlerimi sunarım.

Son teşekkürümü de, her zaman başarılı olmayı hedefleyen, çocuklarımı en doğru şekilde yetiştirmek için çaba sarf eden, bir işi yaparken severek ve en güzel şekilde yapmayı öğreten, hayatımın her alanında örnek aldığım canım annem Sevdije Akel'e sunmaktan onur duyarım.

Muradiye BOZAL

## İÇİNDEKİLER

TEZ KABUL VE ONAYI .....	i
BİLİMSEL ETİĞE UYGUNLUK BEYANI .....	ii
TEZ ÇALIŞMASI ÖZGÜNLÜK RAPORU BEYANI .....	ii
ÖZET .....	iii
ABSTRACT .....	iv
ÖNSÖZ VE TEŞEKKÜR .....	v
İÇİNDEKİLER.....	vi
SİMGELER VE KISALTMALAR .....	vii
ŞEKİLLER DİZİNİ .....	viii
TABLolar DİZİNİ .....	ix
<b>1. GİRİŞ.....</b>	<b>1</b>
1.1. Problem Durumu .....	2
1.2. Araştırmanın Amacı .....	5
1.3. Araştırmanın Önemi .....	6
1.4. Araştırmanın Sınırlılıkları.....	6
1.5. Araştırma Soruları .....	7
1.6. Tanımlar .....	7
<b>2. KURAMSAL TEMELLER.....</b>	<b>8</b>
2.1. BİD Tarihçesi .....	8
2.2. Bilgi İşlemsel Düşünmenin İsimlendirilmesi.....	8
2.3. BİD'in Tanımlanması.....	9
2.4. BİD'in Alt Bileşenleri.....	10
2.5. Alt Bileşenlerin Tanımları .....	12
2.6. BİD Becerisinin Değerlendirilmesi .....	15
2.7. Bilgi İşlemsel Düşünmenin Öğretilmesinde Kullanılan Araçlar.....	16
2.8. Scratch .....	18
2.9. Scratch ve BİD İlişkisi.....	19
2.10. Matematik, Scratch ve BİD İlişkisi .....	22
<b>3. YÖNTEM .....</b>	<b>25</b>
3.1. Araştırmanın Modeli.....	25
3.2. Çalışma Grubu.....	26
3.3. Veri Toplama Araçları .....	26
3.3.1. BİD Testi .....	27
3.3.2. Bilgi İşlemsel Düşünme Becerisine Yönelik Öz Yeterlik Algısı Ölçeği ..	28
3.4. Deneysel İşlem .....	29



3.4.1. Deney Grubu Etkinlikleri .....	29
3.4.2. Kontrol Grubu Etkinlikleri .....	38
3.5. Verilerin Analizi.....	47
<b>4. BULGULAR.....</b>	<b>48</b>
4.1. BİD Testine Ait Bulgular.....	49
4.1.1. Temel Sıralama Boyutu.....	49
4.1.2. Belirli Sayıda Tekrar Eden Döngüler Boyutu .....	50
4.1.3. Şart Gerçekleşene Kadar Çalışan Döngüler Boyutu .....	52
4.1.4. Basit Şartlı İfadeler Boyutu .....	53
4.1.5. Karmaşık Şartlı İfadeler Boyutu .....	55
4.1.6. Şart Gerçek Oldukça Çalışan Döngüler Boyutu.....	57
4.1.7. Basit Fonksiyonlar Boyutu .....	58
4.2. BİDBÖA Ölçeğine Ait Bulgular .....	60
4.2.1. Algoritma Tasarlama Yeterliği .....	60
4.2.2. Problem Çözme Yeterliği .....	61
4.2.3. Veri İşleme Yeterliği.....	63
4.2.4. Temel Programlama Yeterliği .....	64
4.2.5. Özgüven Yeterliği .....	66
<b>5. TARTIŞMA, SONUÇ VE ÖNERİLER.....</b>	<b>68</b>
5.1. Bilgi İşlemsel Düşünme Becerisi .....	68
5.2. Bilgi İşlemsel Düşünme Becerisi Özyeterlik Algısı.....	71
5.3. Öneriler .....	73
<b>KAYNAKÇA.....</b>	<b>75</b>
<b>EKLER.....</b>	<b>80</b>
Ek 1. BİLGİ İŞLEMSEL DÜŞÜNME TESTİ.....	80
Ek 2. Ortaokul Öğrencileri İçin Bilgi İşlemsel Düşünme Becerisine Yönelik Öz Yeterlik Algısı Ölçeği.....	93
Ek 3. Araştırma İzni .....	95
Ek 4. Etik Kurul Kararı .....	96
<b>ÖZ GEÇMİŞ .....</b>	<b>97</b>

## **SİMGELER VE KISALTMALAR**

- BİD : Bilgi İşlemsel Düşünme  
BİDBÖA : Bilgi İşlemsel Düşünme Becerisi Özyeterlik Algısı  
B<sup>3</sup> : Bilgisayarsız Bilgisayar Bilimi

## ŞEKİLLER DİZİNİ

Şekil 2.1. BİD Alt Bileşenleri (Kalelioğlu ve Gülbahar, 2015).....	12
Şekil 2.2. BİD öğretiminde kullanılan yaklaşımlar.....	17
Şekil 2.3. BİD öğretiminde kullanılabilecek örnek programlama araçları (Hsu vd., 2018) ...	18
Şekil 2.4. Scratch Program Geliştirme Ortamı. (Kaynak: <a href="https://scratch.mit.edu/">https://scratch.mit.edu/</a> ) .....	19
Şekil 3.1. Doğal sayılarda işlem önceliği etkinliği, ekran çıktısı .....	30
Şekil 3.2. Doğal sayılarda işlem önceliği etkinliği, kuklaya yazılan kodlar .....	30
Şekil 3.3. Doğal sayılarda işlem önceliği etkinliği, işlem isimlerine ait karakterlere yazılan kodlara bir örnek .....	31
Şekil 3.4. Üslü ifadeleri hesaplama etkinliği, oyunun ana ekran görüntüsü.....	32
Şekil 3.5. Üslü ifadeleri hesaplama etkinliği, karaktere yazılan kodlar .....	33
Şekil 3.6. Bölünebilme kurallarını kullanma etkinliği, programın ekran görüntüsü.....	34
Şekil 3.7. Bölünebilme kurallarını kullanma etkinliği, karaktere yazılan kodların görüntüsü .....	35
Şekil 3.8. Asal sayıları bulma etkinliği, programın ana ekran görüntüsü.....	36
Şekil 3.9. Asal sayıları bulma etkinliği, karaktere yazılan kodların görüntüsü .....	37
Şekil 3.10. Başkent bulma oyunu etkinliği, programın ana ekran görüntüsü .....	38
Şekil 3.11. Başkent bulma oyunu etkinliği, karaktere yazılan kodların görüntüsü.....	39
Şekil 3.12. Elma toplama oyunu etkinliği, programın ana ekran görüntüsü.....	40
Şekil 3.13. Elma toplama oyunu etkinliği, elma karakterine yazılan kodların görüntüsü .....	41
Şekil 3.14. Elma toplama oyunu etkinliği, kase karakterine yazılan kodların görüntüsü.....	42
Şekil 3.15. İngilizce kelime oyunu etkinliği, ana ekran görüntüsü .....	43
Şekil 3.16. İngilizce kelime oyunu etkinliği, ortada görünen karaktere yazılan kodların görüntüsü .....	44
Şekil 3.17. İngilizce kelime oyunu etkinliği, buton karakterine yazılan kodların görüntüsüne bir örnek.....	44
Şekil 3.18. Burcunu bulma oyunu etkinliği, ana ekran görüntüsü .....	45
Şekil 3.19. Burcunu bulma oyunu etkinliği, karaktere yazılan kodların görüntüsü .....	46
Şekil 3.20. Burcunu bulma oyunu etkinliği, karaktere yazılan kodların görüntüsü .....	46

## TABLolar DİZİNİ

Tablo 3.1. Uygulama süreci.....	25
Tablo 3.2. BİD testi boyutları ve soru sayıları.....	27
Tablo 3.3. BİDBÖA faktörler ve madde sayıları.....	28
Tablo 4.1. Temel sıralama boyutunda ANCOVA sonuçları.....	49
Tablo 4.2. Temel sıralama ifadesi değişkenine yönelik deney ve kontrol grubu ortalamaları.....	50
Tablo 4.3. Belirli sayıda tekrar eden döngüler boyutunda ANCOVA sonuçları.....	50
Tablo 4.4. Belirli sayıda tekrar eden döngüler değişkenine yönelik deney ve kontrol grubu ortalamaları.....	51
Tablo 4.5. Şart gerçekleşene kadar çalışan döngüler boyutunda ANCOVA sonuçları.....	52
Tablo 4.6. Şart gerçekleşene kadar çalışan döngüler değişkenine yönelik deney ve kontrol grubu ortalamaları.....	53
Tablo 4.7. Basit şartlı ifadeler boyutunda ANCOVA sonuçları.....	53
Tablo 4.8. Basit şartlı ifadeler değişkenine yönelik deney ve kontrol grubu ortalamaları.....	54
Tablo 4.9. Karmaşık Şartlı İfadeler boyutunda ANCOVA sonuçları.....	55
Tablo 4.10. Karmaşık şartlı ifadeler değişkenine yönelik deney ve kontrol grubu ortalamaları.....	56
Tablo 4.11. Şart gerçek oldukça çalışan döngüler boyutunda ANCOVA sonuçları.....	57
Tablo 4.12. Şart gerçek oldukça çalışan döngüler değişkenine yönelik deney ve kontrol grubu ortalamaları.....	58
Tablo 4.13. Basit Fonksiyonlar boyutunda ANCOVA sonuçları.....	58
Tablo 4.14. Basit Fonksiyonlar değişkenine yönelik deney ve kontrol grubu ortalamaları.....	59
Tablo 4.15. Algoritma Tasarlama Yeterliği ifadesi boyutunda ANCOVA sonuçları.....	60
Tablo 4.16. Algoritma Tasarlama Yeterliği ifadesi değişkenine yönelik deney ve kontrol grubu ortalamaları.....	61
Tablo 4.17. Problem Çözme Yeterliği ifadesi boyutunda ANCOVA sonuçları.....	61
Tablo 4.18. Problem Çözme Yeterliği ifadesi değişkenine yönelik deney ve kontrol grubu ortalamaları.....	62
Tablo 4.19. Veri İşleme Yeterliği ifadesi boyutunda ANCOVA sonuçları.....	63
Tablo 4.20. Veri İşleme Yeterliği ifadesi değişkenine yönelik deney ve kontrol grubu ortalamaları.....	64
Tablo 4.21. Temel Programlama Yeterliği ifadesi boyutunda ANCOVA sonuçları.....	64
Tablo 4.22. Temel Programlama Yeterliği ifadesi değişkenine yönelik deney ve kontrol grubu ortalamaları.....	65
Tablo 4.23. Özgüven Yeterliği ifadesi boyutunda ANCOVA sonuçları.....	66
Tablo 4.24. Özgüven Yeterliği ifadesi değişkenine yönelik deney ve kontrol grubu ortalamaları.....	66

# 1. GİRİŞ

İçinde bulunduğumuz yüzyılda teknoloji büyük bir hızla gelişmeye devam etmekte ve hayatın her alanında daha çok yer kaplamaktadır. İnternetin tüm dünyayı bir ağ gibi sarmasıyla her an her yerde internete bağlı olan bilgi ve iletişim araçları kullanılmaktadır. Gelişmiş ülkeler ekonomik büyümenin teknolojik gelişmeyle paralel olmasından dolayı teknoloji alanındaki çalışmalara daha fazla yatırım yapmaktadır. Ekonomiye yön veren teknolojileri geliştirmek ve kullanmak için bireyleri teknolojik becerilerle donatıp geleceğin dünyasına hazırlıklı hale getirmeyi amaçlamaktadır. Bu yüzden bilgisayar bilimleri ile ilgili konuları ana sınıfından itibaren öğretim programlarına entegre etmektedirler. Ülkelerin öğretim programları incelendiğinde, İngiltere (2014), Fransa (2016), Finlandiya (2016), Polonya (2017), Danimarka (2017) gibi ülkelerin son yıllarda öğretim programlarında BİD (Bilgi işlemsel düşünme) becerilerini içeren güncellemeler yaptıkları ve algoritma, kodlama, programlama dilleri, enformatik gibi konuları öğretim programlarına dâhil ettikleri görülmektedir. Türkiye’de de son yıllarda BİD becerisinin geliştirilmesi için bazı derslerin öğretim programları yeniden düzenlenmekte ve içerikler üretilmektedir (Gülbahar ve Kalelioğlu, 2018). Talim ve Terbiye Kurulu Başkanlığınca ortaokullarda zorunlu olarak okutulan “Bilişim Teknolojileri” ders müfredatına 2018 yılında yazılım ve kodlama etkinlikleri eklenmiş ve dersin adı “Bilişim Teknolojileri ve Yazılım” şeklinde güncellenmiştir. Türkiye’deki Bilişim Teknolojileri ve Yazılım dersi müfredatı ile bir yandan öğrencilerin etik ilkelere dikkat ederek bilişim teknolojileri araçlarını ve interneti doğru şekilde kullanmayı öğrenmeleri amaçlanırken; diğer yandan da programlama becerisi, algoritmik düşünme ve problem çözme becerisi kazanmaları hedeflenmektedir. Geleceğin üreten, geliştiren, inovatif düşünen bireylerini yetiştirirken üst düzey düşünme becerileri edindirmek en önemli kazanımlardan biridir. Bu üst düzey düşünme becerileri kısaca BİD kapsamında ele alınmaktadır.

Bu bölümde araştırmaya ait problem durumu, araştırmanın amacı, önemi, sınırlıkları, araştırma soruları ve tanımlara yer verilmektedir.

## 1.1. Problem Durumu

Bilgisayar bilimi ve teknolojinin hızla gelişmesi sonucunda öğrencilerin teknolojik konularda ortaya çıkan problemleri çözmeye yeterliklerinin geliştirilmesi eğitim sistemlerinde önemli bir konu olarak el alınmaktadır. Eğitim, sağlık, ticaret, sanayi, ulaşım, eğlence gibi daha pek çok alanda ortaya çıkan sorunlar, geliştirilen donanım ve yazılımlar sayesinde çözülmektedir. Wing (2006)'e göre günlük hayatımızı yönetmek, diğer insanlarla iletişim kurmak ve etkileşime geçmek için geliştirilen çözümler yalnızca fiziksel olan yazılım ve donanım ürünleri ile sınırlı kalmamalıdır. Asıl üzerinde durulması gereken nokta, öğrencilere bilgi işlemsel kavramlarının kazandırılmasını sağlamaktır. Bilgi işleme, soyutlama, parçalara ayırma, algoritmik düşünme ve problem çözmeye gibi becerileri içermektedir. Bir problemin çözümünde bilişim teknolojilerinden yararlanmayı Wing (2006) BİD becerisi olarak ifade etmiştir.

Eğitim teknolojileri konusunda önemli bir kuruluş olan Uluslararası Eğitim Teknolojileri Topluluğu (ISTE - The International Society for Technology in Education), eğitim kurumlarının teknolojiyi planlı bir şekilde kullanmalarına katkı sağlamayı ve bir yol gösterici olmayı amaçlamaktadır. Bu sebeple öğrenciler, öğretmenler, yöneticiler, koçlar ve bilgisayar eğitimcileri için bazı standartlar geliştirmiştir. Uluslararası Eğitim Teknolojileri Topluluğu (ISTE)'nin 2016 yılında öğrenciler için geliştirdiği standartlar aşağıdaki şekildedir:

1. Güçlendirilmiş Öğrenen
2. Dijital Vatandaş
3. Bilgi İnşa Edici
4. Yenilikçi Tasarımcı
5. Bilgi-İşlemsel Düşünür
6. Yaratıcı İletişimci
7. Küresel İşbirlikçi

Bu standartlar; BİD'in, içinde bulunduğumuz yüzyılda yaşayan bireylerin edinmesi gereken önemli becerilerden biri olduğunu göstermektedir. BİD'in bu kadar önemli hale gelmesi, "BİD becerisi nedir?", "Nasıl kazandırılır?", "Nasıl ölçülür?" gibi soruları beraberinde getirmiştir. Wing'e göre BİD problem çözerken matematiksel düşünmenin yanı sıra bir mühendis gibi düşünmeyi de gerektirir (Wing, 2008). Papert (1980), öğrencilerinin bilgisayar programı yazarken düşünme

kabiliyetlerinin önemli ölçüde farklılaştığını ve bilişsel yeteneklerini geliştirdiğini gözlemlemiştir. ISTE (2015), BİD becerisinin yaratıcı düşünme, algoritmik düşünme, problem çözme ve işbirlikçi öğrenme gibi becerileri barındırdığını ifade etmiştir. Bu ifadeler göstermektedir ki; BİD, matematik, mühendislik, bilgisayar programlama gibi derslerdeki kazanımların harmanlanmasıyla ortaya çıkan bir beceri türüdür. Bu doğrultuda düşünen pek çok ülke öğrencilerin, bir mühendis gibi düşünebilmesi ve bu becerileri kazanabilmesi için bu konuları içeren öğretim programları uygulamaktadır. Ülkemizde de ana sınıftan itibaren imkânlar ölçüsünde bilgisayarlı ve bilgisayarsız kodlama ile robotik çalışmaları yapılmaktadır. Ortaokul 5. ve 6.sınıflarda Bilişim Teknolojileri ve Yazılım dersinde algoritma ve blok tabanlı temel programlama eğitimi verilmektedir. Ortaöğretimde ise metin tabanlı programlama, mobil programlama, robot programlama ve web programlama konuları öğretilmektedir. Ortaokulda blok tabanlı programlama aracı olarak çoğunlukla Scratch yazılımı kullanılmaktadır.

Scratch öğrenciler tarafından öğrenmesi kolay, eğlenceli, güncel ve yaratıcılığı geliştirmeye yarayan bir görsel programlama aracıdır. Öğrenciler Scratch programını bilgisayarlarına kurabildikleri gibi çevrimiçi olarak da kullanabilmektedir. Online platformun aynı zamanda paylaşım yapmaya olanak sağlaması ve sosyal bir ortam oluşturması öğrenciler için bir avantajdır. Scratch programında eğitsel içerikler, oyunlar ve animasyonlar geliştirilebilmektedir. Öğrenciler verilen bir problemin çözümüne ulaşmaya çalışırken değişkenler, döngüler, şart yapıları ve matematiksel işlemleri kullanırlar. Tıpkı bir matematik problemini çözerken yaptıkları gibi parçalara ayırma, örüntü tanıma, soyutlama, birleştirme gibi adımları uygularlar. Hem matematik dersinde hem de programlamada kullanılan bu problem çözme yöntemlerinin BİD’i geliştirdiği bilinmektedir (MEB, 2020).

Oluk, Korkmaz ve Oluk (2018) tarafından yürütülen bir araştırmada ortaokul öğrencileri ile Scratch programı kullanılarak algoritma yazma çalışmaları yapılmış ve çalışmanın sonucunda öğrencilerin BİD becerisinde anlamlı bir gelişme ortaya çıkmıştır. Sırakaya (2019) tarafından yüksekokul programlama bölümü birinci sınıfta okuyan öğrencilerle yürütülen bir başka araştırmada programlama öğretiminin BİD becerisini geliştirme üzerinde olumlu bir etkisi olduğu görülmüştür. Şimşek (2018) tarafından ortaokul öğrencileri üzerinde yapılan bir araştırmada hem Scratch hem de robotik araçlar ile programlama anlatılmış ve ikisinin de öğrencilerin BİD becerilerini geliştirdiği sonucuna ulaşılmıştır. Bu araştırmalar göstermektedir ki

Scratch ile blok tabanlı programlama öğretimi öğrencilerin BİD becerilerini geliştirmede katkı sağlamaktadır.

Bununla birlikte, BİD becerisinin yalnızca programlama yaparak öğrenileceğini düşünmek sınırlı bir bakış açısını yansıtmaktadır. BİD sadece bir programın kodlarını bilgisayar ortamında yazılmasını içermemekte; programı yazmadan önce yapılan; problemi anlama, ayrıştırma, soyutlama, algoritmik düşünme, akış diyagramı oluşturma gibi işlemler sırasında kazanılmaktadır (MEB, 2020). Ayrıca programlama ile kodlama kavramları da toplumda birbirinin yerine kullanılmaktadır. Bu durum bir kavram yanılgısına işaret etmektedir. Bir bilgisayar programcısı sıra ile algoritma tasarımı, gerekli işlemleri yapacak programı yazma, hata ayıklama ve test etme işlemlerini yapmaktadır. Kodlama ortamları ise küçük boyutta program yazma süreçlerini ve algoritmik düşünmenin kazanılması için bir ortam sunar. Programlama sürecinde kodlamadan daha üst seviyede zihinsel işlemler yapılmaktadır. Dolayısıyla kodlama ortamında öğrencinin soyutlama, model oluşturma, ayrıştırma, veri düzenleme, hata ayıklama becerileri ile karşılaştırılmaması durumunda sadece belirli hedefler yerine getirilirken BİD becerisi kazanma süreci tamamlanmamış olur.

Temel olarak problem çözme adımlarını uygulayan disiplinler BİD’i geliştirmektedir. Alanyazında Biyoloji, Fizik, Matematik ve İngilizce gibi derslerin BİD becerisini geliştirdiğine dair çalışmalar yer almaktadır (Lockwood ve Mooney, 2017). Matematik dersi öğretim programında özellikle problem çözme becerilerini kazandırmaya yönelik hedefler yer almaktadır (MEB, 2018). Bilgisayar bilimi, gerçek dünyayla etkileşime giren sistemler oluşturmak için matematiksel düşünceden ve mühendislik düşüncesinden yararlanmaktadır. Temeldeki bilgi işlem aygıtının kısıtlamaları, bilgisayar bilimcileri yalnızca matematiksel olarak değil, sayısal olarak da düşünmeye zorlamaktadır (Wing, 2006). Matematik ile BİD arasındaki ilişki son yıllarda araştırılmaya başlanmıştır. Bazı araştırmalar Matematik dersinin BİD becerisini geliştirmeye katkı sağladığını öne sürerken bazı araştırmalar da BİD’in matematik dersindeki kazanımların öğrenilmesine yardımcı olduğunu ifade etmektedir. BİD ve matematik arasında karşılıklı fayda sağlayan bir ilişki vardır. Sung ve Black (2020) tarafından yürütülen yarı deneysel bir çalışmada öğrenciler, bir bilgisayar programcısı gibi düşünerek çalıştıklarında, görev ayrıştırma, sıralı düşünme, prosedürel düşünme ve kodlama becerilerinin geliştiği gözlemlenmiştir. Bu durum BİD becerisinin matematik içeriğinin öğrenilmesini derinleştirdiğini ve



disiplinler arası bir şekilde mevcut matematik müfredatına BİD bakış açıları yerleştirildiğinde etkinliklerin daha etkili olduğunu doğrulamaktadır. BİD'in alt bileşenlerinin matematik öğretimine dâhil edilmesinin, alanın daha gerçekçi algılanmasını sağlarken matematik içeriğinin öğrenilmesini de destekleyeceği düşünülmektedir. Rodríguez-Martínez, González-Calero ve Sáez-López (2020), Scratch programının en büyük ortak bölen (EBOB) ve en küçük ortak kat (EKOK) gibi kavramların öğrenilmesi üzerindeki etkisini incelemiştir. Altıncı sınıf öğrencileri üzerinde gerçekleştirdikleri çalışmada, Scratch ile programlama etkinliklerini tamamladıktan sonra öğrencilerin Ebob ve Ekok ile ilgili problemleri çözümedeki yeterliliklerinde göreceli bir gelişme olduğunu ortaya koymuştur. Bu noktalardan hareketle matematik ile ilgili kazanımların bilgisayar bilimleri kavramları ile entegrasyonunun sağlanması ve BİD becerilerine etkisi olabileceğine yönelik çıkarımlar yapılabilir.

Araştırmalar göstermektedir ki programlama öğretimi BİD becerisinin geliştirilmesine ve Matematik öğretimine olumlu katkılar sağlamaktadır. Bu araştırma ile ortaokul temel programlama öğretimi sırasında Matematik konuları içeren etkinlikler uygulandığında öğrencilerin BİD becerilerinde anlamlı bir fark oluşup oluşmadığı araştırılmaktadır.

## **1.2. Araştırmanın Amacı**

Bu araştırma ile ortaokul Bilişim Teknolojileri ve Yazılım dersinde matematik destekli temel programlama eğitiminin 6.sınıf öğrencilerinin BİD test performansı ve BİD özyeterliği üzerindeki etkisinin araştırılması amaçlanmaktadır. BİD becerisi kazandırma konusunda etkili olan bilişim teknolojileri ve matematik derslerinin kazanımlarının aynı etkinlikte sunulmasıyla, Scratch ile temel programlama öğretilirken matematik destekli etkinliklerin BİD becerisinin gelişimindeki rolüne odaklanılmıştır.

Bu araştırma ile BİD becerisini ölçmede kullanılan araçlardan biri olan BİD testinde yer alan alt boyutlarda deney gurubundaki öğrencilerin kontrol grubu öğrencilerine kıyasla gelişme gösterip göstermediklerine bakılarak uygulanan etkinliklerin etkili olup olmadığı konusunda çıkarımlarda bulunulması hedeflenmiştir. Öğrencilerin kendilerinin programlama ve BİD becerileri konusundaki özyeterlik algılarının ölçülmesi ile çıkan sonuçlara göre kullanılan etkinliklerin ve yöntemlerin etkililiğinin incelenmesi amaçlanmıştır.

Araştırma sonucunda elde edilen bilgilere göre temel programlama öğretiminde kullanılan içerik ve etkinliklerin disiplinler arası konularla desteklenmesinin olumlu bir katkısının olup olmadığını ortaya çıkarılmasına çalışılacaktır. Matematik destekli programlama eğitiminin öğrencilerin BİD becerilerini geliştirmedeki rolüne odaklanılarak disiplinler arası çalışmaların ne yöne evrileceği ile ilgili çıkarımlara ulaşılarak alanyazına katkı sağlanması hedeflenmiştir. Aynı zamanda, bu çalışma ile programlama, BİD ve matematik konularında yapılacak araştırmalara içerik, yöntem, süre ve etkinlikler bakımından uygulayıcılar ve araştırmacılar için örnek uygulamaların teşkil edilmesi amaçlanmıştır.

### **1.3. Araştırmanın Önemi**

Bu araştırma ile temel programlama eğitiminde matematik ve programlama etkinliklerinin birlikte kullanımının öğrencilerin BİD becerileri üzerindeki etkisinin anlaşılması için bir adım atılmıştır. Araştırma bulgularının, öğrencilerin BİD becerilerinin gelişiminde yararlanılabilecek öğrenme ve öğretme yollarının belirlenmesinde araştırmacı ve uygulayıcılara katkı sağlayacağı düşünülmektedir. Ayrıca araştırma, bilgisayar bilimi eğitiminin yanında matematik eğitimi için de bazı öneriler sunma potansiyeli barındırmaktadır. BİD’i daha iyi anlama ve ilişkili olduğu alanların tespitine yönelik yeni fikirler sunma potansiyeli vardır. BİD halen gelişmekte olan bir konu olduğundan bu alanda deneysel araştırmalara ihtiyaç duyulmaktadır. Ulusal alanyazındaki araştırmaların uluslararası alanyazına kıyasla daha az olmasından dolayı bu deneysel araştırmanın ülkemizdeki araştırmacılara fayda sağlayacağı düşünülebilir. Son yıllarda programlama ve matematik konularının birlikte kullanılmasına dair çalışmalara uluslararası alanyazında fazlaca rastlanırken ülkemizde bu konuların birlikte araştırılmasına çok az rastlanmaktadır. Bu sebeple bu çalışma matematik ve programlama etkinliklerinin birlikte öğretilmesinin uygulandığı deneysel bir çalışma olmasından dolayı da önemlidir.

### **1.4. Araştırmanın Sınırlılıkları**

Araştırma Samsun’da bir devlet ortaokulunda 2021-2022 eğitim-öğretim yılında 6.sınıfta okuyan öğrenciler ile yürütülmüştür. Öğrencilerin BİD ile ilgili yeterlikleri aşağıda yer alan ölçeklerle sınırlı tutulmuştur:

- BİD Testi (Çetin, Otu, Oktaç, 2020)
- BİD Becerisine Yönelik Öz Yeterlik Algısı Ölçeği (Gülbahar, Kert, Kalelioğlu, 2018)

### 1.5. Arařtırma Soruları

1. Matematik destekli temel programlama eđitiminin 6. sınıf öđrencilerinin BİD testi performansı üzerinde etkisi var mıdır?
2. Matematik destekli temel programlama eđitiminin 6. sınıf öđrencilerinin BİD öz-yeterlik algısı üzerinde etkisi var mıdır?
3. Deney ve kontrol grubundaki öđrencilerin BİD testi ön-test ve son-test puanları arasında anlamlı bir farklılık var mıdır?
4. Deney ve kontrol grubundaki öđrencilerin BİD öz yeterliđi ön-test ve son-test puanları arasında anlamlı bir farklılık var mıdır?

### 1.6. Tanımlar

*BİD*: Bilgisayar biliminde kullanılan temel kavramlardan yararlanarak problem çözüme, sistemler tasarlama ve insan davranıřlarını anlamaktır (Wing, 2006).

*Özyeterlik ve BİD Öz yeterliđi*: Bireyin belli bir performansını göstermek için gerekli etkinlikleri organize edip başarılı olma kapasitesi hakkında kendine iliřkin yargısına öz yeterlik olarak tanımlanmaktadır (Bandura, 1986, s.2). BİD öz yeterliđi ise bireyin kendisinin BİD becerisi açasından öz yeterliđine iliřkin kiřisel algısıdır.

*Matematik destekli temel programlama eđitimi*: Temel programlama eđitimi ortaokul 5. ve 6.sınıflarda okutulan Biliřim Teknolojileri ve Yazılım dersinde gösterilen algoritma yazma ve blok tabanlı programlama konularının öđretilmesidir. Matematik destekli temel programlama ise programlama öđretimi konularının disiplinler arası öđrenme yaklařımından hareketle matematik konuları ile iliřkilendirilmesidir.

## 2. KURAMSAL TEMELLER

Bu bölümde BİD'in tarihçesi, isimlendirilmesi, tanımlanması, alt bileşenleri ve bunların tanımları, BİD becerisinin değerlendirilmesi, BİD'in öğretilmesinde kullanılan araçlar, Scratch, Matematik ve BİD ilişkisinden bahsedilmiştir.

### 2.1. BİD Tarihçesi

BİD becerisi son yıllarda oldukça popüler hale gelmiş olsa da kavramın oluşmaya başlaması oldukça eskiye dayanmaktadır. Bilgisayar çalışma mantığının bir problem çözme yöntemi olarak öğretilmesi ile ilgili çalışmalar ilk olarak 1960'larda Alan Perlis tarafından başlatılmıştır (Özçınar, 2017). BİD kavramı ise ilk olarak 1996 yılında Papert tarafından kullanılmıştır. Ancak çok daha önce Papert (1980) öğrencilerinin bilgisayar programı yazarlarken düşünme kabiliyetlerinin önemli ölçüde farklılaştığını ve bu durumun da bilişsel yeteneklerini geliştirdiğini gözlemlemiştir. Çocukların LOGO programlama dilini öğrenerek BİD becerilerini geliştirebileceklerini ifade etmiştir. Bu düşüncesi öğrencilerin programlama yaparken uyguladıkları süreçler sonunda düşünme becerilerinde meydana gelen gelişmenin fark edilmesiyle ortaya çıkmıştır. BİD'in kavram olarak tanımlanması ise ilk olarak Jeannette M. Wing'in (2006) yılındaki çalışmasında yer almıştır. Bilgisayar biliminin bilgisayar programlamaktan ibaret olmadığını, bir bilgisayar bilimcisi gibi düşünmenin çoklu soyutlama seviyelerinde düşünmeyi gerektirdiğini ifade etmiştir. Wing (2008)'e göre BİD bir tür analitik düşüncedir ve bir problemi çözmenin genel yolları matematiksel, mühendislik ve bilimsel düşünme becerilerine dayanır.

### 2.2. Bilgi İşlemsel Düşünmenin İsimlendirilmesi

Uluslararası alanyazında bilgi işlemsel düşünme kavramı *Computational Thinking* (Wing, 2006; Angeli vd., 2016; Brennan ve Resnick, 2012) olarak kullanılmaktadır. Ulusal alanyazın incelendiğinde isim konusunda fikir ayrılıkları olduğu görülmektedir. Kullanılan bazı isimler şunlardır:

- Bilgisayarca düşünme (Korkmaz vd., 2015)
- Komputasyonel düşünme (Şahiner ve Kert, 2016)
- Bilişimsel düşünme (Ökkeş, 2016)
- Hesaplamalı düşünme (Özçınar, 2017)
- Bilgi işlemsel düşünme (Gülbahar, Kert, Kalelioğlu, 2019)
- Bilgisayımsal düşünme (Çınar vd., 2019)

MEB tarafından çıkarılan ‘Bilgi İşlemsel Düşünme Becerisinin Disiplinlerarası Yaklaşım İle Öğretimi (2020)’ isimli kitapta kavram; *Bilgi İşlemsel Düşünme* olarak kullanılmaktadır. Bu doğrultuda, bu araştırmada bu adlandırmanın kullanılmasının uygun olduğuna karar verilmiştir.

### **2.3. BİD’in Tanımlanması**

Wing (2006), BİD’i, bilgisayar biliminde kullanılan temel kavramlardan yararlanarak problem çözme, sistemler tasarlama ve insan davranışlarını anlama olarak ifade etmiştir. Öğrencilerin okuma, yazma ve aritmetik becerilerine BİD’in eklenmesi gerektiğini söylemiştir. Keşfedilmeyi bekleyen bilimsel problemlerin çözüme kavuşması ve anlaşılması için bu becerinin öğrencilere erken yaşlarda öğretilmesinin önemini vurgulamıştır. Gelişmenin yalnızca kişinin kendi yaratıcılığı ile sınırlı olduğunu ve öğrencilerin bilgisayar biliminde uzmanlaşarak tıp, hukuk, işletme, politika, her türlü bilim dalı, mühendislik ve hatta sanat alanlarında kariyer yapabileceklerini ifade etmiştir.

Aho’ya göre (2012), problemlerin çözümlerinin bilgisayar mantığına uygun adımlar ve algoritmalarla sunulabildiği düşünce süreçlerini içermektedir. Sysło ve Kwiatkowska (2013), bir problemi formüle etmedeki zihinsel aktivite olarak tanımlamışlardır. Özden, Korkmaz, Çakır (2015), bir çeşit problem çözme, sistem tasarımı ve bilgisayar biliminin temel kavramlarına dikkat çekerek insan davranışlarını anlama yöntemi olarak tanımlamışlardır. Curzon (2015), insanlar için bir çeşit problem çözme becerisi olarak ifade etmektedir. Angeli vd. (2016) göre, soyutlamanın unsurlarını kullanan bir düşünce sürecidir. Şahiner ve Kert (2016)’e göre, içerisinde eleştirel düşünme, problem çözme, algoritmik düşünme gibi becerileri bulunduran ve bilgisayarın çalışma tarzının gündelik yaşama uyarlandığı kapsamlı bir beceridir.

Yünkül, Durak, Çankaya ve Mısırlı (2017) BİD’i, probleme yönelik çözümün tasarlanması, uygulanması ve algoritmik düşünme becerisinin geliştirilmesi süreci olarak tanımlamıştır. Çınar vd. (2019), mantıksal becerilerin algoritmik adımlarla sergilenmesini ifade eden bir problem çözme yaklaşımıdır.

Uluslararası Eğitim Teknolojileri Topluluğu (ISTE) ve Bilgisayar Bilimi Öğretmenleri Derneği (CSTA) tarafından yapılan ortak bir bildiriye BİD Becerisi aşağıda yer alan özellikleri içeren bir problem çözme süreci olarak ifade edilmiştir (2011):

- Problemleri bilgisayar yardımıyla çözebilmek için formülleştirme
- Verileri mantık çerçevesinde düzenleme ve çözümlenme
- Modeller ve simülasyonlar aracılığı ile verileri sunma
- Algoritmik düşünme ile çözümleri otomatikleştirme
- Kaynakları etkin bir şekilde kullanarak en uygun tanımlama, çözümlenme ve uygulama
- Bulunan çözümü farklı problemlere transfer etme ve geliştirme

Ulusal ve uluslararası alanyazında BİD konusunda net bir tanım olmadığı ancak tanımların problem çözme, algoritma, soyutlama, eleştirel düşünme gibi kavramlarla ifade edildiği görülmektedir. Uzmanların çoğu BİD'in bilişim araçlarını kullanarak bir çeşit problem çözme süreci olduğu konusunda hemfikir olduğu söylenebilir.

#### **2.4. BİD'in Alt Bileşenleri**

BİD'in isimlendirmesi ve tanımlanmasında olduğu gibi alt bileşenleri belirlemede de bir fikir birliği söz konusu değildir. Ulusal ve uluslararası alanyazın taramalarında karşılaşılan alt bileşenler şu şekildedir:

Wing (2006), BİD becerisinin beş alt bileşeni olduğunu ifade etmiştir:

- Problem çözme
- Soyutlama
- Ayırıştırma
- Sezgisel akıl yürütme
- Matematik ve mühendislik temelli düşünme

Birleşik Krallık'taki BBC'nin eğitim web sayfasında, BİD'in alt boyutlarını içeren bir rehber yer almaktadır ([www.bbc.com/education](http://www.bbc.com/education)). Bu rehberde BİD dört boyut ile ifade edilmektedir;

- Parçalara ayırma
- Soyutlama
- Örüntü tanıma
- Algoritma

ISTE'ye göre bilgisayar alanındaki gelişmeler, bireylerin yeni beceriler öğrenmelerini ve uygulamalarını gerektirmektedir. ISTE'nin 2016 yılında belirlediği ve 21.yüzyıl bireylerinin kazanması gereken yedi standarttan biri olan Bilgi İşlemsel

Düşünme becerisine sahip olmaktır. Buna göre öğrenciler, problemleri anlamak ve çözmek için teknolojiye son derecede yararlanmalıdırlar. ISTE (2016), tarafından BİD becerisinin alt bileşenleri şöyle sıralanmıştır;

- Verilerin Toplanması
- Verilerin Analizi
- Verilerin Sunulması
- Ayrıştırma
- Soyutlama
- Algoritmalar
- Otomasyon
- Test Etme
- Paralleştirme
- Simülasyon

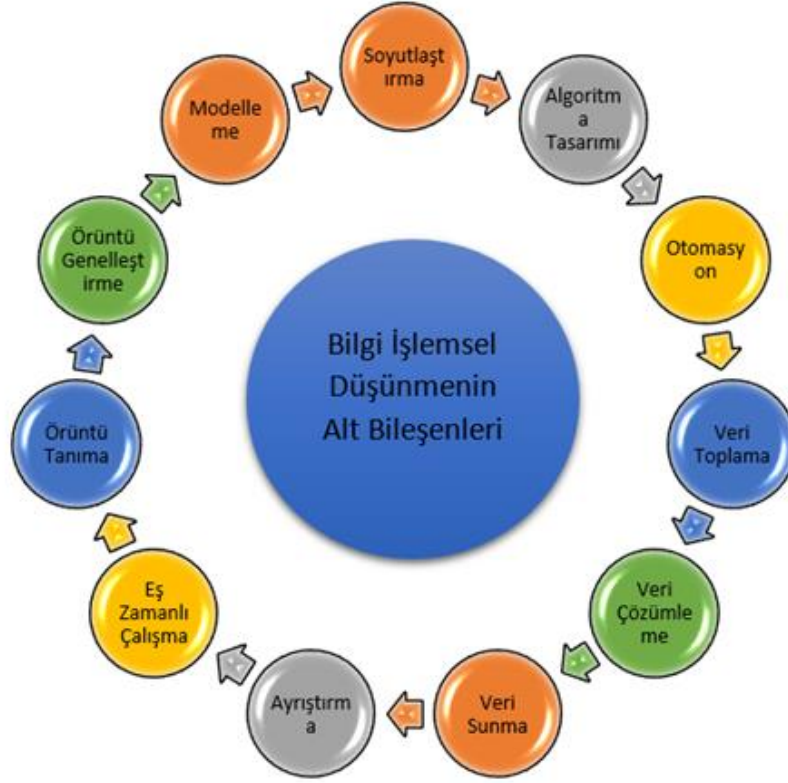
Tosik ve Güyer (2019) BİD alanında sistematik bir alanyazın araştırması yapmışlardır. Araştırmada BİD becerisinin değerlendirilmesinde en çok kabul edilen bileşenlerin şunlar olduğu görülmüştür:

- Soyutlama,
- Algoritmik düşünme,
- Ayrıştırma,
- Test etme ve hata ayıklama,
- Veri okuryazarlığı,
- Sıralama,
- Akış kontrol yapıları

Kalelioğlu ve Gülbahar (2015) tarafından yapılan uluslararası alanyazın incelemeleri sonucunda en sık karşılaşılan bilgi işlemsel düşünme alt boyutlarının şunlar olduğu görülmektedir;

- Problemi anlama
- Parçalara ayırma,
- Örüntü bulma/tanımlama,
- Soyutlama,
- Algoritma
- Test etme/hata ayıklama
- Otomasyon,

- Veri toplama/çözümleme
- Modelleme



Şekil 2.1. BİD Alt Bileşenleri (Kalelioğlu ve Gülbahar, 2015)

## 2.5. Alt Bileşenlerin Tanımları

Bu araştırmada Milli Eğitim Bakanlığı tarafından hazırlanan Bilgi İşlemsel Düşünme Becerisinin Disiplinlerarası Yaklaşım ile Öğretimi (2020) isimli kitapta yer alan BİD alt boyutları temel alınmıştır. Aşağıdaki bölümlerde bu bileşenlere ilişkin açıklamalara yer verilmiştir.

**Veri Düzenleme:** Verilerin düzenlenmesi sürecinde; veri toplama, veri çözümleme ve veri gösterimi gibi işlemler yer almaktadır (MEB, 2020). Veri toplama, nitel ve nicel yöntemler kullanarak elde edilen bilgileri kaydetmektir. Veri toplama sürecine örnek verecek olursak; bir bitkinin büyümesinin gözlemlenmesi sürecinde her gün boyunun ölçülüp yazılması, Mayıs ayı sıcaklık ortalamasını bulmak için hava



sıcaklığının her gün not edilmesi, sınıftaki her gün öğrencilerin kaç sayfa kitap okuduklarının yazılması.

**Veri çözümleme:** Verileri anlamlı hale getirmek için detaylı olarak inceleme, ilişki bulmaya çalışma, örüntü bulup sonuçlar çıkarmaya çalışmaktır. Veri çözümleme sürecine; kitap okuma kayıtlarına bakarak en çok kitap okuyan öğrencinin bulunması, mayıs ayına ait sıcaklık ortalaması, bir sınıftaki öğrencilerin boy ve kilo ortalaması gibi örnekler verilebilir.

**Veri gösterimi:** Veri çözümleme sürecinde yapılan analizlerin görselleştirilmesi olarak ifade edilebilir. Veri gösterimine; bir sınıftaki öğrencilerin boy ve kilo bilgilerini gösteren sütun grafiği çizme, sınava hazırlanan bir öğrencinin günlük çözdüğü soru, doğru ve yanlış soru sayılarını çizelgeye yazması gibi örnekler verilebilir (MEB, 2020).

**Soyutlama:** Bilgisayar biliminin temel kavramlarından biridir. Bir işi yaparken önemli olmayan veya ihtiyaç duyulmayan bilgileri göz ardı etme işlemidir. Bir kitabın özetini anlatmak, yemek tarifi vermek, yol tarif etmek gibi işlemler soyutlama işlemine örnektir. Matematikte sayıları yuvarlama işlemi de bir çeşit soyutlamadır. Pi sayısının virgülden sonraki kısmını atmak buna bir örnektir (MEB, 2020).

**Ayrıştırma:** Çevremizde karşılaştığımız problemler bazen çok karmaşık görünebilmektedir. Karmaşıklığın sebebi problemin aşamalı olmasından ya da alt problemleri olduğundan olabilir. BİD'in bir alt boyutu olan ayrıştırma, bir problemin alt problemlere bölünmesidir. Karmaşık bir problemi 3 tane daha basit alt probleme böldüğümüzü varsayalım, bu alt problemleri çözdüğümüzde aslında problemin kendisi de çözülmüş olacaktır (MEB, 2020). Örneğin aşağıda bir matematik probleminin çözümüne örnek verilmiştir.

*Problem: Sınıf başkanlığı seçiminde Ali 13 oy, Ege Ali'den 7 oy daha az, Ayşe ise Ege'den 5 oy fazla almıştır. 8 oy geçersiz sayılmıştır. Bu bilgilere göre kaç kişi oy kullanmıştır?*

*Çözüm: Ali:13*

*Ege: 13-7=6*

$$\text{Ayşe: } 6+5=11$$

$$\text{Geçersiz oy}=6$$

$$\text{Toplam: } 13+6+11=30 \quad 30-6=24 \text{ kişi oy kullanılmıştır.}$$

Problemi alt problemlere ayırabilen bir öğrenci birden fazla işlem yapabileceğini görebilirken, bu beceriye sahip olmayan bir öğrenci muhtemelen tüm sayıları toplayıp sonuca ulaşmaya çalışacak, ya da bazı işlemleri atlayarak toplama çıkarmalar yapacak ve yanlış sonuç bulacaktır.

**Örüntü Tanıma:** Elde edilen verideki benzer ve farklı özellikleri fark edip ortaya çıkarmak, tekrar eden işlemi belirlemek şeklinde ifade edilebilir. Örüntüyü keşfetmek karmaşık problemlerin kolaylaşmasını sağlar. Bir problemin çözümünde kullanılan yöntemler işe yaradığında bir sonraki seferde bu işlem sırasını tekrar ederek çözümü daha kısa sürede gerçekleştirebiliriz. Matematik dersinde ilkökul müfredatında örüntülerle ilgili konular yer almaktadır (Matematik Dersi Öğretim Programı, 2018). Programcıların da program ve algoritma yazarken en çok kullandığı yöntemlerden biridir. Döngüleri kullanabilmek için önce örüntüyü keşfetmek gerekir. Döngü içindeki komutları yazarken hangi işlemlerin hangi sırada yazılması gerektiğini iyi bir şekilde analiz etmek gerekmektedir (MEB, 2020).

**Eş zamanlı çalışma:** Problemi çözmek ya da bir projeyi tamamlayabilmek için, iş bölümü yaparak aynı anda çalışmayı ifade etmektedir. Örneğin, gösteri sırasında bir grup sahnede dans ederken bazı kişiler müzik, ışık, sahne düzeni, kostüm işleri için çalışır. Herkesin aynı anda ortak bir projenin farklı işlerini yerine getirmesi eş zamanlı çalışmaya örnek bir durumdur. Örneğin akşam yemeği hazırlayan bir anne ana yemeği hazırlayıp fırına verip pişmesini beklerken bir yandan pilavı yaparken aynı anda başka bir tencerede çorbayı da karıştırabilir. Sonuçta 1 saat içinde 3 çeşit yemeği eş zamanlı bir şekilde çalışarak yapmış olacaktır (MEB, 2020).

**Algoritma Tasarımı:** Algoritma, bir işin yapım aşamalarının ayrıntılarıyla yazılmasıdır. Bir yemeğin tarifi, gidilecek adresin anlatımı, bir ders içeriğinin planlanması gibi günlük hayatta yapılan birçok eylem algoritma tasarımına örnek olarak verilebilir. Algoritmalar çoğunlukla bir bilgisayar programını yazmadan önce işlemleri daha net görebilmek ve programın çözümünü ortaya çıkarabilmek için

kullanılır. Genellikle programcının anadilinde emir cümleleri ile yazılır ve bir akış şeması çizilerek şekil ile ifade edilebilir (MEB, 2020).

**Otomasyon:** Bilgisayar veya makineler tarafından bazı işlemlerin sürekli tekrar edilmesidir. Fabrikalardaki paketleme işlemi yapan robot kollar, içecek veren makineler, asansörler gibi programlanarak çalışan sistemler otomasyona örnek olarak verilebilir (MEB, 2020).

**Modelleme:** Gerçek yaşamdaki nesnelerin, olayların şekiller veya çizimler ile ifade edilmesidir. Bilgisayar programları ile 2B ya da 3B modeller yapılabileceği gibi somut nesnelerle de modeller yapılabilir. Simülasyonlar ve animasyonlar da modellemeye örnektir. Deney simülasyonları, animasyonlar ve 3 boyutlu çıktılar model oluşturma örnekleridir (MEB, 2020).

## 2.6. BİD Becerisinin Değerlendirilmesi

BİD becerisini ölçmek üzere araştırmacılar tarafından çeşitli araçlar geliştirilmiştir. Bu bölümde Türkiye’de yapılan bazı araştırmalara yer verilmektedir.

Korkmaz, Çakır ve Özden (2015) tarafından yürütülen bir araştırmada üniversite öğrencileri için geliştirilmiş olan “Bilgisayarca Düşünme Becerileri Ölçeği” ortaokul öğrencilerine uygulanmaktadır. Ölçek beşli likert tipinde, 29 maddeden ve beş faktörden oluşmaktadır. Ölçekte yaratıcılık, algoritmik düşünme, işbirlikçilik ve eleştirel düşünme ve problem çözme alt bileşenleri değerlendirilmiştir. Ölçeğin iç tutarlılık katsayısı 0.82 olmasından dolayı Bilgisayarca Düşünme Beceri Düzeyleri Ölçeği’nin, Türkiye’deki ortaokul öğrencilerinin bilgisayarca düşünme beceri düzeylerinin belirlenmesinde kullanılabilir geçerli ve güvenilir bir ölçek olduğu söylenilmektedir.

Özmen (2016) tarafından Ortaokul Öğrencilerine Yönelik Bilgi İşlemsel Düşünme Becerileri Testi geliştirilmiştir. Bilişim teknolojileri ve yazılım dersi kapsamında algoritma ve temel programlama kavramları konusunda eğitim almış 90 6. sınıf öğrencisine 10 çoktan seçmeli sorudan oluşan bir test uygulanmıştır. Yapılan değerlendirmeler sonunda testin KR-20 değeri .73, standart sapması 2.43 olarak belirlenmiştir. Bu durum testin ortaokul öğrencilerinin BİD becerilerinin ölçülmesinde kullanılabilir geçerli ve güvenilir bir ölçüm aracı olduğunu göstermektedir.

Kaleliođlu ve Glbahar(2018) tarafından Bilgi İřlemsel Dřnme Becerisi z-yeterlik Algısı (BİDBA) leđi geliřtirilmiřtir. Bu lek 3'l likert tipinde, 5 faktr ve 36 maddeden oluřmaktadır. lekte algoritma tasarlama, problem zme, veri iřleme, temel programlama ve zgven boyutları yer almaktadır. Faktrlere ait gvenirlik katsayılarının .93 ile .76 arasında deđiřtiđi grlmekte bu da leđin geerli ve gvenilir bir lek olduđunu dođrulamaktadır.

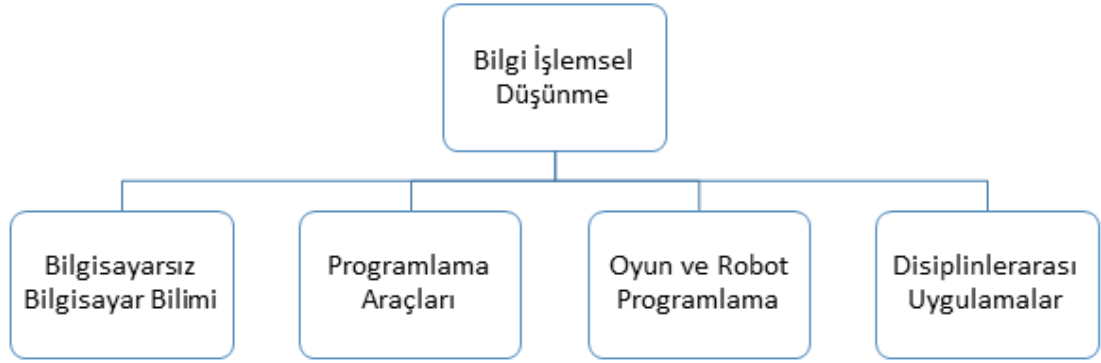
Tosik ve Gyer (2019) tarafından yapılan Bilgi İřlemsel Dřnme Becerisinin Deđerlendirilmesine İliřkin Sistematik Alanyazın Taraması arařtırmasına gre en ok grlen alt bileřenler; soyutlama, algoritmik dřnme, ayırırma, test etme hata ayıklama ve veri okuryazarlıđıdır. BİD becerisini deđerlendirmek amacıyla kullanılan veri toplama yntemlerinin grev, oktan semeli soru, proje, aık ulu soru, grřme, sistem kayıtları, gzlem ve anket olduđu grlmřtir.

Kukul ve Karatař (2019) tarafından Biliřimsel Dřnme z-yeterlik leđi geliřtirilmiřtir. lek 4 faktr ve 18 maddeden oluřmaktadır. Faktrlerde yer alan BİD alt boyutları akıl yrtme, soyutlama, ayırırma ve genellemedir. Uygulanan gvenirlik analizi sonucunda Cronbach Alpha gvenirlik katsayısının .884 olarak hesaplandıđı grlmektedir. Buna gre Biliřimsel Dřnme z-yeterlik leđi'nin, BİD becerisini lmede kullanılabilecek geerli ve gvenilir bir ara olduđu grlmektedir.

Dolmacı ve Akhan (2020) tarafından Biliřimsel Dřnme Becerileri leđi geliřtirilmiřtir. đrencilerin BİD becerilerini lmek amacıyla geliřtirilen bu lek 40 madde ve 5 alt boyuttan oluřmaktadır. Arařtırmada algoritmik-analitik dřnme, yaratıcı problem zebilme, iřbirliđi yapabilme, eleřtirel dřnme ve bilgisayar kullanabilme becerileri deđerlendirilmiřtir. leđin alt boyutlarının gvenirlik katsayıları .74 ile .91 arasında deđiřmekte olduđundan lek gvenilir olarak kabul edilmektedir.

## **2.7. Bilgi İřlemsel Dřnmenin đretilmesinde Kullanılan Aralar**

Weinberg'e (2013) gre BİD'in đrenilmesinde drt farklı yaklařım vardır. Bunlar Bilgisayarsız Bilgisayar Bilimi (B<sup>3</sup>), programlama araları, oyun veya robot programlama ve disiplinler arası uygulamalardır.



Şekil 2.2. BİD öğretiminde kullanılan yaklaşımlar(Bilgi işlemsel düşünmeden programlamaya, 2017)

*Bilgisayarsız Bilgisayar Bilimi:* Bilgisayar bilimini ve BİD alt becerilerini öğretme amacıyla sınıf içinde veya dışında fiziksel oyunlarla, etkileşimli aktivitelerle ve materyaller kullanılarak yapılan etkinliklere denilmektedir. Bilgisayar biliminin temellerinin, programlama ve kodlama mantığının, BİD alt bileşenlerinin bilgisayar kullanılmadan öğretilmesidir (Kalelioğlu, 2017).

*Programlama Araçları:* BİD becerisi kazandırma sürecinde en çok başvurulan öğretim yöntemlerinden biri programlama araçları kullanmaktır. Programlama araçları öğrencilerin seviyesine veya ulaşılabilen teknolojik imkânlarla göre çeşitlilik göstermektedir.

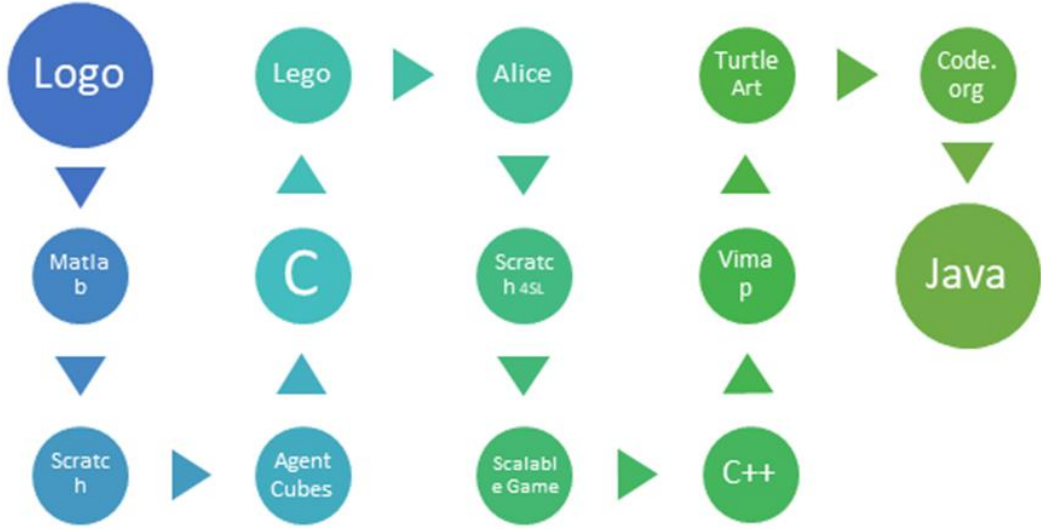
*Blok Tabanlı Programlama araçları:* Kodların bloklar halinde sürüklenerek ard arda eklenmesiyle oluşturulan ve başlangıç seviyesindeki öğrenciler tarafından öğrenmesi kolay olan programlardır. Scratch, Kodlabüyü, Alice, Appinventor blok tabanlı programlama araçlarının başlıcalarındandır (Yükseltürk, 2017).

*Metin Tabanlı Programlama araçları:* Kodların düz yazı şeklinde yazıldığı ve görsel unsurların yer almadığı programlama dilleridir. Örneğin Python, C, C++, Java (Kandemir, 2017).

*Oyun ve Robot Programlama:* Programlama etkinliklerinin somutlaştırılarak yapılmasını sağlayan bu setler BİD'in alt boyutlarının gerçek hayatta kullanımına örnek oluşturmaktadır. En çok kullanılan eğitsel robot setleri şunlardır: Lego Mindstorm EV3, Lego Wedo, Vex, Robotis, Arduino, Mbot (Üçgül, 2017).

*Disiplinlerarası Uygulamalar:* BİD becerisinin STEAM(Bilim, Teknoloji, Mühendislik, Sanat ve Matematik) gibi düşünme ve problem çözme becerisinin öğretildiği derslerde öğretilmesi ve kullanılmasıdır (Yıldız, 2017).

Hsu vd. (2018) yaptıkları literatür taraması çalışmasında BİD'in öğrencilere kazandırılmasında kullanılan stratejileri özetlemiştir. BİD öğretiminde kullanılan öğrenme yaklaşımlarından bazıları şunlardır: probleme dayalı öğrenme, işbirlikçi öğrenme, proje tabanlı öğrenme, oyun tabanlı öğrenme, yapı iskelesi, hikâye anlatımı, hesaplamalı öğrenme teorisi, estetik deneyim, kavram temelli öğrenme, somutlaştırma temelli öğrenme, insan-bilgisayar etkileşimli öğretim ve öğrenme için evrensel tasarım. Bu yaklaşımlara göre BİD öğretiminde kullanılacak örnek araçlar Şekil 2.3'de gösterilmiştir. Bu araçlar içinde ilkökul ve ortaokul düzeyindeki öğrenciler için uygun olan ortamlar; Scratch, Alice, LEGO, Code.org, LOGO, Turtle Art olarak ortaya çıkmıştır.

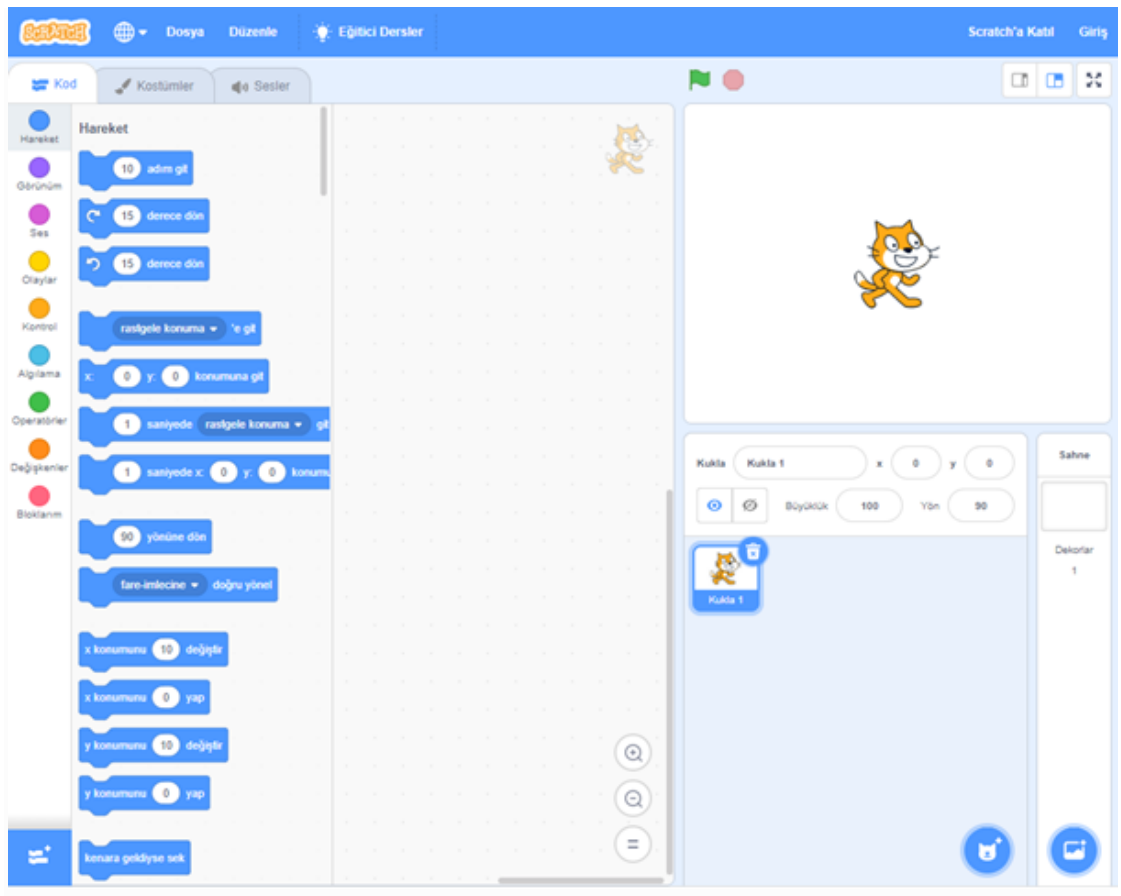


Şekil 2.3. BİD öğretiminde kullanılabilir örnek programlama araçları (Hsu vd., 2018)

## 2.8. Scratch

Scratch 2003 yılında Massachusetts Institute of Technology (MIT) laboratuvarlarında geliştirilmeye başlanmış olan ücretsiz bir görsel programlama aracıdır. Scratch, çocukların temel programlamayı kolayca öğrenebilmesi için geliştirilmiş dünyanın en büyük kodlama topluluğudur

(<https://scratch.mit.edu/about>). Bu ortamda çocuklar dijital hikayeler, oyunlar ve animasyonlar oluşturabilir ve diğer kişilerin görmesi için paylaşabilirler. Basit bir arayüze sahip olması kullanımı çok kolaylaştırmaktadır. Scratch Vakfı tarafından tasarlanmış ve geliştirilmeye devam edilmektedir. Scratch'ın BİD ve problem çözme becerilerini desteklediği pek çok araştırma ile kabul edilmiştir. Ayrıca yaratıcı düşünme, kendini ifade etme ve işbirliği gibi becerileri de geliştirdiği bilinmektedir. Scratch programının çok popüler olmasının sebebi sürükle bırak mantığı ile çalışması ve programlama bilgisi olmayan kişilerin bile kolayca kullanabilmesine imkan tanınmasıdır (Recnick vd., 2009).



Şekil 2.4. Scratch Program Geliştirme Ortamı. (Kaynak: <https://scratch.mit.edu/>)

## 2.9. Scratch ve BİD İlişkisi

Bilgi işlemsel düşünmenin alt becerileri ile programlama yapmanın adımları benzerlik gösterdiğinden programlama yapan kişilerin BİD becerilerinin doğal olarak geliştiği düşünülmektedir. Bu sebeple İngiltere, Amerika gibi pek çok ülke eğitim programlarında programlama öğretimine yer vermektedir. Programlama eğitimine

erken yaşta başlamanın daha iyi sonuçlar vereceği düşünüldüğünden programlama öğretim aracı olarak Scratch tercih edilmektedir. Scratch programı hem görsel olması hem de kod bloklarıyla programlama yapılmasını sağladığı için en düşük seviyeden yüksek seviyeye kadar kullanıcılara hitap etmektedir (Grover ve Pea, 2013a). Sürükle bırak yöntemi ile kodları ekleyerek program yazan küçük yaşta öğrenciler basit bir şekilde programlama öğrenirken aynı zamanda yüksek düzeyli düşünme becerilerini de geliştirmektedir.

BİD becerisini; robotik uygulamaları, bilgisayarsız etkinlikler ve disiplinlerarası çalışmalar da geliştirmektedir (Brennan ve diğerleri, 2011; Grover, & Pea, 2013; Sentance, & Csizmadia, 2015).

Alanyazında Scratch ile programlama öğretiminin BİD becerisi üzerindeki etkisini araştıran çeşitli çalışmalara rastlanmaktadır.

Oluk, Korkmaz ve Oluk (2018) tarafından, ortaokul 5.sınıf öğrencileriyle yürütülen yarı deneysel bir çalışmada deney grubu öğrencilerine 6 hafta süresince Scratch programı kullanılarak algoritma ile ilgili bir öğrenme ve öğretme süreci düzenlerken, kontrol grubu öğrencilerine mevcut müfredata göre algoritma ile ilgili etkinlikler yaptırmıştır. Öğrencilere değerlendirme aracı olarak BİD ölçeği ve algoritma geliştirme başarı testi uygulanmıştır. Çalışma sonunda deney grubu öğrencilerinin algoritma geliştirme ve BİD becerileri düzeyleri, kontrol grubu öğrencilerinininkine göre anlamlı derecede yüksek bulunmuştur. Buna göre Scratch programının BİD becerilerini geliştirme ve algoritma öğretimi konusunda faydalı bir araç olduğu söylenebilir.

Uslu, Mumcu ve Eğin (2018) tarafından yapılan bir araştırmada görsel programlama etkinliklerinin ortaokul 6.sınıf öğrencilerinin BİD becerilerine etkisi araştırılmıştır. 12 haftalık süreç içerisinde öğrencilere Scratch programı ile kodlama etkinlikleri yaptırılmış ve öğrencilerin oyun tasarımları sağlanmıştır. Araştırmada nitel ve nicel verilerin kullanıldığı karma yöntem uygulanmıştır. Nicel verilere göre ölçümler arasında anlamlı bir farklılık olmadığı tespit edilmiş olmasına karşın nitel verilerin analizi sonucunda öğrencilerin bilgisayar bilimine yönelik farkındalıklarının arttığı ve bu sürecin hayal güçlerini geliştirdiğini ifade ettikleri görülmüştür.

Yünkül, Durak, Mısırlı ve Çankaya (2017) tarafından yapılan bir araştırmada Scratch eğitimi alan öğrencilerin Bilgisayarca Düşünme Becerileri Ölçeği (BDBÖ) puanları ile bilişim teknolojileri ve yazılım dersine ait başarı puanları arasındaki ilişki incelenmiştir. 6.sınıfta okuyan öğrencilerle bir dönem süresince yürütülen



deneysel çalışmada deney grubunda Scratch ile ders işlenirken kontrol grubunda Scratch kullanılmadan programlama anlatılmıştır. Deney öncesi öğrencilerin akademik başarı durumları göz önüne alınarak birbirine denk gruplar oluşturulmuştur. Nicel araştırma yönteminin kullanıldığı çalışmada Scratch eğitiminin BDBÖ puanları üzerindeki etkisine bakılarak dönem sonunda Scratch eğitimi alan öğrencilerin BDBÖ puanlarının daha yüksek düzeyde olduğu sonucuna ulaşılmıştır. Bu sonuca göre, Scratch kullanımı BİD becerileri üzerinde etkili olduğu yönünde bulgulara ulaşılmıştır. Ayrıca BDBÖ puanları yüksek bulunan öğrencilerin bilişim dersi akademik başarı puanlarının da yüksek olduğu görülmüştür. Buna göre, araştırmacılar Scratch eğitiminin bilgi işlemsel düşünme becerilerini geliştirmesinden dolayı okullarda kullanımının yaygınlaştırılması önerilmiştir.

Vatansever (2018)'in ortaokul 5. ve 6.sınıf öğrencileri ile yürüttüğü çalışmada Scratch ile programlama öğretiminin öğrencilerin problem çözme becerileri üzerindeki etkisi araştırılmaktadır. Karma yöntem ile gerçekleştirilen çalışmada Scratch ile programlama öğretimi öğrencilerin problem çözme becerileri üzerinde olumlu etki yaratmaktadır. Bu çalışmaya göre Scratch ile programlama öğretiminde tasarım odaklı bir yaklaşım uygulanması önerilmektedir.

Adsay, Korkmaz, Çakır ve Erdoğmuş (2020)'un yürüttükleri bir çalışmada, ortaokul 6,7 ve 8.sınıf öğrencilerinin kodlama eğitime yönelik öz-yeterlik algı düzeyleri, STEM beceri düzeyleri ve bilgi işlemsel düşünme beceri düzeyleri araştırılmıştır. Karma desen ile yapılan çalışmada nicel veriler Blok Temelli Programlamaya İlişkin Öz-yeterlik Algısı Ölçeği, Bilgisayarca Düşünme Beceri Düzeyleri Ölçeği ve STEM Beceri Düzeyleri Algı Ölçeği kullanılarak toplanırken nitel veriler yarı yapılandırılmış görüşme formu ile elde edilmiştir. Çalışmanın sonucuna göre öğrencilerin blok temelli kodlama eğitime yönelik öz-yeterlik algı düzeyi ile bilgisayarca düşünme beceri düzeyleri ve STEM beceri düzeyleri faktörleri arasında anlamlı bir ilişki olduğu görülmektedir. Bilgi işlemsel düşünme beceri düzeyleri ve STEM beceri düzeyleri yükseldikçe, blok temelli kodlama eğitime yönelik öz-yeterlik algıları da artmaktadır. Blok temelli kodlama eğitiminde Scratch yazılımı hakkında öğrencilerin; Scratch yazılımı basit, eğlenceli ve kolay bir program olarak değerlendirdikleri görülmüştür.

## 2.10. Matematik, Scratch ve BİD İlişkisi

ISTE (2016) tarafından ortaya konulan standartlar BİD'i sadece bilgisayar bilimleri alanı ile sınırlamamaktadır. Bilgisayar bilimleri dersinin yanı sıra; matematik, fen ve teknoloji, sosyal bilgiler ve dil alanları ile de ilişkilendirilmiştir. Lockwood ve Mooney (2017) alanyazında Biyoloji, Fizik, Matematik ve İngilizce gibi mevcut derslere BİD becerisinin entegre edilebildiğine yönelik birçok çalışma olduğunu belirtmektedir.

Matematik ve BİD, verilerin analiz edilmesi-yorumlanması ve bilgilerin iletilmesi ile yakından ilişkilidir. Matematiksel yöntemler, veri toplama-analiz araçları ve görselleştirmeler, öğrencilerin büyük miktarlarda veriyle çalışabilmeleri için kolaylık sağlar. Ayrıca öğrenciler, bulguları metin veya konuşma diliyle ifade etmekte zorlandıklarında matematiksel temsilleri, veri görüntülerini, simülasyonları ve grafik temsilleri kullanırlar (Wilkerson, 2017). Matematik dersinde kullanılan bu alt araçlar BİD alt bileşenleri ile örtüşmekte ve iki alanın birbiri ile bağlantılı olduğunu göstermektedir.

Alanyazında matematik etkinlikleri yoluyla BİD becerilerinin gelişiminde Scratch'in kullanıldığı pek çok araştırma yürütülmüştür. Sung, Ahn ve Black (2017) tarafından yürütülen bir çalışmada çeşitli derecelerde somutlaştırılmış etkinlikler yoluyla öğrencilere BİD becerisi kazandırılması amaçlanarak, matematik ve programlamadaki başarı düzeyine etkisi incelenmiştir. 66 anaokulu ve birinci sınıf öğrencisi ile yürütülen bu deneysel çalışmanın ilk aşamalarında öğrenciler bedensel etkinlikler ve kâğıt, çubuk gibi somut materyallerle matematik işlemlerini öğrenirken, son aşamada öğrencilerden aynı görevleri yardım etmeksizin Scratch Jr ile yapmaları istenmiştir. Çalışmanın sonuçlarına göre, BİD ile desteklenmiş matematik problemini çözme etkinlikleri, öğrencilerin matematik anlama ve Scratch Jr'da programlama becerilerini geliştirdiğini göstermektedir.

Scratch etkinlikleri ile zenginleştirilen matematik dersi öğretiminin başarıya olan etkisini araştıran pek çok araştırma vardır. Örneğin Okuducu (2020), tarafından yürütülen deneysel çalışmada Scratch yazılımı kullanımının öğrencilerin cebirsel ifadeler konusundaki akademik başarısına ve cebir tutumuna etkisi incelenmiştir. Deneysel gruba Scratch destekli öğrenme etkinlikleri uygulanırken, kontrol grubuna öğretim programına uygun dersler uygulanmıştır. Nicel veri toplama yöntemleriyle elde edilen bulgulara göre Scratch ile yapılan ders etkinliklerinin cebirsel ifadeler

konusundaki başarısı ve cebir tutumu üzerinde olumlu yönde bir fark oluşturduğu görülmüştür. Nitel verilere göre öğrenciler Scratch uygulamalarını ilgi çekici, başarıyı arttırıcı ve eğlenceli bulduklarını ifade etmişlerdir. Bu araştırmaya göre Scratch destekli öğrenme etkinliklerinin matematik dersinde etkili olduğu ve diğer konularda da uygulanabilir olduğu sonucuna ulaşılmıştır.

Çubukluöz (2019) tarafından 6. Sınıf seviyesindeki öğrenciler ile yapılan bir araştırmada Scratch ile tasarlanan matematiksel oyunlarla öğrencilerin derste yaşadıkları zorlukların giderilmesi amaçlanmıştır. Öğrencilere araştırmacı tarafından 6 hafta boyunca kodlama eğitimi verilerek Scratch programlama aracıyla matematiksel oyun hazırlamaları sağlanmıştır. Araştırmada nitel yaklaşıma dayalı eylem araştırması yöntemi kullanılmıştır. Araştırmanın sonunda, öğrencilerin Scratch programında matematiksel oyunları tasarlama ile ilgili olumlu görüşlere sahip oldukları ortaya çıkmıştır. Öğrencilerin büyük bir çoğunluğunun matematik dersinde yaşadıkları öğrenme zorluklarının Scratch programlama aracı ile tasarlanan matematiksel oyunlarla giderildiği ortaya çıkmıştır. Özellikle, öğrencilerin uygulamadan sonra dört işlem önceliğini kavradıkları, üslü ifadelerle ilgili işlemleri doğru yapabildikleri, matematiksel problem çözme becerilerinin geliştiği, değişken kavramını, asal çarpanları öğrendikleri tespit edilmiştir.

Deryal (2021) tarafından yapılan bir araştırmada ortaokul 5. ve 6. sınıf öğrencilerinin BİD beceri düzeyleri ile matematiksel problem çözme başarıları arasındaki ilişkinin ortaya çıkarılması amaçlanmıştır. Araştırmada ilişkisel tarama modeli benimsenmiş ve örneklem seçiminde gelişigüzel örneklem seçilmiştir. Araştırma verileri, “Genel Bilgiler Formu”, “Matematiksel Problem Çözme Başarı Testi” ve “Bilgi İşlemsel Düşünme Testi” ölçekleri kullanılarak temin edilmiştir. 5. ve 6. sınıf öğrencilerinin BİD beceri düzeyleri birbirine yakın bulunurken, 6. sınıf öğrencilerinin matematiksel problem çözme düzeyleri 5. sınıf öğrencilerinden daha yüksek bulunmuştur. Kodlama eğitimi alan öğrencilerin hem BİD beceri düzeyleri hem de matematiksel problem çözme başarı puanı ortalamaları, bu eğitimi almayan öğrencilerden daha yüksek bulunmuştur. Benzer şekilde okul dışı zamanlarda blok kodlama etkinlikleri yapan öğrencilerin BİD beceri düzeyleri ve matematiksel problem çözme başarı puanı ortalamaları yapmayan öğrencilere göre yüksek bulunmuştur.

Araştırmalara bakıldığında genel olarak Matematik dersi ile Scratch programı ya da kodlama araçlarının birlikte kullanılmasının bir avantaj sağladığı

görülmektedir. Scratch ile birlikte Matematik öğretimi yapıldığında geleneksel yöntemlere göre daha başarılı olduğu sonucuna ulaşılmıştır (Okuducu,2020); Çubukluöz,2019; Deryal,2021). Öğrencilerin kodlamayı daha somut, kolay ve anlaşılır buluyor olmalarının matematik dersinde aktif öğrenmeyi sağladığı ifade edilmektedir (referans?). Matematiğe karşı var olan olumsuz tutumu yumuşattığı da çıkarımı yapılabilir. Programlama dersinde öğrenilen ayrıştırma, soyutlama, modelleme, simülasyon, örüntü tanıma veri düzenleme gibi alt işlemlerin matematik dersinde de sıklıkla kullanılıyor olması matematik konularını anlamayı kolaylaştırabilir.

### 3. YÖNTEM

Bu bölümde araştırma yöntemi, çalışma grubu, veri toplama araçları ve deneysel işlemden bahsedilmiştir. Araştırma dört hafta etkinlik uygulama süreci ve iki hafta testlerin uygulanma süreci olmak üzere toplam altı hafta olacak şekilde planlanmıştır. Çalışma tamamlandıktan sonra ön test sonuçlarının kontrol değişkeni olduğu ve son test sonuçlarının bağımlı değişken olduğu ANCOVA analizi ile gruplar arasındaki fark incelenmiştir.

#### 3.1. Araştırmanın Modeli

Araştırmada ön test son test kontrol gruplu yarı deneysel desen kullanılmıştır. Elde edilen nicel veriler ANCOVA analizi ile araştırma modeli Tablo 3.1’de sunulmuştur.

Tablo 3.1. Uygulama süreci

	Deney Grubu	Kontrol Grubu
1.Hafta	Ön Test	Ön Test
2.Hafta	İşlem Önceliği konulu etkinlik	Başkent Bulma Oyunu
3.Hafta	Üslü Sayılar Etkinliği	Elma Toplama Oyunu
4.Hafta	Bölünebilme Kuralları Etkinliği	İngilizce Kelime Oyunu
5.Hafta	Asal Sayılar Etkinliği	Burcunu Bul Oyunu
6.Hafta	Son Test	Son Test

Tablo 3.1’de görüldüğü üzere, veri toplama sürecinde öncelikle öğrencilere ön test olarak BİD testi ve BİD özyeterlik algısı ölçeği uygulanmıştır. Ardından deney grubuna 4 hafta matematik destekli programlama anlatılmış ve örnekler yapılmıştır. Matematik destekli örnekler öğrencilerin 6.sınıf müfredatında yer alan ve ilk dönemin ilk aylarında öğrenmiş oldukları matematik konularını içermektedir. Aynı

süreçte kontrol grubuna öğretim programına uygun olarak oyun hazırlama etkinlikleri yaptırılmıştır. 4 haftalık sürecin sonunda söz konusu ölçme araçları son test olarak uygulanmıştır.

### **3.2. Çalışma Grubu**

Araştırmanın çalışma grubu, Samsun ilinin bir ilçesindeki devlet ortaokulunda 2021-2022 eğitim öğretim yılında 6.sınıfta okuyan 8 farklı şubede öğrenim gören 200 öğrenci oluşturmuş, bu öğrencilerden ön test – son test verileri tam olan ve kaynaştırma öğrencisi olmayan 176'sının verisi nicel analizlerde kullanılmıştır. Deney ve kontrol grupları halihazırda kurumdaki sınıflara göre oluşmuştur. Bu göre 6A, 6B, 6C, 6D sınıfları kontrol, 6E, 6F, 6G, 6H sınıfları da deney grubu olarak belirlenmiştir. Deney grubunda 46 kız 43 erkek olmak üzere toplam 89 öğrenci bulunmaktadır. Kontrol grubunda ise 47 kız 40 erkek olmak üzere 87 öğrenci çalışmaya katılmıştır.

Öğrenciler araştırmaya katılmadan önce 5. sınıfta, bilişim teknolojileri ve yazılım dersini birinci ve ikinci dönem MEB öğretim programına uygun şekilde pandemi koşulları nedeniyle çevrimiçi olarak almıştır. Beşinci sınıfın ikinci dönem konuları arasında yer alan programlamaya giriş konularında code.org, kodlabüyü ve Scratch ortamlarında kodlama ve programlama eğitimine yönelik etkinlikler gerçekleştirmiştir. Bu bağlamda, çalışma grubundaki öğrencilerin programlama konusunda bir ön bilgilerinin olduğunu söylenebilir.

### **3.3. Veri Toplama Araçları**

Bu çalışmada veri toplama aracı olarak BİD Testi ve BİD Becerisine Yönelik Öz Yeterlik Algısı Ölçeği uygulanmıştır.

Öğrencilerin uygulama öncesi ve sonrası BİD beceri seviyelerini ölçmek için kullanılan araç, Román-González, Pérez-González ve Jiménez-Fernández (2017) tarafından geliştirilen ve Çetin, Otu ve Oktaç (2020) tarafından Türkçe'ye çevrilerek uyarlama çalışmaları tamamlanan BİD Testidir. Diğer ölçme aracı ise Gülbahar, Kert ve Kalelioğlu (2018) tarafından geliştirilen Bilgi İşlemsel Düşünme Becerisine Yönelik Öz Yeterlik Algısı Ölçeği (BİDBÖA) dir.

### 3.3.1. BİD Testi

BİD Testi, Román-González ve diğerleri (2017) tarafından geliştirilen bilgisayarsız kodlama ve görsel kodlama araçlarını barındıran, 7 bölüm ve 28 maddeden oluşan bir testtir. Programlama ve kodlama bağlamında BİD düzeyini belirlemek amacıyla geliştirilmiştir. Test, programlama dillerinin döngüler, şartlı yapılar, değişken ve dizi kavramı, fonksiyonlar gibi temel kavramlarını kullanarak problemleri çözme ve formülleştirme yeteneklerini ölçmektedir. Testin özgün formu, 12-14 yaş aralığındaki (7. ve 8. sınıf) öğrenciler için geliştirilmiştir. Ancak testi geliştirenler tarafından, 5.-6. sınıf öğrencileri ve 9.-10. sınıf öğrencileri için de testin kullanılabilmesi ileri sürülmektedir. Tablo 3.2’de testteki bölümler ve soru sayıları verilmiştir.

Tablo 3.2. BİD testi boyutları ve soru sayıları

Test boyutu	Soru aralığı
Temel Sıralama	1-4
Belirli sayıda tekrar eden döngüler	5-8
Şart gerçekleşene kadar çalışan döngüler	9-12
Basit şartlı ifadeler	13-16
Karmaşık şartlı ifadeler	17-20
Şart gerçek oldukça çalışan döngüler	21-24
Basit fonksiyonlar	25-28

Tablo 3.2 incelendiğinde; temel sıralama (4 soru), belirli sayıda tekrar eden döngüler (4 soru), şart gerçekleşene kadar tekrar eden döngüler (4 soru), basit şartlı ifadeler (4 soru), karmaşık (if-else) şartlı ifadeler (4 soru), şart gerçek oldukça çalışan döngüler (while) (4 soru), basit fonksiyonlar (4 soru) olmak üzere 28 sorunun olduğu görülmektedir (Çetin, 2020).

### 3.3.2. Bilgi İşlemsel Düşünme Becerisine Yönelik Öz Yeterlik Algısı Ölçeği

Araştırmada kullanılan diğer bir ölçme aracı da Gülbahar, Kert ve Kalelioğlu (2018) tarafından geliştirilen BİD Becerisine Yönelik Öz Yeterlik Algısı Ölçeğidir (BİDBÖA). Bu ölçek 5 faktör ve 36 maddeden oluşmaktadır: (a) algoritma tasarlama yeterliği (9 madde), (b) problem çözme yeterliği (11 madde), (c) veri işleme yeterliği (7 madde), (d) temel programlama yeterliği (6 madde), (e) özgüven yeterliği (6 madde) (Tablo 3.3).

Tablo 3.3. BİDBÖA faktörler ve madde sayıları

Bilgi İşlemsel Düşünme Kavramı	Soru aralığı
Algoritma tasarlama yeterliği	1-9
Problem çözme yeterliği	10-19
Veri işleme yeterliği	20-26
Temel programlama yeterliği	27-31
Özgüven yeterliği	32-36

Ölçekte yanıtlar 3'lü Likert tipindedir (Evet-1, Kısmen-2, Hayır-3). Algoritma tasarlama yeterliliği boyutunda 9 madde yer almaktadır ve bu boyuta ait güvenilirlik katsayısı .93 olarak hesaplanmıştır. Bu değer ilgili boyutta yüksek güvenilirliğe işaret etmektedir. Problem çözme yeterliği boyutunda 10 madde bulunmakta ve güvenilirlik katsayısı .88 olarak hesaplanmıştır. Bu değer ilgili boyutun güvenilirliğinin yüksek olduğunu göstermektedir. Veri işleme yeterliği boyutunda 7 madde bulunmakta ve güvenilirlik katsayısı .85 olarak hesaplanmıştır. Bu sonuç bu boyutun güvenilirliğinin yüksek olduğunu göstermektedir. Temel programlama yeterliği boyutunda 5 madde yer almakta ve güvenilirlik katsayısı .83 olarak hesaplanmaktadır. Bu değer ilgili boyutta yüksek güvenilirlik bulunduğunu göstermektedir. Özgüven yeterliği boyutunda 5 madde yer almakta ve bu boyuta ait güvenilirlik katsayısı .76 olarak hesaplanmaktadır. Bu değer ilgili boyutta yüksek güvenilirliğe işaret etmektedir.



### 3.4. Deneysel İşlem

Araştırmada deney grubunda kullanılan etkinlikler Matematik öğretmenlerinden görüş alınarak hazırlanmıştır. Bölünebilme kuralları ile ilgili etkinlikte Scratch programının kısıtlı imkanlarından dolayı bölünebilme kuralları matematik dersinde öğretildiği gibi uygulanamamıştır. Örneğin 3 ile tam bölünebilme kuralı; “Bir doğal sayının basamaklarındaki rakamların sayı değerleri toplamı 3 ile kalansız (tam) bölünüyorsa bu sayı 3 ile kalansız (tam) bölünebilir” şeklinde tanımlanmaktadır. Ancak Scratch programında klavyeden girilen sayıların basamaklarındaki rakamların 3’e tam bölünebildiğine bakma imkanımız olmadığından; bunun yerine bu sayının 3’e bölümünden kalanı veren MOD komutu kullanılarak uygulanmıştır.

#### 3.4.1. Deney Grubu Etkinlikleri

Deney grubundaki öğrencilere dört etkinlik uygulanmıştır: (a) doğal sayılarda işlem önceliği, (b) üslü ifadeleri hesaplama, (c) bölünebilme kuralları, (d) asal sayılar. Bu etkinlikler ile ilgili açıklamalar sırasıyla aşağıda verilmiştir.

##### 3.4.1.1. Doğal Sayılarda İşlem Önceliği Etkinliği

*Etkinliğin Tanımı:* Scratch ortamında işlem önceliği kuralına göre işlemleri sıralama oyunu

*Etkinliğin Amacı:* Bu etkinlik ile öğrenci yeni bir karakter yaratmayı, karakterini konuşmaması ve kılık değiştirmesini, gizleyip göstermeyi, haber salma ve haber alma komutlarını kullanmayı, değişken kullanmayı, rastgele sayı üretmeyi, döngü ve şart yapısını kullanarak işlem yapmayı öğrenir.

*Ders Süresi:* 80 dk.

*Gerekli Ön Bilgiler:* Scratch ara yüzünü ve komutlarını bilme, değişken kavramını ve döngü yapısını bilme.

*Kurallar:*

- Matematiksel fonksiyonlar kullanılmamalı
- Döngü ile sırlamanın doğruluğu kontrol edilmeli
- Kukla kullanıcıya gerekli açıklamaları yapmalı

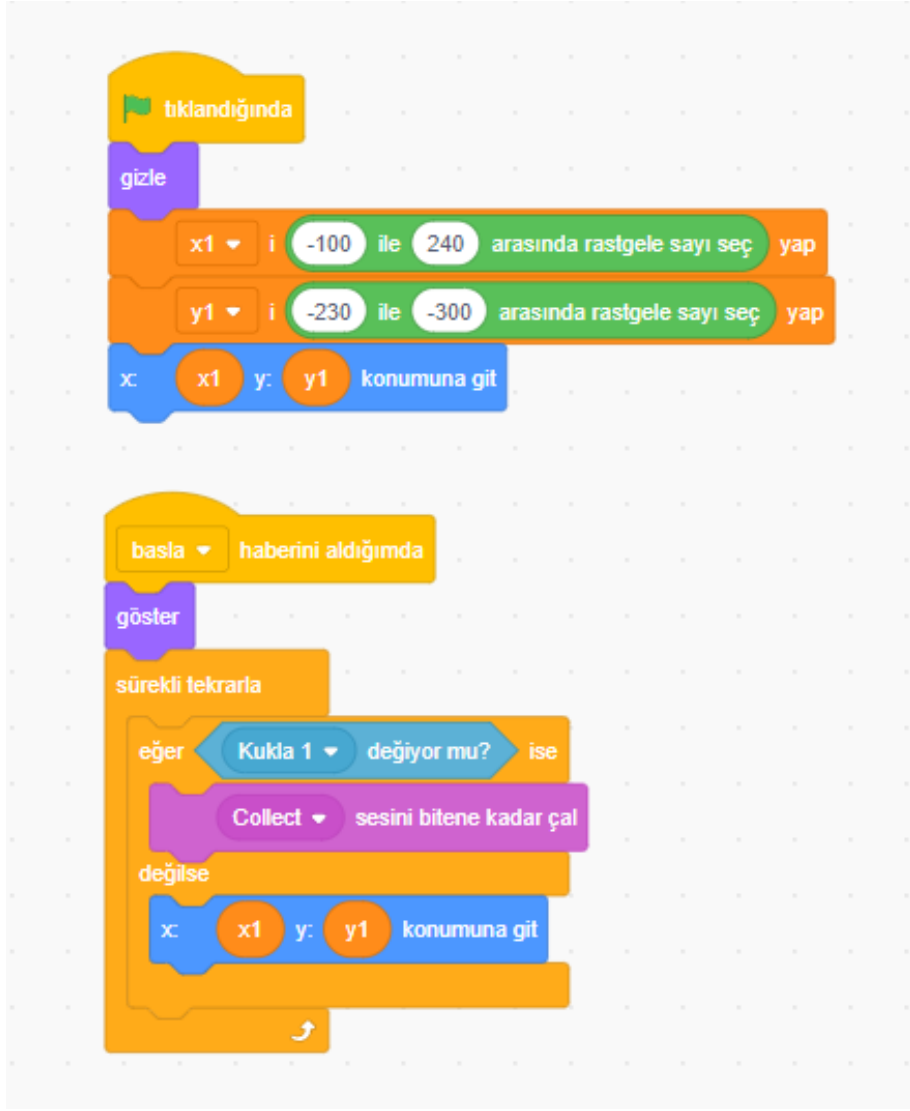
*Etkinliğin Ekran Görüntüsü ve Kod Örnekleri:*



Şekil 3.1. Doğal sayılarda işlem önceliği etkinliği, ekran çıktısı



Şekil 3.2. Doğal sayılarda işlem önceliği etkinliği, kuklaya yazılan kodlar



Şekil 3.3. Doğal sayılarda işlem önceliđi etkinliđi, işlem isimlerine ait karakterlere yazılan kodlara bir örnek

Bu etkinlik ile öğrenci doğal sayılarda işlem önceliđi konusunun pekiştirilmesini sağlayacak bir oyun tasarlamaktadır. Oyunun amacı kullanıcının işlemleri doğru şekilde sıralamasını sağlamaktır. Kullanıcının görevi oyun başladığında ekrana rastgele dağılan işlem isimlerini gösterilen yere doğru sırayla yerleştirmeye çalışmaktır. Öğrenci kullanıcının yaptığı sıralamanın doğruluđunu her an kontrol etmek zorundadır. Bunun için arka plana başka bir karakter yerleştirerek işlem ismi ile karakterin doğru şekilde eşleşip eşleşmediđini kontrol etmelidir. Tüm işlem isimleri istenen karakterlerle doğru şekilde eşleştirildiyse oyun sonlanmaktadır. Bu etkinlikte BİD'in alt bileşenlerden algoritmik düşünme, soyutlama, sıralama, test etme ve hata ayıklama becerileri geliştirilmeye çalışılmıştır.

### 3.4.1.2. Üslü İfadeleri Hesaplama Etkinliđi

*Etkinliđin Tanımı:* Scratch ortamında kullanıcının gireceđi bir sayının, yine kullanıcının gireceđi sayı kadar üslü ifadesini hesaplayan ve sonucu ekrana yazdıran bir program yazma.

*Etkinliđin Amacı:* Bu etkinlik ile öđrenci döngü yapısını kullanarak işlem yapmayı öđrenir. Klavyeden girilen sayıyı okumayı, bunu bir deđiřkene aktarmayı, oluřturduđu döngünün tekrar sayısının deđiřkene bađlı olacađını öđrenir.

*Ders Süresi:* 80 dk.

*Gerekli Ön Bilgiler:* Scratch ara yüzünü ve komutlarını bilme, deđiřken kavramını ve döngü yapısını bilme.

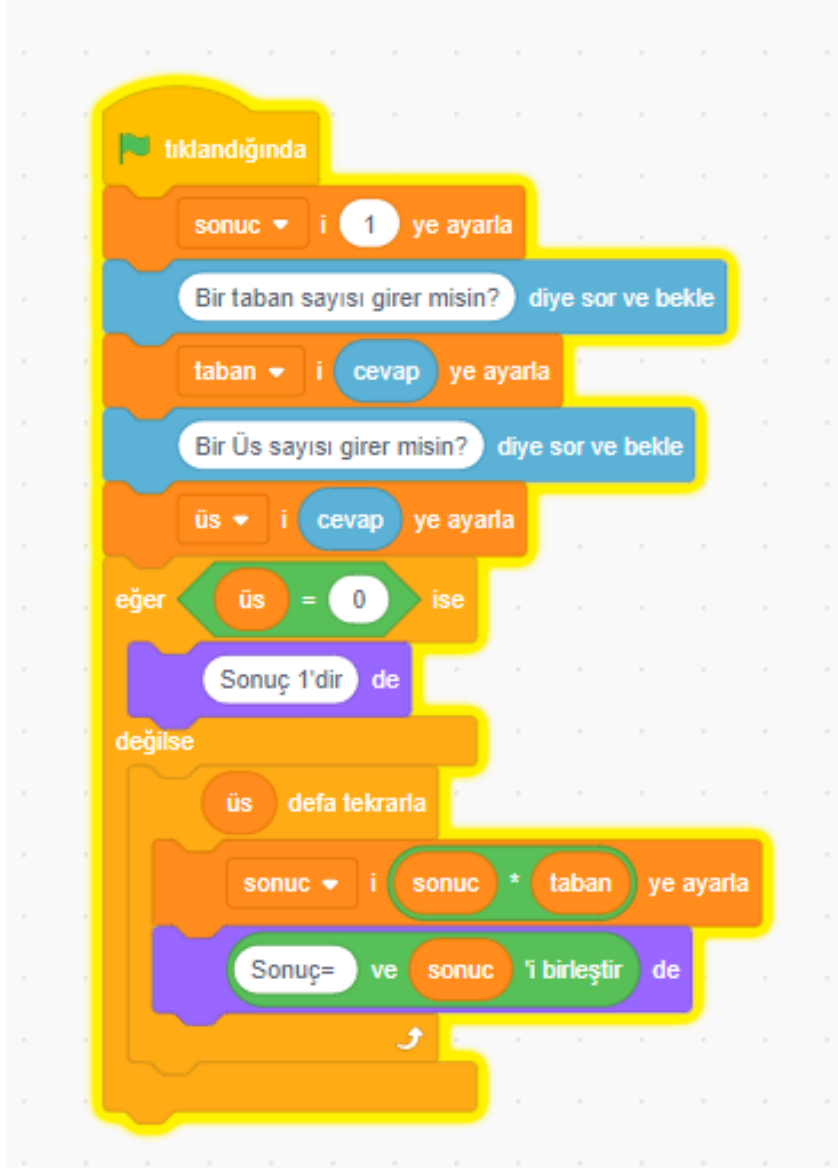
*Kurallar:*

- Matematiksel fonksiyonlar kullanılmamalı
- Döngü ile üslü sayı hesaplanmalı
- Kukla kullanıcıya gerekli açıklamaları yapmalı

*Etkinliđin Ekran Görüntüsü ve Kod Örnekleri:*



Şekil 3.4. Üslü ifadeleri hesaplama etkinliđi, oyunun ana ekran görüntüsü



Şekil 3.5. Üslü ifadeleri hesaplama etkinliği, karaktere yazılan kodlar

Bu etkinlik ile öğrencilerin üslü ifadelerin nasıl hesaplanacağını öğrenmeleri ve önceden öğrendikleri bilgilerin pekiştirilmesi amaçlanmaktadır. Programı yazarken öğrenci hesaplayacağı üslü ifadenin taban ve üs sayısını kullanıcıdan isteyecek. Bu bilgileri kaydetmesi için bir nesneye ihtiyacı olduğunu fark edecek ve değişken kullanacaktır. Üslü ifadeyi hesaplamak için taban sayısını girilen üs sayısı kadar çarpması gerektiğini, bu işlemi yapmak için döngü kullanması gerektiğini fark edecektir. Bu etkinlik sayesinde öğrencinin algoritmik düşünme, şart ifadeleri, döngüler ve değişkenler gibi programlama becerileri ile matematiksel işlem becerilerinin geliştirilmesi hedeflenmiştir.

### 3.4.1.3. Bölünebilme Kurallarını Kullanma Etkinliği

*Etkinliğin Tanımı:* Scratch ortamında kullanıcının gireceği bir sayının, 2, 3, 4, 5, 6, 9 ve 10'a tam bölünüp bölünmediğini bulup sonucu ekrana yazdıran programı yazma.

*Etkinliğin Amacı:* Bu etkinlik ile öğrenci şart yapısını, dizi kavramını ve matematik fonksiyonu olan MOD komutunu kullanmayı ve metin birleştirmeyi öğrenir. Klavyeden girilen sayıyı okumayı, bunu bir değişkene aktarmayı, mod komutu ile sayının hangi sayılara kalansız bölünebildiğini tespit etmeyi ve bunları bir diziye yazmayı, şart yapısını kullanmayı ve dizinin elemanlarını ekrana bir metin olarak yazdırmayı öğrenir.

*Ders Süresi:* 80 dk.

*Gerekli Ön Bilgiler:* Scratch ara yüzünü ve komutlarını bilme, değişken kavramını ve döngü yapısını bilme.

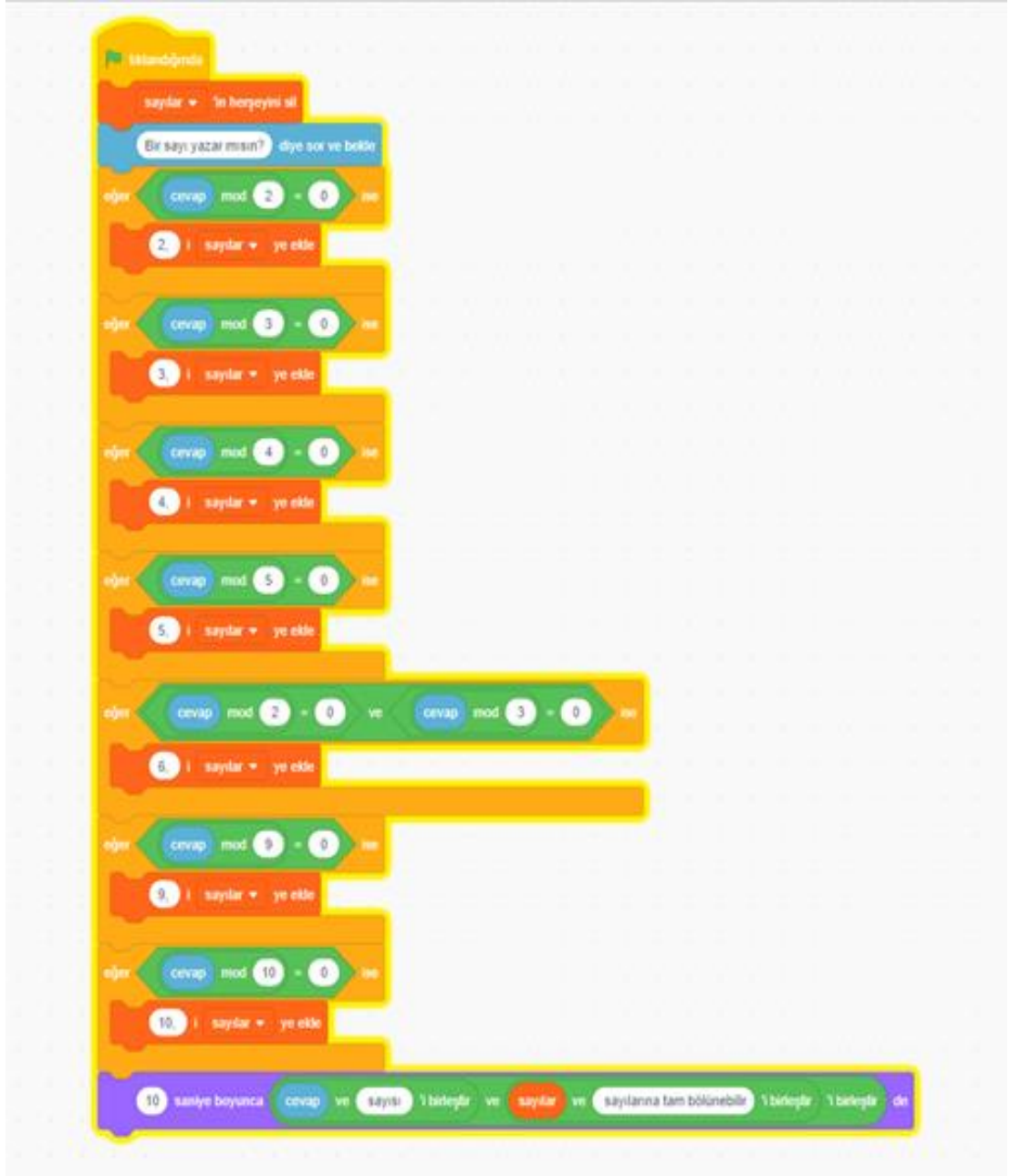
*Kurallar:*

- Matematiksel fonksiyonlardan MOD komutu kullanılmalı
- Şart yapısı kullanılmalı
- Kukla kullanıcıya gerekli açıklamaları yapmalı
- Bulunan sayılar bir diziye yazdırılmalı
- Dizi elemanları bir metin olarak ekrana yazdırılmalı

*Etkinliğin Ekran Görüntüsü ve Kod Örnekleri:*



Şekil 3.6. Bölünebilme kurallarını kullanma etkinliği, programın ekran görüntüsü



Şekil 3.7. Bölünebilme kurallarını kullanma etkinliği, karaktere yazılan kodların görüntüsü

Bu etkinlikle amaçlanan öğrenciye bölünebilme kurallarını kullanarak oyun tasarlatmaktır. Bölünebilme kurallarının özelliklerini (2'ye bölünme için son rakamın çift olup olmadığına bakmak, 5'e bölünme için son rakamın 0 ya da 5 olup olmadığına bakmak gibi) kullanabilmek için Scratch programı yeterli gelmediğinden dolayı mod komutunun kullanılması tercih edilmiştir. Öğrenci girilen sayının belirtilen sayılara bölünüp bölünmediğini mod komutu ile kontrol edip sonuç olumlu ise o sayıyı diziye yazdırmayı ve en sonunda dizi elemanlarını ekrana yazdırmayı öğrenmiştir. Bu etkinlikte bilgi işlemsel düşünmenin alt boyutlarından olan

soyutlama, algoritmik düşünme, sıralama, hata tespit etme ve giderme, veri düzenleme becerilerinin kazandırılması amaçlanmıştır.

#### 3.4.1.4. Asal Sayıları Bulma Etkinliği

*Etkinliğin Tanımı:* Scratch ortamında kullanıcının gireceği bir sayının, asal sayı olup olmadığını bulup sonucu ekrana yazdıran programı yazma.

*Etkinliğin Amacı:* Bu etkinlik ile öğrenci değişkenleri, şart yapısını, döngü yapısını, “veya” ifadesini, matematik fonksiyonu olan MOD komutunu kullanmayı ve metin birleştirmeyi öğrenir. Klavyeden girilen sayıyı okumayı, bunu bir değişkene aktarmayı, döngüyü şartlı olarak çalıştırmayı, MOD komutu ile sayının hangi sayılara kalansız bölünebildiğini bulup şart yapısını kullanarak asal olup olmadığına karar vermeyi ve iki metni birleştirerek ekrana yazdırmayı öğrenir.

*Ders Süresi:* 80 dk.

*Gerekli Ön Bilgiler:* Scratch ara yüzünü ve komutlarını bilme, değişken kavramını ve döngü yapısını bilme.

*Kurallar:*

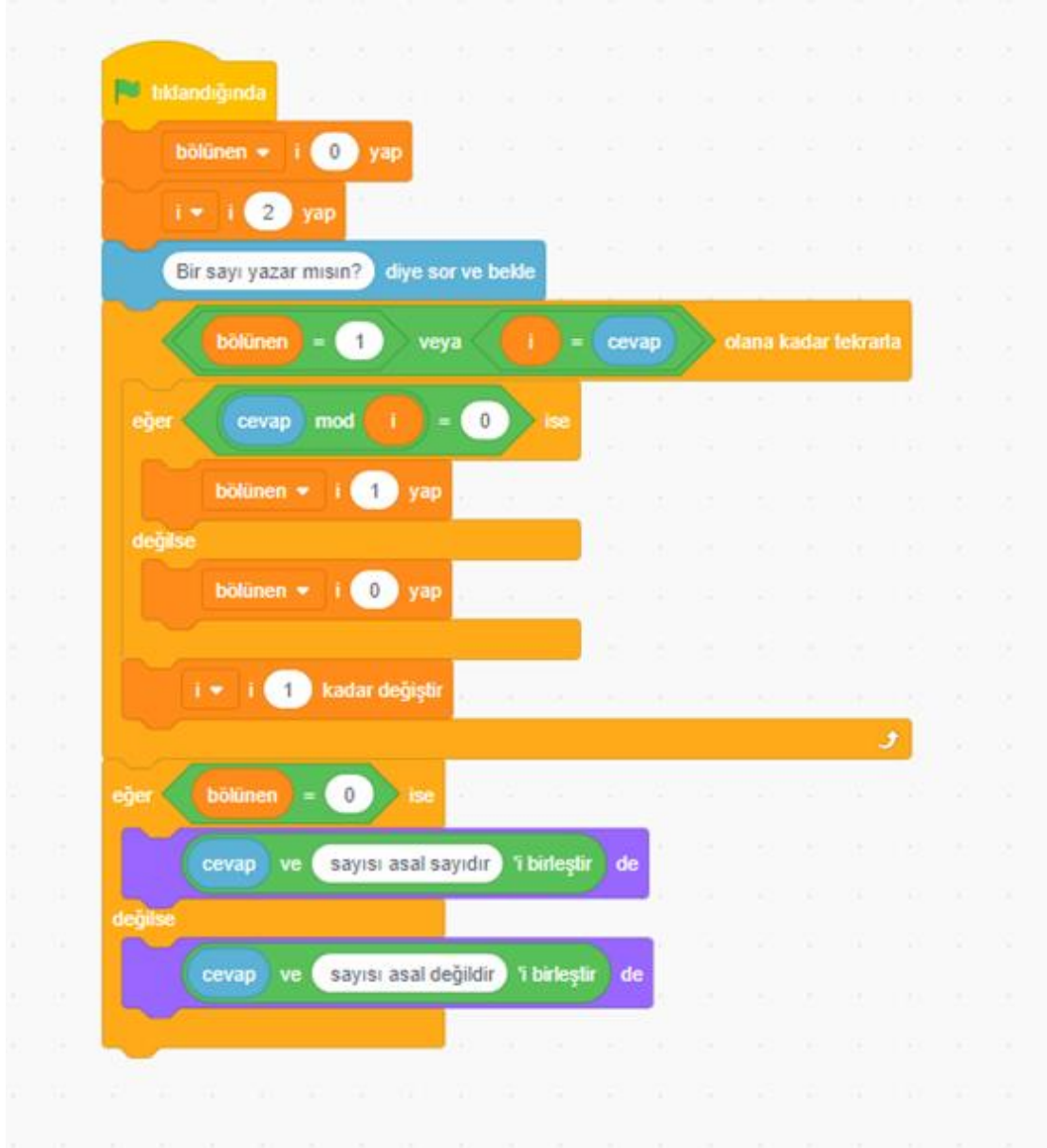
- Kullanıcının girdiği sayı değişkene yazdırılmalı
- Asal sayı olup olmadığı bulunana kadar döngü çalıştırılmalı
- Matematiksel fonksiyonlardan MOD komutu kullanılmalı
- Şart yapısı kullanılmalı
- Sonucu ekrana yazdırmalı

*Etkinliğin Ekran Görüntüsü ve Kod Örnekleri:*



Şekil 3.8. Asal sayıları bulma etkinliği, programın ana ekran görüntüsü





Şekil 3.9. Asal sayıları bulma etkinliği, karaktere yazılan kodların görüntüsü

Bu etkinliğin amacı öğrenciye kullanıcı tarafından girilen bir sayının asal olup olmadığını bulması ve sonucu ekrana yazdıracak programı tasarlatmaktır. Öğrenci bu etkinlikte mantıksal yapıları, karar yapılarını ve döngüleri kullanarak matematik dersinde öğrendiği bir işlemi bilgisayara nasıl yaptıracağını düşünmektedir. Bu sayede algoritmik düşünme, soyutlama, sıralama, veri düzenleme ve problem çözme becerilerini geliştirmektedir.

### 3.4.2. Kontrol Grubu Etkinlikleri

Kontrol grubundaki öğrencilere dört etkinlik uygulanmıştır: (a) başkent bulma oyunu, (b) elma toplama oyunu, (c) ingilizce kelime oyunu, (d) burçları bulma oyunu. Bu etkinlikler ile ilgili açıklamalar sırasıyla aşağıda verilmiştir.

#### 3.4.2.1. Başkent Bulma Oyunu Etkinliği

*Etkinliğin Tanımı:* Scratch ortamında verilen ülkenin başkentini bilme oyunu yapma

*Etkinliğin Amacı:* Bu etkinlik ile öğrenci karakter ve arka plan eklemeyi, karakterini konuşturmayı, haber salma ve haber alma komutlarını kullanmayı, değişken kullanmayı ve şart yapısını kullanarak işlem yapmayı öğrenir.

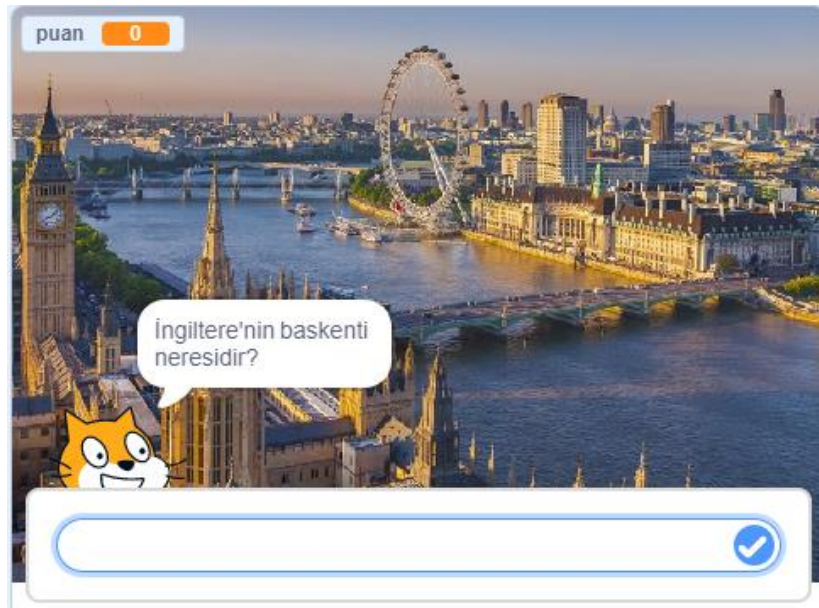
*Ders Süresi:* 80 dk.

*Gerekli Ön Bilgiler:* Scratch ara yüzünü ve komutlarını bilme, değişken kavramını ve şart yapısını bilme

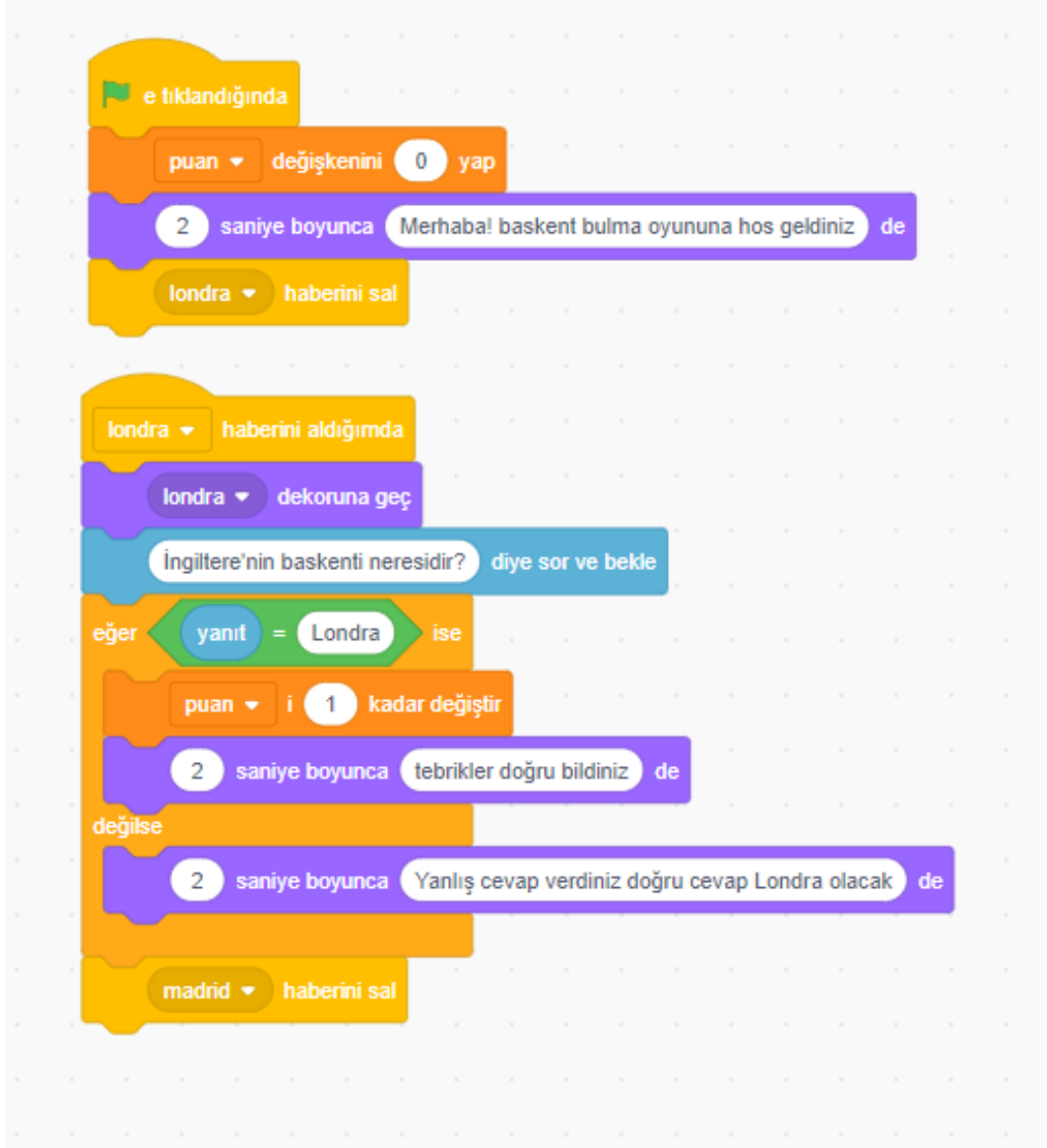
*Kurallar:*

- Ülke ve başkent isimlerinin yazımına dikkat edilecek ve doğru şekilde eşleştirilecektir.
- Ülkenin başkenti sorulunca yanıt kullanıcıdan alınacaktır.
- Yanıt doğru da olsa yanlış da olsa kullanıcıya geri dönüt verilecektir.
- Doğru bilirse puan değişkeni arttırılacaktır.

*Etkinliğin Ekran Görüntüsü ve Kod Örnekleri:*



Şekil 3.10. Başkent bulma oyunu etkinliği, programın ana ekran görüntüsü



Şekil 3.11. Başkent bulma oyunu etkinliği, karaktere yazılan kodların görüntüsü

### 3.4.2.2. Elma Toplama Oyunu Etkinliği

*Etkinliğin Tanımı:* Scratch ortamında kullanıcının yukarıdan düşen elmaları sepete toplayarak belirtilen sürede yeterli puanı alırsa kazanılan oynanan bir oyun tasarlanması.

*Etkinliğin Amacı:* Bu etkinlik ile öğrenci döngü yapısını ve şart yapısını kullanmayı öğrenir. Oyunun en önemli unsurları olan puan ve süre öğelerini sürekli kontrol etmeyi ve buna bağlı olarak şart yapılarını kullanmayı öğrenir. Kullanıcının oyundaki karakteri istediği gibi yönetmesini sağlar. Süre bittiğinde ya da hedeflenen puana

ulaşıldığında oyunun bitirilmesini sağlar. Basit anlamda bir oyun tasarlamayı deneyimlemiş olur.

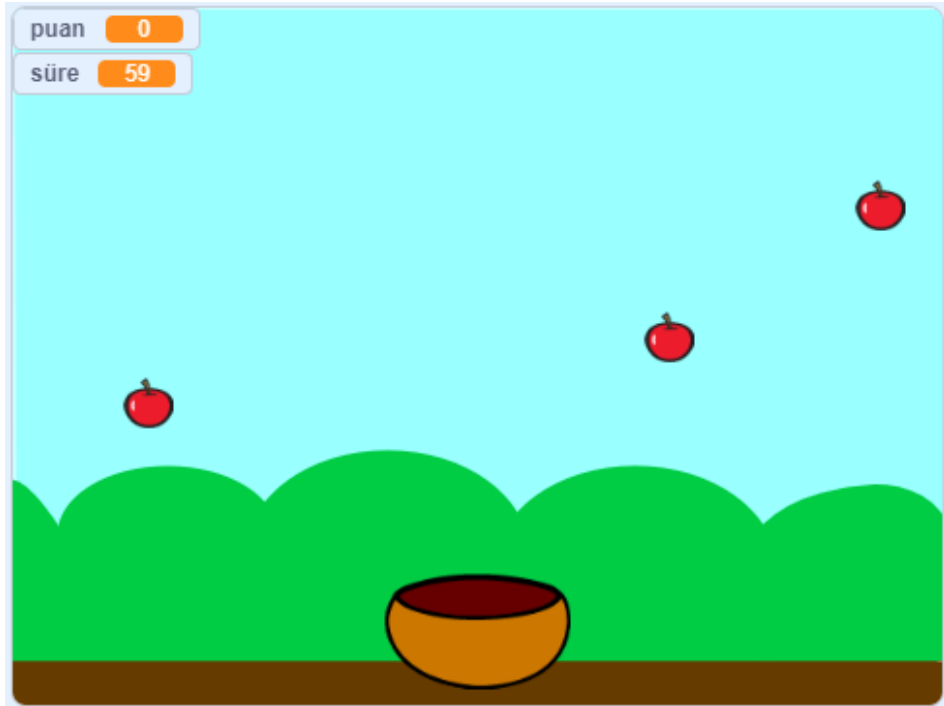
*Ders Süresi:* 80 dk.

*Gerekli Ön Bilgiler:* Scratch ara yüzünü ve komutlarını bilme, değişken kavramını, döngü ve şart yapısını bilme.

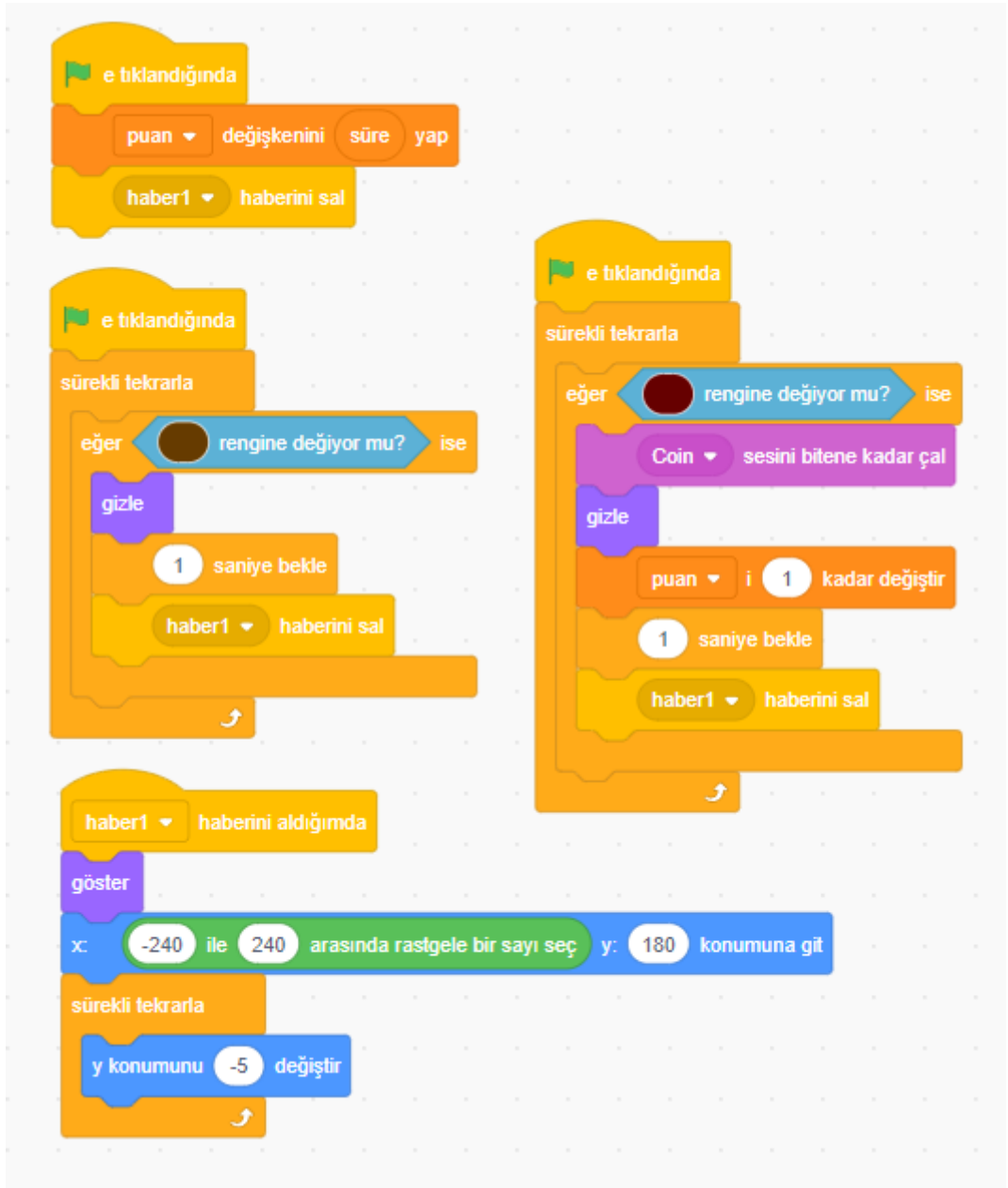
*Kurallar:*

- Sepeti kullanıcı yönetmeli.
- Elmalar rastgele konumlardan ve farklı hızlarda düşmelidir.
- Elmaları sepete aldığımızda puan artmalı ve elma gizlenmelidir.
- Süre tüm karakterlerden ve değişkenlerden bağımsız olarak değişmeli ve kullanıcıya gösterilmelidir.
- Süre bittiğinde ya da istenilen puana ulaşıldığında oyun bitirilip kullanıcıya kazanıp kazanmadığı konusunda bilgi verilmelidir.

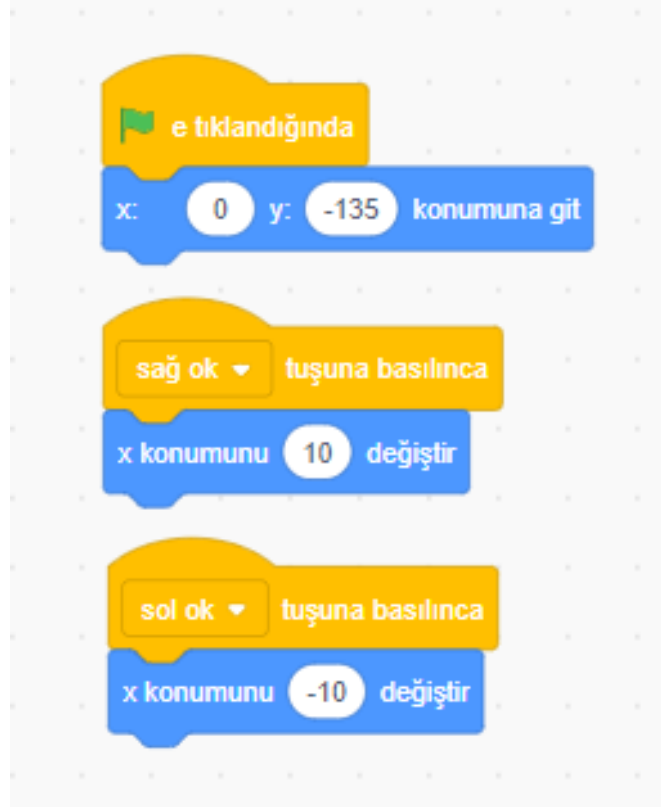
*Etkinliğin Ekran Görüntüsü ve Kod Örnekleri:*



Şekil 3.12. Elma toplama oyunu etkinliği, programın ana ekran görüntüsü



Şekil 3.13. Elma toplama oyunu etkinliği, elma karakterine yazılan kodların görüntüsü



Şekil 3.14. Elma toplama oyunu etkinliği, kase karakterine yazılan kodların görüntüsü

### 3.4.2.3. İngilizce Kelime Oyunu Etkinliği

*Etkinliğin Tanımı:* Scratch ortamında görseli verilen nesnenin İngilizce anlamının sunulan seçenekler arasından seçilmesi şeklinde oynanan bir oyun yazma.

*Etkinliğin Amacı:* Öğrenciye eğlenceli ve eğitici bir oyun geliştirmeyi öğretmeyi amaçlamaktadır. Bu etkinlikte öğrenciler hem İngilizce kelime bilgilerini geliştirebilir hem de eğlenceli ve eğitici bir oyun oynayabilir. Scratch ortamında eş zamanlı işlemleri gerçekleştirmeyi, kullanıcıyı süre ve puan gibi değişkenlerle yönetebilmeyi, işlemleri sıralamayı, şart yapılarını ve değişkenleri kullanmayı öğrenirler.

*Ders Süresi:* 80 dk.

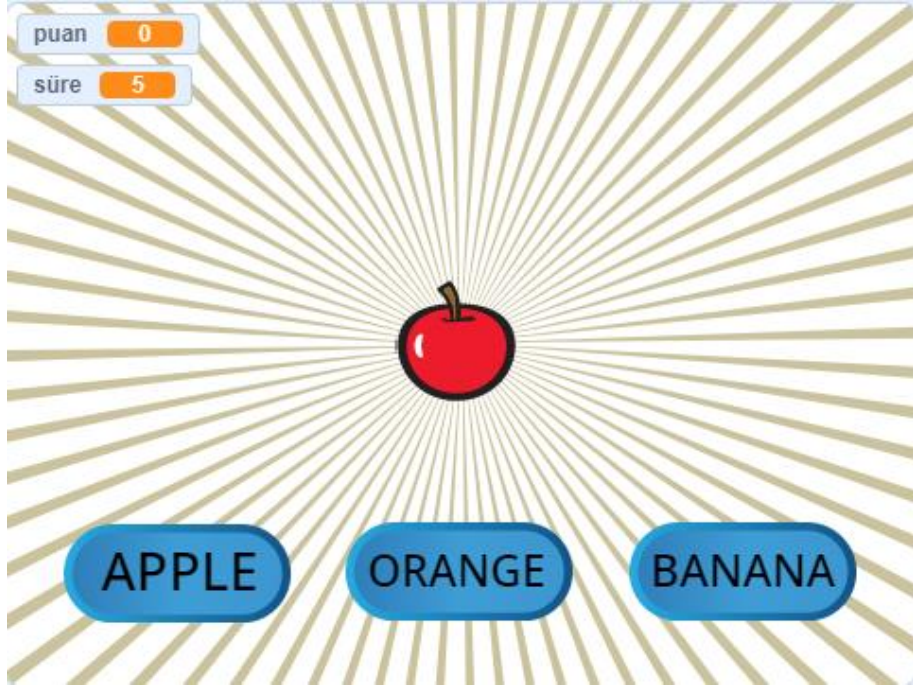
*Gerekli Ön Bilgiler:* Scratch ara yüzünü ve komutlarını bilme, değişken kavramını ve şart yapısını bilme.

*Kurallar:*

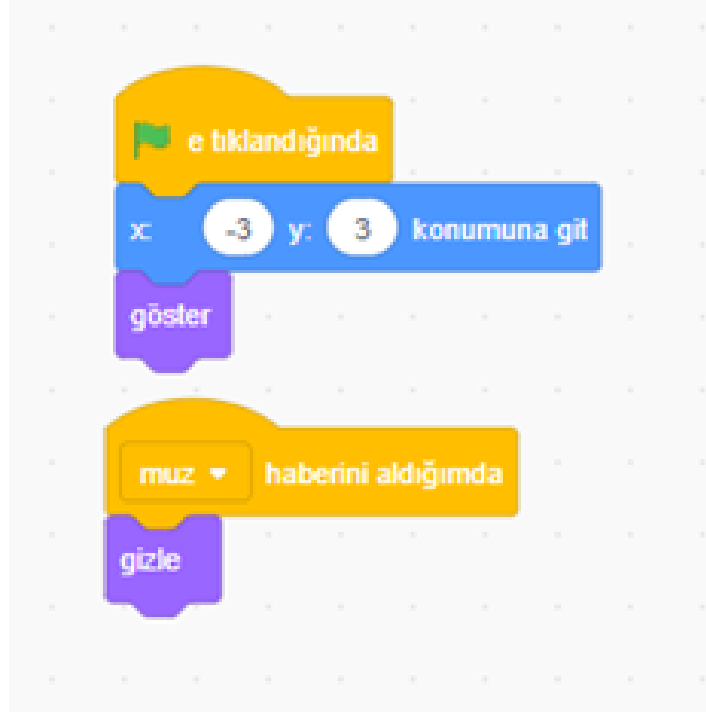
- Ekranı gelecek görseller kısa bir süre ekranda kalmalı kullanıcının hızlı bir şekilde cevaplardan birini seçmesi sağlanmalı.

- Kullanıcı belirlenen sürede gösterilen kelimelerden birini seçtikten sonra ya da seçmese bile diğer görsele geçilmelidir.
- Kullanıcının seçtiği cevaplar doğru ise puanı artmalıdır.
- Baştan belirlenen görsellerin hepsi gösterildikten sonra kullanıcının puanına göre oyunu kazandığı ya da kaybettiği şekilde dönüt verilmelidir.

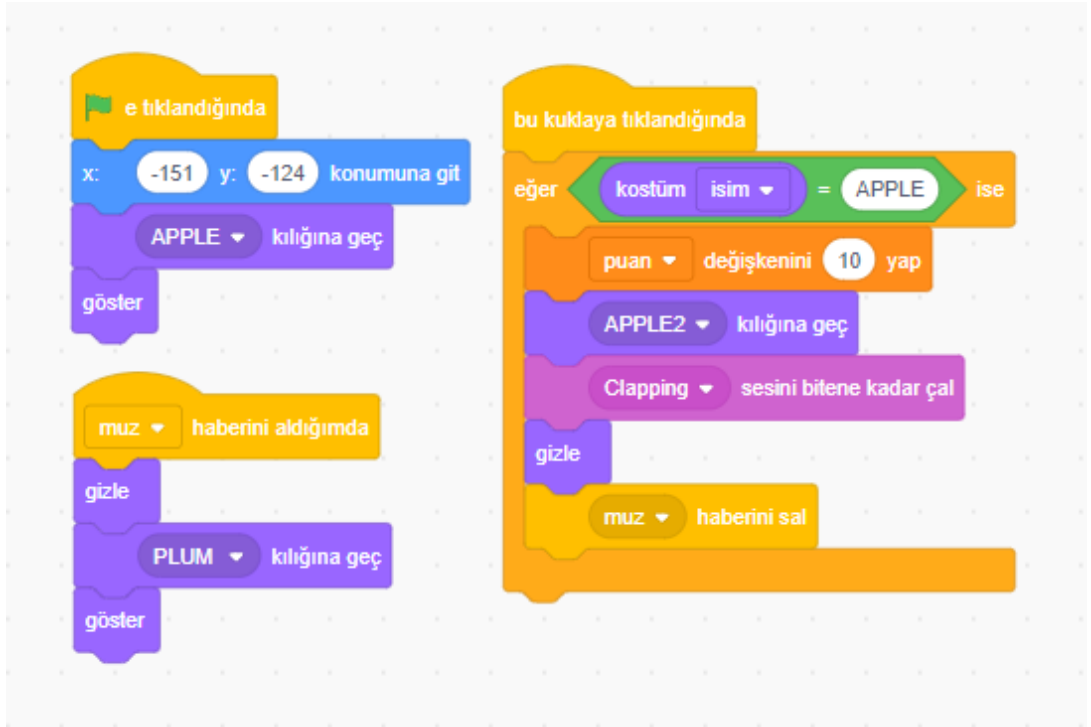
*Etkinliğin Ekran Görüntüsü ve Kod Örnekleri:*



Şekil 3.15. İngilizce kelime oyunu etkinliği, ana ekran görüntüsü



Şekil 3.16. İngilizce kelime oyunu etkinliği, ortada görünen karaktere yazılan kodların görüntüsü



Şekil 3.17. İngilizce kelime oyunu etkinliği, buton karakterine yazılan kodların görüntüsüne bir örnek



### 3.4.2.4. Burcunu Bulma Oyunu Etkinliđi

*Etkinliđin Tanımı:* Öğrencilere günlük hayatta ilgilerini çeken bir konu olan burçlarla ilgili bir etkinlik yaptırmak. Klavyeden girilen gün ve ay bilgilerine göre kullanıcının burcunu hesaplayan oyunu tasarlama.

*Etkinliđin Amacı:* Kullanıcıdan ay ve gün bilgilerini alarak deđişkenlere aktarmak. Bu deđişkenleri şart yapılarında kullanarak elde edilen sonuçlara göre kullanıcının burç bilgisini dođru şekilde ona sunmak.

*Ders Süresi:* 80 dk.

*Gerekli Ön Bilgiler:* Scratch arayüzünü ve komutlarını bilme, deđişken kavramını ve şart yapısını bilme.

*Kurallar:*

- Gün ve ay bilgisi mutlaka kullanıcıdan alınmalı.
- Şart yapıları dođru şekilde oluşturulmalıdır.
- Haber sal komutu ile bazı kodlar harekete geçirilmelidir.
- Kullanıcının doğum tarihine göre ait olduđu burç ekrana yazılmalıdır.

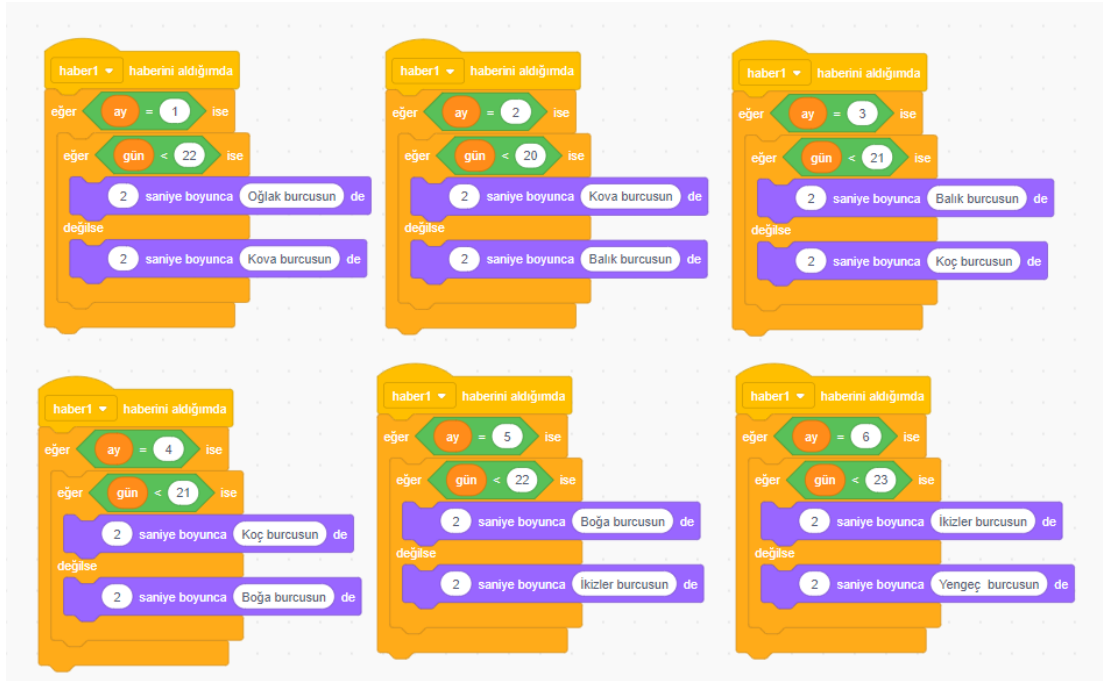
*Etkinliđin Ekran Görüntüsü ve Kod Örnekleri:*



Şekil 3.18. Burcunu bulma oyunu etkinliđi, ana ekran görüntüsü



Şekil 3.19. Burcunu bulma oyunu etkinliği, karaktere yazılan kodların görüntüsü



Şekil 3.20. Burcunu bulma oyunu etkinliği, karaktere yazılan kodların görüntüsü

### 3.5. Verilerin Analizi

Çalışmada kullanılan her bir ölçeğin alt boyutlarında gruplar arasında bir fark olup olmadığını anlamak için ANCOVA'lar uygulanmıştır. Her bir ANCOVA, her bir alt boyut için uygulanmış ve bu alt boyutun analizinde ön test sonuçları kontrol değişkeni olarak yer almıştır. Bu sayede öğrencilerin çalışma öncesindeki ilgili boyuta dair farklılıklarının sonuçlar üzerindeki etkisinin kontrol edilmesi sağlanmıştır.

Her bir boyut için ANCOVA uygulanmadan önce bu analizin gerektirmiş olduğu varsayımlar test edilmiştir. Bu varsayımlar arasında bağımlı ve kontrol değişkeninin sürekli olması, bağımsız değişkenin en az iki boyut içeren kategorik bir değişken olması, grupların birbirinden bağımsız olarak ölçülmesi, uç değerlerin sınırlı olması ve artık değerlerin (residuals) her bir kategori bağlamında normal dağılması yer almaktadır. İlgili durumlara bakıldığında tüm varsayımların sağlandığı görülmüştür.

Ayrıca varyansların homojen dağılımı, kontrol değişkeni ile bağımlı değişken arasındaki doğrusal korelasyon, regresyon eğrilerinin doğrusallığı, sabit varyans durumu (homosdasticity) ve normal dağılımlar da ayrıca kontrol edilen varsayımlardır. Çalışmada elde edilen ve ANCOVA'ya ait varsayım analizler, toplanan verilerle bu analizlerin gerçekleştirilebileceği sonucuna işaret etmiştir.

#### 4. BULGULAR

Bu bölümde araştırma sürecinde elde edilen verilerin nitel analizi sonucunda ortaya çıkan bulgulara yer verilmiştir. Bulgular araştırma sorularına cevap aranarak yorumlanmıştır. İki farklı ölçme aracı kullanıldığından bu araçlara ait alt başlıklarda araştırma sorularına cevap aranmıştır. BİD testinde elde edilen bulgular 7 başlık altında yorumlanmıştır: a) Temel sıralama boyutu b) Belirli sayıda tekrar eden döngüler boyutu c) Şart gerçekleşene kadar çalışan döngüler boyutu d) Basit şartlı ifadeler boyutu e) Karmaşık şartlı ifadeler boyutu f) Şart gerçek oldukça çalışan döngüler boyutu g) Basit fonksiyonlar boyutu. BİDBÖA ölçeğine ait bulgular 5 başlık altında yorumlanmıştır: a) Algoritma tasarlama yeterliği b) Problem çözme yeterliği c) Veri işleme yeterliği d) Temel programlama yeterliği e) Özgüven yeterliği

Çalışmaya katılan öğrencilerin yer aldıkları gruptan bağımsız olarak bilgi işlemsel düşünme becerileri ve öz yeterlilik algıları bakımından gelişim gösterip göstermedikleri gruplar arasındaki olası farkın ortaya konulmasından önce ayrıca test edilmiştir. Buna göre öğrencilerin BİD testinden elde edilen ön test ve son test sonuçları arasında bağımlı örneklem T-testi uygulanmıştır. Analizin sonuçlarına göre öğrencilerin çalışma sonunda BİD testinden aldıkları ön test (M=13,52, SD=4,33) ve son test (M=14,15, SD=5,01) puanları arasında anlamlı düzeyde fark olduğu gözlemlenmiştir ( $t(175)=2.33, p<.05$ ). Bu durum çalışmaya katılan öğrencilerin gurplardan bağımsız olarak bilgi işlemsel düşünme becerilerinin anlamlı düzeyde geliştiğini ifade etmektedir.

Benzer kontrol öğrencilerin BİD öz yeterlilik üzerinde de gerçekleştirilmiştir. Buna göre öğrencilerin BİD öz yeterlilik ölçeğinden aldıkları ön ve son test sonuçları bağımlı örneklem T-testi ile karşılaştırılmıştır. Gerçekleştirilen analizin sonuçlarına göre BİD öz yeterlilik ölçeğinden alınan ön test (M=76,73, SD=13,23) ve son test sonuçları (M=82,23, SD=13,66) arasında anlamlı düzeyde farklılık tespit edilmiştir ( $t(175)=5.72, p<.05$ ). Bu durum çalışmaya katılan öğrencilerin BİD öz yeterlilik algılarında yer aldıkları gruptan bağımsız olarak bir gelişim olduğuna işaret etmektedir.

#### 4.1. BİD Testine Ait Bulgular

Bu bölümde BİD testinin her bir alt boyutuna ait sonuçlar yer almaktadır. Deney ve kontrol grubunun BİD testine verdikleri ön test ve son test yanıtları ANCOVA testi ile analiz edilmiştir. Elde edilen sonuçlara göre BİD becerisinde gelişme olup olmadığı yorumlanmıştır.

##### 4.1.1. Temel Sıralama Boyutu

Tablo 4.1. Temel Sıralama boyutunda ANCOVA sonuçları

Kaynak	Tip III Kareler Toplamı	df	Ortalama Kare	F	Sig.	Kısmi Eta Kare
Düzeltilmiş Model	26.741 <sup>a</sup>	2	13.37	19.05	.00	.18
Sabit	77.466	1	77.47	110.38	.00	.39
Ön Test	25.950	1	25.95	36.97	.00	.18
Grup	2.387	1	2.39	3.40	.07	.02
Hata	121.418	173	.70			
Toplam	1626.000	176				
Düzeltilmiş Toplam	148.159	175				

Tablo 4.1 Temel sıralama boyutunda ANCOVA sonuçlarını özetlemektedir. BİD testi Temel Sıralama boyutu son test puanlarının bağımlı değişken olduğu ve ön test sonuçlarının kontrol değişkeni olarak yer aldığı ANCOVA analizi sonucunda deney ve kontrol grupları arasında anlamlı bir fark bulunmamıştır.

Tablo 4.2. Temel Sıralama ifadesi deęişkenine yönelik deney ve kontrol grubu ortalamaları

Grup	Ortalama	Standart Hata	% 95 Güven Aralığı	
			Alt Sınır	Üst Sınır
Kontrol	3.017 <sup>a</sup>	.09	2.84	3.195
Deney	2.782 <sup>a</sup>	.09	2.67	2.96

Bu tabloya göre deney grubunda (M=2.782) yer alan katılımcılar ön test sonuçları kontrol edildiğinde kontrol grubu öğrencilerinden (M=3.017) daha düşük puanlar elde etmişlerdir(F(1.176)=19.05, p=.07).

#### 4.1.2. Belirli Sayıda Tekrar Eden Döngüler Boyutu

Tablo 4.3. Belirli Sayıda Tekrar Eden Döngüler boyutunda ANCOVA sonuçları

Kaynak	Tip III Kareler Toplamı	df	Ortalama Kare	F	Sig.	Kısmi Eta Kare
Düzeltilmiş Model	61.629 <sup>a</sup>	2	30.81	35.47	.00	.29
Sabit	39.432	1	39.43	45.39	.00	.21
Ön Test	61.164	1	61.16	70.41	.00	.29
Grup	.354	1	.35	.41	.52	.00
Hata	150.280	173	.87			
Toplam	1292.000	176				
Düzeltilmiş Toplam	211.909	175				

Tablo 4.3 Belirli sayıda tekrar eden döngüler boyutunda ANCOVA sonuçlarını özetlemektedir. BİD testi Belirli Sayıda Tekrar Eden Döngüler boyutunda test puanlarının bağımlı değişken olduğu ve ön test sonuçlarının kontrol değişkeni olarak yer aldığı ANCOVA analizi sonucunda deney ve kontrol grupları arasında anlamlı bir fark bulunmamıştır.

Tablo 4.4. Belirli Sayıda Tekrar Eden Döngüler değişkenine yönelik deney ve kontrol grubu ortalamaları

Grup	Ortalama	Standart Hata	% 95 Güven Aralığı	
			Alt Sınır	Üst Sınır
Kontrol	2.432	.10	2.23	2.63
Deney	2.522	.10	2.33	2.72

Bu tabloya göre deney grubunda (M=2.522) yer alan katılımcılar ön test sonuçları kontrol edildiğinde kontrol grubu öğrencilerinden (M=2.432) daha yüksek puanlar elde etmelerine rağmen bu fark anlamlı düzeyde değildir ( $F(1,176)=35.47, p=.52$ ).

### 4.1.3. Şart Gerçekleşene Kadar Çalışan Döngüler Boyutu

Tablo 4.5. Şart Gerçekleşene Kadar Çalışan Döngüler boyutunda ANCOVA sonuçları

Kaynak	Tip III Kareler Toplamı	df	Ortalama		Sig.	Kısmi Eta Kare
			Kare	F		
Düzeltilmiş Model	32.562	2	16.28	16.38	.00	.16
Sabit	47.866	1	47.87	48.15	.00	.22
Ön Test	31.477	1	31.48	31.67	.00	.15
Grup	.466	1	.47	.47	.49	.00
Hata	171.984	173	.99			
Toplam	974.000	176				
Düzeltilmiş Toplam	204.545	175				

Tablo 4.5 Şart gerçekleşene kadar çalışan döngüler boyutunda ANCOVA sonuçlarını özetlemektedir. BİD testi Şart Gerçekleşene Kadar Çalışan Döngüler test puanlarının bağımlı değişken olduğu ve ön test sonuçlarının kontrol değişkeni olarak yer aldığı ANCOVA analizi sonucunda deney ve kontrol grupları arasında anlamlı bir fark tespit edilmemiştir.



Tablo 4.6. Şart Gerçekleşene Kadar Çalışan Döngüler değişkenine yönelik deney ve kontrol grubu ortalamaları

Grup	Ortalama	Standart Hata	% 95 Güven Aralığı	
			Alt Sınır	Üst Sınır
Kontrol	2.039 <sup>a</sup>	.11	1.83	2.25
Deney	2.142 <sup>a</sup>	.11	1.93	2.35

Buna göre deney grubunda (M=2.142) yer alan katılımcılar ön test sonuçları kontrol edildiğinde kontrol grubu öğrencilerinden (M=2.039) daha yüksek puanlar elde etmelerine rağmen bu fark anlamlı düzeyde değildir(F(1.176)=16.38, p=.49).

#### 4.1.4. Basit Şartlı İfadeler Boyutu

Tablo 4.7. Basit Şartlı İfadeler boyutunda ANCOVA sonuçları

Kaynak	Tip III Kareler Toplamı	df	Ortalama Kare	F	Sig.	Kısmi Eta Kare
Düzeltilmiş Model	14.258 <sup>a</sup>	2	7.13	5.80	.00	.06
Sabit	108.784	1	108.78	88.46	.00	.34
Ön Test	14.154	1	14.15	11.51	.00	.06
Grup	.451	1	.45	.37	.55	.00
Hata	212.737	173	1.23			
Toplam	695.000	176				
Düzeltilmiş Toplam	226.994	175				

Tablo 4.7, Basit şartlı ifadeler boyutunda ANCOVA sonuçlarını özetlemektedir. BİD testi Basit Şartlı İfadeler boyutunda son test puanlarının bağımlı değişken olduğu ve ön test sonuçlarının kontrol değişkeni olarak yer aldığı ANCOVA analizi sonucunda deney ve kontrol grupları arasında anlamlı bir fark bulunmamıştır.

Tablo 4.8. Basit Şartlı İfadeler değişkenine yönelik deney ve kontrol grubu ortalamaları

Grup	Ortalama	Standart Hata	% 95 Güven Aralığı	
			Alt Sınır	Üst Sınır
Kontrol	1.682 <sup>a</sup>	.12	1.45	1.92
Deney	1.580 <sup>a</sup>	.12	1.35	1.81

Buna göre deney grubunda (M=1.580) yer alan katılımcılar ön test sonuçları kontrol edildiğinde kontrol grubu öğrencilerinden (M=1.682) daha düşük puanlar elde etmişlerdir( $F(1,176)=5.80, p=.55$ ).

#### 4.1.5. Karmaşık Şartlı İfadeler Boyutu

Tablo 4.9. Karmaşık Şartlı İfadeler boyutunda ANCOVA sonuçları

Kaynak	Tip III Kareler Toplamı	df	Ortalama Kare	F	Sig.	Kısmi Eta Kare
Düzeltilmiş Model	22.761 <sup>a</sup>	2	11.38	9.96	.00	.10
Sabit	83.546	1	83.55	73.10	.00	.30
Ön Test	20.814	1	20.81	18.21	.00	.09
Grup	1.800	1	1.80	1.57	.21	.01
Hata	197.733	173	1.14			
Toplam	763.000	176				
Düzeltilmiş Toplam	220.494	175				

Tablo 4.9, Karmaşık Şartlı İfadeler boyutunda ANCOVA sonuçlarını özetlemektedir. BİD testi Karmaşık Şartlı İfadeler son test puanlarının bağımlı değişken olduğu ve ön test sonuçlarının kontrol değişkeni olarak yer aldığı ANCOVA analizi sonucunda deney ve kontrol grupları arasında anlamlı bir fark bulunmamıştır.

Tablo 4.10. Karmaşık Şartlı İfadeler değişkenine yönelik deney ve kontrol grubu ortalamaları

Grup	Ortalama	Standart Hata	% 95 Güven Aralığı	
			Alt Sınır	Üst Sınır
Kontrol	1.858 <sup>a</sup>	.11	1.63	2.08
Deney	1.656 <sup>a</sup>	.11	1.43	1.88

Buna göre deney grubunda (M=1.656) yer alan katılımcılar ön test sonuçları kontrol edildiğinde kontrol grubu öğrencilerinden (M=1.858) daha düşük puanlar elde etmişlerdir( $F(1,176)=9.96$ ,  $p=.21$ ).

#### 4.1.6. Şart Gerçek Oldukça Çalışan Döngüler Boyutu

Tablo 4.11. Şart Gerçek Oldukça Çalışan Döngüler boyutunda ANCOVA sonuçları

Kaynak	Tip III Kareler Toplamı	df	Ortalama Kare	F	Sig.	Kısmi Eta Kare
Düzeltilmiş Model	11.018 <sup>a</sup>	2	5.51	6.32	.00	.07
Sabit	55.440	1	55.44	63.56	.00	.27
Ön Test	10.563	1	10.56	12.11	.00	.06
Grup	.402	1	.40	.46	.50	.00
Hata	150.891	173	.87			
Toplam	546.000	176				
Düzeltilmiş Toplam	161.909	175				

Tablo 4.11, Şart Gerçek Oldukça Çalışan Döngüler boyutunda ANCOVA sonuçlarını özetlemektedir. BİD testi Şart Gerçek Oldukça Çalışan Döngüler boyutunda son test puanlarının bağımlı değişken olduğu ve ön test sonuçlarının kontrol değişkeni olarak yer aldığı ANCOVA analizi sonucunda deney ve kontrol grupları arasında anlamlı bir fark tespit edilmemiştir.

Tablo 4.12. Şart Gerçek Oldukça Çalışan Döngüler değişkenine yönelik deney ve kontrol grubu ortalamaları

Grup	Ortalama	Standart Hata	% 95 Güven Aralığı	
			Alt Sınır	Üst Sınır
Kontrol	1.526 <sup>a</sup>	.10	1.33	1.72
Deney	1.430 <sup>a</sup>	.10	1.23	1.62

Buna göre deney grubunda (M=1.430) yer alan katılımcılar ön test sonuçları kontrol edildiğinde kontrol grubu öğrencilerinden (M=1,526) daha düşük puanlar elde etmişlerdir(F(1.176)=6.32, p=.50).

#### 4.1.7. Basit Fonksiyonlar Boyutu

Tablo 4.13. Basit Fonksiyonlar boyutunda ANCOVA sonuçları

Kaynak	Tip III Kareler Toplamı	df	Ortalama		Sig.	Kısmi Eta Kare
			Kare	F		
Düzeltilmiş Model	38.211 <sup>a</sup>	2	19.10	13.77	.00	.13
Sabit	53.394	1	53.39	38.49	.00	.18
Ön Test	38.211	1	38.21	27.55	.00	.14
Grup	.002	1	.00	.00	.97	.00
Hata	239.970	173	1.39			
Toplam	860.000	176				
Düzeltilmiş Toplam	278.182	175				

Tablo 4.13, Basit Fonksiyonlar boyutunda ANCOVA sonuçlarını özetlemektedir. BİD testi Basit Fonksiyonlar son test puanlarının bağımlı değişken olduğu ve ön test sonuçlarının kontrol değişkeni olarak yer aldığı ANCOVA analizi sonucunda deney ve kontrol grupları arasında anlamlı bir fark tespit edilmemiştir.

Tablo 4.14. Basit Fonksiyonlar değişkenine yönelik deney ve kontrol grubu ortalamaları

Grup	Ortalama	Standart Hata	% 95 Güven Aralığı	
			Alt Sınır	Üst Sınır
Kontrol	1.821 <sup>a</sup>	.13	1.57	2.07
Deney	1.815 <sup>a</sup>	.12	1.57	2.06

Bu tabloya göre deney grubunda (M=1.815) yer alan katılımcılar ön test sonuçları kontrol edildiğinde kontrol grubu öğrencilerinden (M=1,821) daha düşük puanlar elde etmelerine rağmen bu fark anlamlı düzeyde değildir(F(1.176)=13.77, p=.97).

## 4.2. BİDBÖA Ölçeğine Ait Bulgular

Bu bölümde BİDBÖA ölçeğine ait ANCOVA sonuçları raporlanmıştır.

### 4.2.1. Algoritma Tasarlama Yeterliği

Tablo 4.15. Algoritma Tasarlama Yeterliği ifadesi boyutunda ANCOVA sonuçları

Kaynak	Tip III Kareler Toplamı	df	Ortalama Kare	F	Sig.	Kısmi Eta Kare
Düzeltilmiş Model	729.757 <sup>a</sup>	2	364.88	18.59	.00	.18
Sabit	2049.913	1	2049.91	104.42	.00	.38
Ön Test	687.004	1	687.00	34.99	.00	.17
Grup	13.823	1	13.82	.70	.40	.00
Hata	3396.237	173	19.63			
Toplam	68539.000	176				
Düzeltilmiş Toplam	4125.994	175				

Tablo 4.15, Algoritma Tasarlama Yeterliği ifadesi boyutunda ANCOVA sonuçlarını özetlemektedir. BİD özyeterlik ölçeği Algoritma Tasarlama Yeterliği son test puanlarının bağımlı değişken olduğu ve ön test sonuçlarının kontrol değişkeni olarak yer aldığı ANCOVA analizi sonucunda deney ve kontrol grupları arasında anlamlı bir fark tespit edilmemiştir.



Tablo 4.16. Algoritma Tasarlama Yeterliđi ifadesi deđiřkenine y6nelik deney ve kontrol grubu ortalamaları

Grup	Ortalama	Standart Hata	% 95 G6ven Aralıđı	
			Alt Sınır	6st Sınır
Kontrol	18.846 <sup>a</sup>	.48	17.90	19.79
Deney	19.409 <sup>a</sup>	.47	18.48	20.34

Bu tabloya g6re deney grubunda (M=19.409) yer alan katılımcılar 6n test sonuđları kontrol edildiđinde kontrol grubu 6đrencilerinden (M=18.846) daha y6ksek puanlar elde etmelerine rađmen bu fark anlamlı d6zeyde deđildir (F(1,176)=18.59, p=.40).

#### 4.2.2. Problem 6özme Yeterliđi

Tablo 4.17. Problem 6özme Yeterliđi ifadesi boyutunda ANCOVA sonuđları

Kaynak	Tip III Kareler Toplamı	df	Ortalama Kare	F	Sig.	Kısmi Eta Kare
D6zelti miř Model	573.810 <sup>a</sup>	2	286.90	29.01	.00	.25
Sabit	334.666	1	334.67	33.84	.00	.16
6n Test	560.393	1	560.39	56.67	.00	.25
Grup	17.996	1	18.00	1.82	.18	.01
Hata	1710.826	173	9.89			
Toplam	106172.000	176				
D6zelti miř Toplam	2284.636	175				

Tablo 4.17, Problem Çözme Yeterliği ifadesi boyutunda ANCOVA sonuçlarını özetlemektedir. BİD özyeterlik ölçeği Problem Çözme Yeterliği son test puanlarının bağımlı değişken olduğu ve ön test sonuçlarının kontrol değişkeni olarak yer aldığı ANCOVA analizi sonucunda deney ve kontrol grupları arasında anlamlı bir fark bulunmamıştır.

Tablo 4.18. Problem Çözme Yeterliği ifadesi değişkenine yönelik deney ve kontrol grubu ortalamaları

Grup	Ortalama	Standart Hata	% 95 Güven Aralığı	
			Alt Sınır	Üst Sınır
Kontrol	24.619 <sup>a</sup>	.34	23.95	25.28
Deney	23.979 <sup>a</sup>	.33	23.32	24.64

Bu tabloya göre deney grubunda (M=23.979) yer alan katılımcılar ön test sonuçları kontrol edildiğinde kontrol grubu öğrencilerinden (M=24.619) anlamlı düzeyde daha düşük puanlar elde etmişlerdir( $F(1,176)=29.01, p=.18$ ).

### 4.2.3. Veri İşleme Yeterliği

Tablo 4.19. Veri İşleme Yeterliği ifadesi boyutunda ANCOVA sonuçları

Kaynak	Tip III Kareler Toplamı	df	Ortalama Kare	F	Sig.	Kısmi Eta Kare
Düzeltilmiş Model	260.643 <sup>a</sup>	2	130.32	11.95	.00	.12
Sabit	1611.190	1	1611.19	147.75	.00	.46
Ön Test	247.865	1	247.86	22.73	.00	.12
Grup	12.545	1	12.54	1.15	.28	.01
Hata	1886.578	173	10.90			
Toplam	46027.000	176				
Düzeltilmiş Toplam	2147.222	175				

Tablo 4.19, Veri İşleme Yeterliği ifadesi boyutunda ANCOVA sonuçlarını özetlemektedir. BİD özyeterlik ölçeği Veri İşleme Yeterliği son test puanlarının bağımlı değişken olduğu ve ön test sonuçlarının kontrol değişkeni olarak yer aldığı ANCOVA analizi sonucunda deney ve kontrol grupları arasında anlamlı bir fark tespit edilmemiştir.

Tablo 4.20. Veri İşleme Yeterliği ifadesi değişkenine yönelik deney ve kontrol grubu ortalamaları

Grup	Ortalama	Standart Hata	% 95 Güven Aralığı	
			Alt Sınır	Üst Sınır
Kontrol	15.520 <sup>a</sup>	.35	14.82	16.22
Deney	16.054 <sup>a</sup>	.35	15.36	16.74

Bu tabloya göre deney grubunda (M=16.054) yer alan katılımcılar ön test sonuçları kontrol edildiğinde kontrol grubu öğrencilerinden (M=15.520) daha yüksek puanlar elde etmelerine rağmen bu fark anlamlı düzeyde değildir ( $F(1,176)=11.95, p=.28$ ).

#### 4.2.4. Temel Programlama Yeterliği

Tablo 4.21. Temel Programlama Yeterliği ifadesi boyutunda ANCOVA sonuçları

Kaynak	Tip III Kareler Toplamı	df	Ortalama Kare	F	Sig.	Kısmi Eta Kare
Düzeltilmiş Model	171.025 <sup>a</sup>	2	85.51	14.05	.00	.14
Sabit	817.348	1	817.35	134.28	.00	.44
Ön Test	156.330	1	156.33	25.68	.00	.13
Grup	11.012	1	11.01	1.81	.18	.01
Hata	1053.015	173	6.09			
Toplam	22807.000	176				
Düzeltilmiş Toplam	1224.040	175				

Tablo 4.21, Temel Programlama Yeterliđi ifadesi boyutunda ANCOVA sonuçlarını özetlemektedir. BİD özyeterlik ölçeđi Temel Programlama Yeterliđi son test puanlarının bađımlı deđişken olduđu ve ön test sonuçlarının kontrol deđişkeni olarak yer aldıđı ANCOVA analizi sonucunda deney ve kontrol grupları arasında anlamlı bir fark bulunmamıştır.

Tablo 4.22. Temel Programlama Yeterliđi ifadesi deđişkenine yönelik deney ve kontrol grubu ortalamaları

Grup	Ortalama	Standart Hata	% 95 Güven Aralıđı	
			Alt Sınır	Üst Sınır
Kontrol	10.821 <sup>a</sup>	.26	10.30	11.34
Deney	11.321 <sup>a</sup>	.26	10.80	11.84

Bu tabloya göre deney grubunda (M=11.321) yer alan katılımcılar ön test sonuçları kontrol edildiđinde kontrol grubu öğrencilerinden (M=10.821) daha yüksek puanlar elde etmelerine rağmen bu fark anlamlı düzeyde deđildir( $F(1,176)=14.05, p=.18$ ).

#### 4.2.5. Özgüven Yeterliği

Tablo 4.23. Özgüven Yeterliği ifadesi boyutunda ANCOVA sonuçları

Kaynak	Tip III Kareler Toplamı	df	Ortalama Kare	F	Sig.	Kısmi Eta Kare
Düzeltilmiş Model	99.487 <sup>a</sup>	2	49.74	11.51	.00	.12
Sabit	487.262	1	487.26	112.70	.00	.39
Ön Test	92.962	1	92.96	21.50	.00	.11
Grup	4.732	1	4.73	1.09	.30	.01
Hata	747.945	173	4.32			
Toplam	25952.000	176				
Düzeltilmiş Toplam	847.432	175				

Tablo 4.23, Özgüven Yeterliği ifadesi boyutunda ANCOVA sonuçlarını özetlemektedir. BİD özyeterlik ölçeği Özgüven Yeterliği son test puanlarının bağımlı değişken olduğu ve ön test sonuçlarının kontrol değişkeni olarak yer aldığı ANCOVA analizi sonucunda deney ve kontrol grupları arasında anlamlı bir fark tespit edilmemiştir.

Tablo 4.24. Özgüven Yeterliği ifadesi değişkenine yönelik deney ve kontrol grubu ortalamaları

Grup	Ortalama	Standart Hata	% 95 Güven Aralığı	
			Alt Sınır	Üst Sınır
Kontrol	12.109 <sup>a</sup>	.22	11.67	12.55
Deney	11.781 <sup>a</sup>	.22	11.35	12.22

Bu tabloya göre deney grubunda ( $M=11.781$ ) yer alan katılımcılar ön test sonuçları kontrol edildiğinde kontrol grubu öğrencilerinden ( $M=12.109$ ) daha düşük puanlar elde etmişlerdir( $F(1.176)=11.51, p=.30$ ).

## 5. TARTIŞMA, SONUÇ VE ÖNERİLER

Bu bölümde, araştırmanın bulgularının tartışılması, yorumlanması, sonuçları ve bunlara dayalı olarak önerilere yer verilmiştir.

Bu çalışmada, ortaokul düzeyinde matematik destekli temel programlama eğitiminin öğrencilerin BİD becerileri ve öz yeterliklerini geliştirmedeki etkisi incelenmiştir.

BİD becerisi kazandırma konusunda bilişim teknolojileri ve matematik derslerinin rolünün anlaşılmasına yönelik yapılan araştırmalar olduğu görülmektedir (Cui ve Ng, 2021; Ng ve Cui, 2020; Sung ve Black, 2020). Bu çalışmada söz konusu iki derse ait kazanımların birlikte sunulduğu etkinliklerin BİD becerisini geliştirmedeki rolünün araştırılmasına odaklanılmıştır. Bu doğrultuda, ortaokul 6.sınıf öğrencileri ile tasarlanmış etkinlikler uygulanmıştır. Deney grubu öğrencileri birinci dönem matematik dersinin öğretim programında yer alan konularla (işlem önceliği, üslü sayılar, bölünebilme kuralları ve asal sayılar) desteklenmiş etkinlikleri Scratch programında uygularken; kontrol grubu öğrencileri Scratch programında bilişim teknolojileri ve yazılım öğretim programına göre oyun tasarlama etkinlikleri gerçekleştirmişlerdir. Ön test- son test kontrol gruplu yarı deneysel desenin kullanıldığı çalışmada öğrencilerin BİD testi ve BİD özyeterlik ölçeğinden elde edilen puan ortalamaları karşılaştırılmıştır.

### 5.1. Bilgi İşlemsel Düşünme Becerisi

Bu bölümde bilgi işlemsel düşünmeye yönelik gerçekleştirilen beceri testine dair elde edilen bulgular tartışılmıştır. BİD testi kapsamında, temel sıralama, belirli sayıda tekrar eden döngüler, şart gerçekleşene kadar çalışan döngüler, basit şartlı ifadeler, karmaşık şartlı ifadeler, şart gerçek oldukça çalışan döngüler, basit fonksiyonlar olmak üzere yedi alt bölüm bulunmaktadır. Deneysel işlemin sonucunda, ön test puanları kontrol edildiğinde deney ve kontrol grubu öğrencileri arasında herhangi bir boyutta istatistiksel olarak anlamlı bir farklılık olmadığı ortaya çıkmıştır.

*Temel sıralama* boyutunda yer alan sorular giriş seviyesindeki kodlama (ileri, sağa dön, sola dön) işlemlerini içeren sorulardan oluşmaktadır. 6.sınıf öğrencileri için kolay bir seviye olduğundan her iki grubun birbirine yakın seviyelerde puan elde ettiği düşünülebilir.



Her iki grupta yer alan öğrencilerin 5.sınıf Bilişim Teknolojileri ve Yazılım derslerinde code.org ve kodlabüyü ortamlarında bu türde kodlama etkinlikleri gerçekleştirdikleri ve bu konuda deneyimli oldukları göz önünde bulundurulmalıdır. Nitekim Çetin vd. (2020) sınıf düzeyi ilerledikçe buna paralel olarak BİD testi puanlarının da arttığını göstermişlerdir. Dolayısıyla bilişim teknolojilerine yönelik derslerin alınması, öğrencilerin doğal bilişsel gelişim süreçleri ve farklı dersler içerisindeki bilgi işlemsel düşünmeye katkı sağlayabilecek içeriklerin var olması bilgi işlemsel düşünmenin gelişimine katkı sunmuş olabilir. Bu durum çalışmada ulaşılan birçok noktada olduğu gibi temel sıralama boyutunda da deney ve kontrol grubu arasında anlamlı bir fark ortaya çıkmamasına neden olmuş olabilir.

*Belirli sayıda tekrar eden döngüler* bölümünde, ön test puanları kontrol edildiğinde deney ve kontrol grubu öğrencileri arasında anlamlı bir farklılık bulunmamıştır. Bununla birlikte, deney grubunda yer alan öğrenciler, kontrol grubundakilerden anlamlı düzeyde olmasa da daha yüksek puanlar elde etmişlerdir. Bu boyutta yer alan sorular belirli bir işlem ya da işlemlerin belirli sayıda tekrar etmesini gerektiren sorulardan oluşmaktadır. Bu sorular, öğrencinin tekrar eden örüntüleri fark ederek tekrar etme sayısını bulmasını, varsa yanlışları tespit etmesini ve buna göre kodları sıralamasını gerektirir. Buna göre, matematik destekli etkinliklerin deney grubundaki öğrencilerin bu alandaki performansını olumlu yönde etkilediği düşünülebilir. Deney grubunda yer alan etkinliklerde işlem önceliğine göre işlemlerin sıralanmasına yönelik etkinliklerin olması öğrencilerin bu boyutta daha yüksek puan almalarını desteklemiş olabilir. Temel sıralama boyutunda olduğu gibi bu boyutta da öğrencinin doğal gelişiminin bilgi işlemsel düşünme becerisi üzerinde etkili olduğu düşünüldüğünde bu etkilerden arındırılmış bir ortamda döngü boyutunda elde edilen ancak anlamlı olmayan farkın daha yüksek olma olasılığı ortaya çıkmaktadır. Bu nedenle örüntü tanıma gibi konuların bilgi işlemsel düşünme becerisine bilgisayar bilimleri içerisinde kullanıldığında anlamlı düzeyde katkı sağlayacağı anlaşılabilir. Nitekim BİD becerisi kapsamında öğrencilerin algoritma adımlarının ne zaman tekrarlandığını tanımaları gerekirken; aynı zamanda matematikte de daha geniş bir amacı gerçekleştirmek için bir birim eklemek veya bir uzunluk birimi yerleştirmek gibi temel bir adımı tekrarlamak yaygındır. Bu iki disiplin arasındaki ilişkinin sinerjik olduğu düşünülmektedir (Rich, Spaepen, Strickland ve Moran, 2020).

*Şart gerçekleşene kadar çalışan döngüler* boyutu, bir şart gerçekleşene kadar tekrar eden işlemleri doğru sıralamayı gerektiren sorulardan oluşmaktadır. Bu boyuttaki ön test puanları kontrol edildiğinde deney grubu öğrencileri kontrol grubu öğrencilerinden daha yüksek puan elde etmiş olmalarına rağmen bu farkın anlamlı düzeyde olmadığı tespit edilmiştir. Deney grubundaki öğrenciler örneğin asal sayılar ile ilgili etkinlikte bir takım şartları gözeterak sayının asal olup olmadığını şart gerçekleşene kadar döngü kavramını kullanarak bulmaktadır. Bu bağlamda, bu tür etkinliklerin öğrencilerin bu alandaki gelişmelerini desteklediği düşünülebilir. Nitekim matematik dersi bağlamındaki etkinliklerin somut olması, öğrencilerin günlük hayatta karşılaştığı kavramları içermesi ve sıklıkla kullanılması, programlama eğitimine aktarılması açısından bir potansiyeli olduğu söylenebilir. Örneğin, büyük sayıdan küçük sayı çıkartılırken geriye doğru tekrarlayan bir sayma işlemi ve durma noktası söz konusudur (Cui ve Ng, 2021).

*Basit şartlı ve karmaşık şartlı ifadeler boyutunda*, ön test sonuçları kontrol edildiğinde deney grubu öğrencileri ile kontrol grubu öğrencileri arasında anlamlı bir farklılık olmadığı bulunmuştur. Bununla birlikte, kontrol grubunda yer alan öğrenciler, deney grubundakilerden anlamlı düzeyde olmasa da daha yüksek puanlar elde etmişlerdir. Basit şartlı ifadeler boyutunda yer alan sorularda basit şart ifadeleri sorulmuş ancak bu ifadeler şart gerçekleşene kadar tekrar eden döngülerin içinde sunulmuştur. Karmaşık şartlı ifadeler boyutundaki sorularda bir önceki boyuttaki sorular gibi yine şart gerçekleşene kadar tekrar eden döngüler içinde verilmiştir. Ancak şart yapısı “Eğer ise..... Değilse..... “ şeklinde birden fazla seçenek ile sunulmuştur. Kontrol grubundaki öğrencilerin uyguladığı etkinliklerin daha çok şartlı ifade içermesi bu gruptaki öğrencilerin bu konuda daha fazla desteklenmesine neden olmuş olabilir. Nitekim, Cui ve Ng (2021), öğrencilerin BİD tabanlı ortamı öğrenmek ve aynı zamanda bu ortamda matematiksel kavramları ve problem çözme uygulamalarını içeren süreçte karşılaştıkları güçlüklerin birleştiğini bulmuştur. Buna göre, matematik destekli programlama eğitimi sürecinde deney grubundaki öğrencilerin matematik ve BİD ile ilgili kavramları birlikte öğrenirken daha fazla çaba sarf etmeleri, matematik ya da BİD’i önceliklendirme açısından çeşitli yolları tercih etmelerine neden olabilir. Bu bağlamda, deney grubundaki öğrencilerin matematik ile ilgili kavramlara öncelik vermesi söz konusu olmuş olabilir. Bunun yanında 6.sınıfa kadar öğrencilerin almış olduğu eğitim sürecinde temel matematik konularının yer alması ve bu konular içerisinde basit veya karmaşık düzeyde şartlı

ifadeler içeren konuların sınırlı olması, olası farkın deney grubu lehine çıkmasını sınırlamış olabilir. Oluşturulan matematik destekli etkinliklerde bu nedenle şartlı ifadelerin kullanımını sınırlı kalmış olabilir. Yukarıda ifade edildiği gibi kontrol grubunda oyun tasarlama temelli etkinliklerin olması ve oyunların doğası gereği basitten karmaşığa birçok şartlı yapı ihtiyacı olması, ilgili sonucun anlamlı olmasa da kontrol grubu lehine çıkmasına neden olmuş olabilir. Bahsedilen bu durum *Şart gerçek oldukça çalışan döngüler* boyutunda da karşımıza çıkmıştır. Ön test sonuçları kontrol edildiğinde bu boyutta deney grubu öğrencileri kontrol grubu öğrencilerinden anlamlı düzeyde olmayacak şekilde daha düşük bir puan elde etmişlerdir. Sorular şart gerçek olduğunda başlayan işlemleri doğru sıralamayı gerektirmektedir. Yukarıda ifade edilen durumun yanı sıra bu tipteki döngülerin deney grubunun etkinliklerinde yer almaması öğrencilerin bu tip sorularda fikir yürütememesine sebep olmuş olabilir. Buna karşın kontrol grubunun etkinliklerinin daha karmaşık olması ve daha çok döngü içermesi öğrencilerin bu tip sorularda fikir yürütmelerini kolaylaştırmış olabilir.

*Basit fonksiyonlar* boyutundaki bulgular incelendiğinde deney ve kontrol grubu öğrencilerinin birbirine çok yakın puanlar elde ettikleri görülmüştür. Basit fonksiyon oluşturma ya da kullanma işlemi her iki grubun etkinliklerinde yer almamaktadır. Dolayısıyla, deneysel sürecin basit fonksiyonlar boyutunda bir değişikliğe neden olmaması beklenen bir durum olarak düşünülebilir. Ancak öğrenciler önceki yıl bilişim teknolojileri dersinde code.org ve kodlabüyü ortamlarında basit fonksiyon içeren etkinlikler uygulamışlardır. Fonksiyonlar açısından her iki grubunda önceki öğrenme deneyimlerinin birbirine yakın olması deneysel süreç sonucunda farklılığın çıkmamasını açıklayabilir.

## **5.2. Bilgi İşlemsel Düşünme Becerisi Özyeterlik Algısı**

*Algoritma Tasarlama Yeterliği* bölümünde algoritmanın ne olduğu, basit ve koşullu algoritmalar oluşturma, algoritmanın çıktısını tahmin etme ve hata tespit edip ayıklama ile ilgili maddeler yer almaktadır. Deney grubundaki öğrencilerin bu konularda kendileri ile ilgili görüşlerinin kontrol grubundaki öğrencilerin görüşlerine göre anlamlı olmayan düzeyde daha yüksek olduğu görülmüştür. Matematik problemlerini çözerken öğrencilerin doğal olarak algoritma tasarlama aşamalarını gerçekleştiriyor olmaları, deney grubundaki öğrencilerin algoritma konusunda kendilerini daha yeterli görmelerine neden olmuş olabilir. Nitekim Lockwood, Asay,

DeJarnette, Thomas (2016), algoritmik düşünmeyi mevcut araçları kullanarak karmaşık bir hedefi bir dizi (sıralı) adıma bölmek için kullanılan mantıklı, organize bir düşünme biçimi olarak tanımlamaktadır. Matematik destekli programlama etkinliklerinin öğrencilerin algoritmik düşünme becerilerini desteklemede daha somut örnekler içerdiği ileri sürülebilir.

*Problem Çözme Yeterliği* boyutundaki ön test sonuçları incelendiğinde deney grubunda yer alan öğrenciler kontrol grubu öğrencilerine göre anlamlı olmayan düzeyde daha düşük puanlar elde etmişlerdir. Bu boyutta verilen maddelere bakıldığında problem çözme becerileri ile ilgili konuların yer aldığı görülmektedir. Matematik dersi öğrencilerin en çok zorlandıkları derslerin başında gelir. Öğrencilerin matematiği zor olarak algılamaları problem çözme becerisindeki algılarını da düşürmüş olabilir. Matematik destekli etkinliklerle öğretilen temel programlama eğitiminde öğrencilerin bilişsel yükü artmış olabilir ve bu durum öğrencilerin problem çözme yeterliğindeki algılarını düşürmüş olabilir. Benzer şekilde, Psycharis ve Kallia (2017), matematik ile birlikte programlama eğitiminin öğrencilerin problem çözme becerisi üzerinde etkili olmadığını bulmuştur. Bu durumun nedenlerini açıklığa kavuşturmak için daha fazla araştırmaya gereksinim olduğu söylenebilir.

*Veri İşleme Yeterliği* boyutundaki ön test sonuçları incelendiğinde deney grubu öğrencileri kontrol grubundaki öğrencilere göre anlamlı olmayan düzeyde daha yüksek puanlar elde etmiştir. Bu bölümde öğrencilere veri ve veri türleri hakkında maddeler sorulmuştur. Scratch programında değişken tanımlarken değişkenlerin türünün belirlenmesine ihtiyaç duyulmaması bu konuda öğrencilerin bilgileri basit düzeyde kalmasına neden olmuş olabilir. Öte yandan deney grubundaki katılımcıların matematik ile ilgili etkinliklerde etkinliğin doğası gereği veri yapıları ile daha fazla etkileşimde olmaları ilgili farkın oluşmasında bir etken olarak öne çıkmış olabilir. Kontrol grubundaki etkinliklerde de değişkenler kullanılmıştır ancak işlemlerin karmaşıklığı ve zorluğu veri kavramı konusunda kendilerini yetersiz hissetmelerine sebep olmuş olabilir.

*Temel Programlama Yeterliği bölümündeki* sorular değişken, şart yapıları, döngüler, aritmetik operatörler ile ilgili maddeler barındırmaktadır. Deney grubu öğrencileri kontrol grubu öğrencilerine göre anlamlı düzeyde olmasa da daha yüksek puan elde etmişlerdir. Bunun sebebi, deney grubunda yer alan matematik destekli

etkinliklerde temel programlama becerilerinin yanı sıra aritmetik operatörlerin yer alması öğrencilerin kendilerine olan güvenlerini artırmış olabilir.

*Özgüven Yeterliği* boyutundaki ön test sonuçları incelendiğinde deney grubunda yer alan öğrencilerin kontrol grubunda yer alan öğrencilerden anlamlı olmayan düzeyde düşük puan elde ettikleri görülmektedir. Bu bölümdeki maddelere bakıldığında öğrencilerin programlama becerileri konusunda kendilerine olan inançlarını değerlendirdikleri maddeler olduğu görülmektedir. Kontrol grubu öğrencilerinin programlama konusunda kendilerine olan inançlarının daha yüksek olmasının sebebi daha karmaşık etkinliklere maruz kalmaları ve oyun tasarlarken soyutlama, ayırıştırma, algoritmik düşünme ve problem çözme becerileri boyutlarında daha çok gelişme görülmesi olabilir.

Her ne kadar yukarda öz yeterlik boyutunda anlamlı olmayan farklar ve gerekçelerinden bahsedilmiş olsa da bu farkların çok küçük olduğu ve bahsedilen gerekçeler dışında etkinliğin gerçekleştiği ortam, öğrencilerle ilgili farklı değişkenler, diğer derslerde öğrenilen bilgiler gibi birçok durum küçük farkların oluşmasına neden olmuş olabilir. Bunların dışında oluşan farklar rastgele ortaya çıkmış ve tekrarlanan ölçümlerde farklı yöne evrilme potansiyeli içeriyor olabilir. Bu nedenle yukarıda bahsedilen durumlarda ilgili sonuçlara daha ihtiyatlı yaklaşılması önerilmektedir.

### **5.3. Öneriler**

Bu bölümde araştırma sonucunda elde edilen bulgulardan yola çıkılarak bazı önerilerde bulunulmuştur.

- Matematik destekli programlama etkinliklerinin öğrencilerin döngüler konusunda destekleme potansiyeli olduğu sonucuna ulaşılmıştır. Bu bağlamda, bu konuyla ilgili matematik destekli programlama etkinliklerinin öğrenme ve öğretme süreçlerinde yer verilmesi önerilmektedir.
- Matematik destekli programlama eğitimine yönelik öğrenme süreçlerinin daha uzun süreli planlanması önerilmektedir.
- Matematik destekli programlama etkinliklerini içeren öğrenme ve öğretme süreçlerinin öğrencilerin matematik dersindeki başarı düzeylerine etkisi gelecekteki araştırmalarda daha detaylı incelenmesi önerilmektedir.
- Matematik destekli programlama etkinliklerini içeren öğrenme ve öğretme süreçlerinin öğrencilerin matematik dersindeki başarı düzeylerine etkisi

gelecekteki arařtırmalarda farklı yař gruplarında incelenmesi önerilmektedir.

- Bu tür etkinliklerde öğrencilerin karşılařtıkları güçlükleri ve öğrenme deneyimlerindeki algıları nitel yöntemlerle derinlemesine incelenmesi önerilmektedir.
- Bu arařtırmada programlama eğitimindeki fonksiyonlar konusu ile ilgili etkinliklerin olmaması bir sınırlılık olarak görülebilir. Öte yandan, bu konunun matematik ve programlama eğitiminin bütünleştirilmesi açısından uygun olduđu söylenebilir. Dolayısıyla, fonksiyonlar konusunda matematik destekli programlama etkinliklerinin tasarlanması, uygulanması ve öğrencilere olan yansımalarının incelenmesi önerilmektedir.
- Matematik destekli programlama etkinliklerinin öğrencilerin problem çözme, özgüven yeterliğindeki rolünü aydınlatmaya yönelik nitel çalışmaların yapılması önerilmektedir.

## KAYNAKÇA

2023 Eğitim Vizyonu, MEB (2018)

Adsay, C., Korkmaz, Ö., Çakır, R., & Erdoğan, F. U. (2020). Ortaokul öğrencilerinin blok temelli kodlama eğitimine dönük öz-yeterlik algı düzeyleri, STEM ve bilgisayarca düşünme beceri düzeyleri. *Eğitim Teknolojisi Kuram ve Uygulama*, 10(2), 469-489.

Aho, A. V. (2012). Computation and computational thinking. *The computer journal*, 55(7), 832-835.

Oluk, A., Korkmaz, Ö., & Oluk, H. A. (2018). Scratch'ın 5. sınıf öğrencilerinin algoritma geliştirme ve BİD becerilerine etkisi. *Türk Bilgisayar ve Matematik Eğitimi Dergisi*, 9(1), 54-71.

Angeli, C., Voogt, J., Fluck, A., Webb, M., Cox, M., Malyn-Smith, J., & Zagami, J. (2016). A K-6 computational thinking curriculum framework: Implications for teacher knowledge. *Journal of Educational Technology & Society*, 19(3), 47-57.

Barcelos, T. S., Muñoz-Soto, R., Villarroel, R., Merino, E., & Silveira, I. F. (2018). Mathematics Learning through Computational Thinking Activities: A Systematic Literature Review. *J. Univers. Comput. Sci.*, 24(7), 815-845.

Bilgi İşlemsel Düşünme Becerisinin Disiplinlerarası Yaklaşım İle Öğretimi, MEB (2020)

Bilici, O., & Güler, Ç. (2021). Ortaokul Öğrencilerinin Bilgi İşlemsel Düşünme Becerileri ile Akademik Öz Yeterlikleri Arasındaki İlişkinin İncelenmesi. *Çağ Üniversitesi Sosyal Bilimler Dergisi*, 18(1), 107-119.

Bilişim Teknolojileri Ve Yazılım Dersi Öğretim Programı, MEB (2018)

Brennan, K., & Resnick, M. (2012, April). New frameworks for studying and assessing the development of computational thinking. In *Proceedings of the 2012 annual meeting of the American educational research association, Vancouver, Canada* (Vol. 1, p. 25).

Cetin, I. (2015). Students' understanding of loops and nested loops in computer programming: An APOS theory perspective. *Canadian Journal of Science, Mathematics and Technology Education*, 15(2), 155-170.

Cui, Z., & Ng, O. L. (2021). The interplay between mathematical and computational thinking in primary school students' mathematical problem-solving within a programming environment. *Journal of Educational Computing Research*, 59(5), 988-1012.

Çetin, İ., Otu, T., & Oktaç, A. (2020). Adaption of the Computational Thinking Test into Turkish. *Turkish Journal of Computer and Mathematics Education (Turcomat)*, 11(2), 343-360.

Çubukluöz, Ö. (2019). *6. sınıf öğrencilerinin matematik dersindeki öğrenme zorluklarının scratch programıyla tasarlanan matematiksel oyunlarla giderilmesi: bir eylem araştırması* (Master's thesis, Bartın Üniversitesi, Eğitim Bilimleri Enstitüsü).

Demir, Ö., & Seferoğlu, S. S. (2017). Yeni kavramlar, farklı kullanımlar: BİDyle ilgili bir değerlendirme. *Eğitim teknolojileri okumaları*, 41, 801-830.

Dolmacı, A., & Akhan, N. E. (2020). Bilişimsel Düşünme Becerileri Ölçeğinin Geliştirilmesi: Geçerlik ve Güvenirlik Çalışması. *Itobiad: Journal of the Human & Social Science Researches*, 9(3).

Elçiçek, M. Türkiye'de Bilgi İşlemsel Düşünme Becerisi ile İlgili Yapılmış Lisansüstü Tezlerin Tematik ve Yöntemsel Eğilimleri.

- Grover, S., & Pea, R. (2013). Computational thinking in K–12: A review of the state of the field. *Educational researcher*, 42(1), 38-43.
- Gulbahar, Yasemin & Kalelioglu, Filiz & Doğan, Dilek & Karataş, Erinç. (2020). Bilge Kunduz: Enformatik ve Bilgi-İşlemsel Düşünmeyi Kavram Temelli Öğrenme için Toplumsal Bir Yaklaşım. Ankara Üniversitesi Eğitim Bilimleri Fakültesi Dergisi. 10.30964/auebfd.560771.
- Gülbahar, Y. & Kalelioğlu, F. (2015). Competencies for e-Instructors: How to Qualify and Guarantee Sustainability. *Contemporary Educational Technology*, 6 (2). 140-154.
- Gülbahar, Y. (2017). Bilgi işlemsel düşünmeden programlamaya. *Pegem Akademi, Ankara*, 318-336.
- Gülbahar, Y., & Kalelioğlu, F. (2018). Bilişim Teknolojileri Ve Bilgisayar Bilimi: Öğretim Programı Güncelleme Süreci. *Millî Eğitim Dergisi*, 47(217), 5-23.
- Gülbahar, Y., Kert, S. B., & Kalelioğlu, F. (2019). Bilgi işlemsel düşünme becerisine yönelik öz yeterlik algısı ölçeği: Geçerlik ve güvenirlik çalışması. *Türk Bilgisayar ve Matematik Eğitimi Dergisi*, 10(1), 1-29.
- Gülbahar, Y. (Ed.). (2020). *Bilgi işlemsel düşünmeden programlamaya*. Pegem Yayıncılık
- Haseski, H. İ., Ilic, U., & Tugtekin, U. (2018). Defining a New 21st Century Skill-Computational Thinking: Concepts and Trends. *International Education Studies*, 11(4), 29-42.
- Hsu, T. C., Chang, S. C., & Hung, Y. T. (2018). How to learn and how to teach computational thinking: Suggestions based on a review of the literature. *Computers & Education*, 126, 296-310.
- International Society for Technology in Education. (2007). *National educational technology standards for students*. ISTE (Interntl Soc Tech Educ).
- Israel, M., & Lash, T. (2020). From classroom lessons to exploratory learning progressions: Mathematics+ computational thinking. *Interactive learning environments*, 28(3), 362-382.
- ISTE & CSTA (2011). Operational definition of computational thinking for K–12 education. Retrieved from <https://cdn.iste.org/www-root/Computational Thinking Operational Definition ISTE.pdf>
- International Society for Technology in Education [ISTE]. (2016). Computational thinking competencies. Retrieved from <https://www.iste.org/standards/computational-thinking>
- İbili, E., Günbatar, M. S., & Sırakaya, M. (2020). BİD becerilerinin incelenmesi: Meslek liseleri örnekleme. *Kastamonu Eğitim Dergisi*, 28(2), 1067-1078.
- Kale, U., Akcaoglu, M., Cullen, T., Goh, D., Devine, L., Calvert, N., & Grise, K. (2018). Computational what? Relating computational thinking to teaching. *TechTrends*, 62(6), 574-584.
- Karataş, H. 21. Yy. Becerilerinden Robotik Ve Kodlama Eğitiminin Türkiye Ve Dünyadaki Yeri. *21. Yüzyılda Eğitim Ve Toplum Eğitim Bilimleri Ve Sosyal Araştırmalar Dergisi*, 10(30), 693-729.
- Kert, Serhat & Yeni, Sabiha & Şahiner, Abdulkadir. (2017). Investigation Of The Sub-Skills Associated With The Computational Thinking.
- Korkmaz, Ö. , Çakır, R. , Özden, M. Y. , Oluk, A. & Sarıoğlu, S. (2016). Bireylerin Bilgisayarca Düşünme Becerilerinin Farklı Değişkenler Açısından İncelenmesi. *Ondokuz Mayıs University Journal of Education Faculty*, 34 (2), 68-87. Retrieved from <https://dergipark.org.tr/pub/omuefd/issue/20284/215276>



- Korkmaz, Ö., Çakır, R., & Özden, M. Y. (2015). Bilgisayarca düşünme beceri düzeyleri ölçeğinin (bdbl) ortaokul düzeyine uyarlanması. *Gazi Eğitim Bilimleri Dergisi*, 1(2), 143-162.
- Kukul, V., & Karatas, S. (2019). Computational thinking self-efficacy scale: Development, validity and reliability. *Informatics in Education*, 18(1), 151-164.
- Lockwood, E., DeJarnette, A. F., Asay, A., & Thomas, M. (2016). Algorithmic Thinking: An Initial Characterization of Computational Thinking in Mathematics. North American Chapter of the International Group for the Psychology of Mathematics Education.
- Mahmure, K. A. Y. A., Korkmaz, Ö., & ÇAKIR, R. (2020). Oyunlaştırılmış robot etkinliklerinin ortaokul öğrencilerinin problem çözme ve bilgi işlemsel düşünme becerilerine etkisi. *Ege Eğitim Dergisi*, 21(1), 54-70.
- Ng, O. L., & Cui, Z. (2020). Examining primary students' mathematical problem-solving in a programming context: towards computationally enhanced mathematics education. *ZDM*, 1-14.
- OECD ANNUAL REPORT 2005 45th ANNİVERSARY
- Okuducu, A. (2020). *Scratch destekli matematik öğretiminin 6. sınıf öğrencilerinin cebirsel ifadeler konusundaki akademik başarılarına ve tutumlarına etkisi* (Master's thesis, Fen Bilimleri Enstitüsü).
- Oluk, A. (2017). *Öğrencilerin bilgisayarca düşünme becerilerinin mantıksal matematiksel zekâ ve matematik akademik başarıları açısından incelenmesi* (Master's thesis, Fen Bilimleri Enstitüsü).
- Onur, T. O. P., & Arabacıoğlu, T. Bilgi İşlemsel Düşünme: Bir Sistematik Alanyazın Taraması. *Uludağ Üniversitesi Eğitim Fakültesi Dergisi*, 34(2), 527-567.
- Ozmen, Busra. (2016). Ortaokul Öğrencilerine Yönelik Bilgi İşlemsel Düşünme Becerileri Testinin Geliştirilmesi: Geçerlik ve Güvenirlik Çalışması.
- Ozmen, Busra. (2016). Ortaokul Öğrencilerine Yönelik Bilgi İşlemsel Düşünme Becerileri Testinin Geliştirilmesi: Geçerlik ve Güvenirlik Çalışması.
- Ökkeş, B. (2016). *Bilişimsel düşünme temelli ders etkinliklerinin öğrencilerin eleştirel düşünme becerileri ve problem çözme becerilerine yönelik algıları üzerine etkisinin incelenmesi* (Master's thesis, Mevlana University).
- Özçınar, H. (2017). Hesaplamalı Düşünme Araştırmalarının Bibliyometrik Analizi. *Eğitim Teknolojisi Kuram ve Uygulama*, 7 (2) , 149-171. DOI: 10.17943/etku.288610
- Özçınar, H. ve Öztürk, E. (2017). Hesaplamalı Düşünmenin Öğretimine İlişkin Özyeterlik Algısı Ölçeği, *Pamukkale Üniversitesi Sosyal Bilimler Enstitüsü Dergisi*, 30, 173-195.
- Özmen, B. (2020). Programlama Öğretiminde Bilgisayım Sal Düşünme Becerilerinin Geliştirilmesine Yönelik Oyun Tabanlı Bir Tasarım Modeli Önerisi.
- P21 Framework Definitions
- Papert, S. (1990). Children, computers and powerful ideas. *New York: Basic Books*, 10, 1095592.
- Papert, S. (1996). An exploration in the space of mathematics educations. *Int. J. Comput. Math. Learn.*, 1(1), 95-123.
- Papert, S. (1996). An exploration in the space of mathematics educations. *Int. J. Comput. Math. Learn.*, 1(1), 95-123.
- Psycharis, S., & Kallia, M. (2017). The effects of computer programming on high school students' reasoning skills and mathematical self-efficacy and problem solving. *Instructional science*, 45(5), 583-602.

- Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., ... & Kafai, Y. (2009). Scratch: Herkes için programlama. *ACM'nin İletişimi*, 52 (11), 60-67.
- Rich, K. M., Yadav, A., & Larimore, R. A. (2020). Teacher implementation profiles for integrating computational thinking into elementary mathematics and science instruction. *Education and Information Technologies*, 25(4), 3161-3188.
- Rich, K. M., Spaepen, E., Strickland, C., & Moran, C. (2020). Synergies and differences in mathematical and computational thinking: Implications for integrated instruction. *Interactive Learning Environments*, 28(3), 272-283.
- Rodríguez-Martínez, J. A., González-Calero, J. A., & Sáez-López, J. M. (2020). Computational thinking and mathematics using Scratch: an experiment with sixth-grade students. *Interactive Learning Environments*, 28(3), 316-327.
- Román-González, M., Pérez-González, 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.
- Sayın, Z. (2020). Öğretmenler İçin Bilgi İşlemsel Düşünmeye Özelleşmiş Bir Çevrimiçi Öğrenme Ortamının Tasarımı.
- Sayın, Z., & Seferoğlu, S. S. (2016). Yeni bir 21. yüzyıl becerisi olarak kodlama eğitimi ve kodlamanın eğitim politikalarına etkisi. *Akademik Bilişim Konferansı, 2016*, 3-5.
- Seferoğlu, S. S. Bir 21. Yüzyıl Becerisi Olarak Kodlamanın Önemi ve Eğitimdeki Yeri.
- Sırakaya, D. A. (2019). Programlama öğretiminin bilgi işlemsel düşünme becerisine etkisi. *Türkiye Sosyal Araştırmalar Dergisi*, 23(2), 575-590.
- Stigberg, H., & Stigberg, S. (2020). Teaching programming and mathematics in practice: A case study from a Swedish primary school. *Policy Futures in Education*, 18(4), 483-496.
- Sung, W., & Black, J. B. (2020). Factors to consider when designing effective learning: Infusing computational thinking in mathematics to support thinking-doing. *Journal of Research on Technology in Education*, 53(4), 404-426.
- Sung, W., Ahn, J., & Black, J. B. (2017). Introducing computational thinking to young learners: Practicing computational perspectives through embodiment in mathematics education. *Technology, Knowledge and Learning*, 22(3), 443-463.
- Sysło, M. M., & Kwiatkowska, A. B. (2013). Informatics for all high school students: A computational thinking approach. In I. Diethelm, & R. T. Mittermeir (Eds.), *Informatics in Schools: Situation, Evolution, and Perspectives* (Vol. 7780, pp. 43-56). Heidelberg: Springer.
- Şahiner, A., & Kert, S. B. (2016). Komputasyonel düşünme kavramı ile ilgili 2006-2015 yılları arasındaki çalışmaların incelenmesi. *Avrupa Bilim ve Teknoloji Dergisi*, 5(9), 38-43.
- Tosik-Gün, E., & Güyer, T. (2019). Bilgi işlemsel düşünme becerisinin değerlendirilmesine ilişkin sistematik alanyazın taraması. *Ahmet Keleşoğlu Eğitim Fakültesi Dergisi*, 1(2), 99-120.
- Tutulmaz, M. (2019). Bilgi-İşlemsel Düşünme Becerisinin Geliştirilmesine Yönelik Veri Görselleştirmenin Tasarlanması, Uygulanması ve Değerlendirilmesi.
- Uslu, N. A. (2018). Görsel programlama etkinliklerinin ortaokul öğrencilerinin BİD becerilerine etkisi. *Ege Eğitim Teknolojileri Dergisi*, 2(1), 19-31.

- Üzümçü, Ö., & Erdal, B. A. Y. (2018). Eğitimde yeni 21. yüzyıl becerisi: Bilgi işlemsel düşünme. *Uluslararası Türk Kültür Coğrafyasında Sosyal Bilimler Dergisi*, 3(2), 1-16.
- Wilkerson, M. H., & Fenwick, M. (2017). Using mathematics and computational thinking. *Helping students make sense of the world using next generation science and engineering practices*, 181-204.
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35.
- Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 366(1881), 3717-3725.
- Yeni, S. (2017). Bilgi işlemsel düşünme becerisi nasıl değerlendirilir? Y. Gülbahar (Ed.), *Bilgi işlemsel düşünmeden programlamaya içinde* (s. 359-391). Ankara: Pegem Akademi Yayıncılık.
- Yokuş, E., & Kahramanoğlu, R. An Overview of Computational Thinking. *Anemon Muş Alparslan Üniversitesi Sosyal Bilimler Dergisi*, 10(1), 157-173.
- Yünkül, E., Durak, G., Çankaya, S., & Mısırlı, Z. A. (2017). The effects of scratch software on students' computational thinking skills. *Necatibey Faculty of Education Electronic Journal of Science and Mathematics Education*, 11(2), 502-517.
- Zhang, L., & Nouri, J. (2019). A systematic review of learning computational thinking through Scratch in K-9. *Computers & Education*, 141, 103607.

## EKLER

### Ek 1. BİLGİ İŞLEMSEL DÜŞÜNME TESTİ

Bilgi İşlemsel Düşünme testine hoşgeldiniz!

#### AÇIKLAMALAR

Bu test, her sayfada 4 soru olacak şekilde, toplam 7 sayfadan ve 28 sorudan oluşmaktadır.

Her sorunun A, B, C ve D olmak üzere sadece biri doğru olan 4 seçeneği vardır . Testin süresi 45 dakikadır. Bu süre zarfında lütfen elinizden gelenin en iyisini yapınız. Yapamadığınız soruları atlayabilirsiniz.

Test sırasında, bir sayfadan diğerine geçmek için, sayfanın alt kısmında bulunan “Devam” tuşuna basmalısınız.

**ÇOK ÖNEMLİ:** Testi bitirdiğinizde veya süre dolduğunda, cevaplarınızı kaydetmek için, son sayfaya kadar ilerlemeniz ve “Gönder” tuşuna basmanız gerekmektedir.

Eğer herhangi bir soruyu detaylı görmek isterseniz, büyütme için “Ctrl +”, küçültme için “Ctrl -” tuşlarına basabilirsiniz.

Teste başlamadan önce, karşınıza çıkacak soru tiplerine ve karakterlere ('Pac-Man', Hayalet ve Ressam) alışmanız için 3 adet örnek soru göreceksiniz.

#### İYİ ŞANSLAR!



Pac-Man



Hayalet



Ressam

#### ÖRNEK 1

Birinci örnekte, Pac-Man'in işaretli yoldan hayalete ulaşması için hangi komutları takip etmesi gerektiği sorulmaktadır.

Yani, Pac-Man'i, sarı renkle işaretli kareleri takip ederek (yoldan çıkmadan ve turuncu karelere dokunmadan) TAM OLARAK hayaletin bulunduğu kareye (önceki veya sonraki karelere değil) nasıl götürürüz?

Bu sorudaki doğru cevap B seçeneğidir. Sorunun altındaki seçeneklerden B'yi işaretleyebilirsiniz.

Pac-Man'in işaretli yoldan hayaletle ulaşması için hangi komutları takip etmesi gerekmektedir?

	Seçenek A → → ↓
	Seçenek B → → ↑
	Seçenek C → ↑ ↑
	Seçenek D → ↓ ↓

Doğru seçeneği işaretleyiniz (bu örnekte, doğru cevap B'dir).

- A
- B
- C
- D

## ÖRNEK 2

Bu örnekte de öncekinde olduğu gibi, Pac-Man'in işaretli yoldan hayaletle ulaşması için hangi komutları takip etmesi gerektiği sorulmaktadır. Fakat bu sefer seçeneklerde oklar yerine birbirinin içine geçebilen komut blokları bulunmaktadır.

Pac-Man'i, sarı renkle işaretli kareleri takip ederek (yoldan çıkmadan ve turuncu karelere dokunmadan) TAM OLARAK hayaletin bulunduğu kareye (önceki veya sonraki karelere değil) nasıl götürürüz?

Bu sorudaki doğru cevap C seçeneğidir. Sorunun altındaki seçeneklerden C seçeneğini işaretleyebilirsiniz.

Pac-Man'in işaretli yoldan hayaletle ulaşması için hangi komutları takip etmesi gerekmektedir?

	Seçenek A ilerle sola dön ↶ ilerle ilerle	Seçenek B ilerle sağa dön ↷ ilerle ilerle
	Seçenek C ilerle ilerle sola dön ↶ ilerle	Seçenek D ilerle ilerle sağa dön ↷ ilerle

Doğru seçeneği işaretleyiniz (bu örnekte, doğru cevap C'dir).

- A
- B
- C
- D

### ÖRNEK 3

Üçüncü örnekte, ressamın ekrandaki şekli çizebilmesi için hangi komutları takip etmesi gerektiği sorulmaktadır. Yani, ressam şekli çizebilmek için kalemi nasıl HAREKET ETTİRMELİDİR?

TAŞI komutu ressamın kalemiyle çizim yaparak ilerlemesini sağlarken, ATLA komutu ise (ATLA komutu ile ilk defa 12. soruda karşılaşıacaksınız) ressamın çizim yapmadan ilerlemesini sağlar.

Gri ok, kalemin ilk hareket yönünü gösterir.

Bu sorudaki doğru cevap A seçeneğidir. Sorunun altındaki cevaplardan A seçeneğini işaretleyebilirsiniz.

Ressamın aşağıdaki şekli çizebilmesi için hangi komutları takip etmesi gerekmektedir? Şekildeki kısa çizgi 50 piksel, uzun çizgi ise 100 pikseldir.

	Seçenek A	Seçenek B
	Seçenek C	Seçenek D

Doğru seçeneği işaretleyiniz (bu örnekte, doğru cevap A 'dır).

- A
- B
- C
- D

## SORULAR

**Soru 1. Pac-Man'ın işaretli yoldan hayalet ulaşması için hangi komutları**

**takip etmesi gerekmektedir?**

Doğru seçeneği işaretleyiniz.

	Seçenek A 
	Seçenek B 
	Seçenek C 
	Seçenek D 

- A
- B
- C
- D

Soru 2. Pac-Man'in işaretli yoldan hayaletle ulaşması için (soldan sağa doğru)

	Seçenek A →
	Seçenek B ←
	Seçenek C ↑
	Seçenek D ↓

verilen komutlar arasındaki soru işaretli kısma hangisi gelmelidir? Doğru seçeneği işaretleyiniz.

- A
- B
- C
- D

Soru 3. Pac-Man'in işaretli yoldan hayaletle ulaşması için verilen komutlardaki hangi adım hatalıdır?

	ilerle → ADIM A
	sola dön ↶ ADIM B
	ilerle → ADIM C
	sola dön ↶ ADIM D

Doğru seçeneği işaretleyiniz.

- A
- B
- C
- D

Soru 4. Ressam aşağıdaki her bir kenarı 100 piksel olan kareyi çizebilmek için hangi komutları takip etmelidir?

	Seçenek A ilerle → 100 piksel kadar sağa → dön 90° derece ilerle → 100 piksel kadar sola → dön 90° derece ilerle → 100 piksel kadar sağa → dön 90° derece ilerle → 100 piksel	Seçenek B ilerle → 25 piksel kadar sağa → dön 90° derece ilerle → 25 piksel kadar sola → dön 90° derece ilerle → 25 piksel kadar sağa → dön 90° derece ilerle → 25 piksel
	Seçenek C ilerle → 50 piksel kadar sağa → dön 90° derece ilerle → 50 piksel kadar sola → dön 90° derece ilerle → 50 piksel kadar sağa → dön 90° derece ilerle → 50 piksel	Seçenek D ilerle → 100 piksel kadar sağa → dön 90° derece ilerle → 100 piksel kadar sola → dön 90° derece ilerle → 100 piksel kadar sağa → dön 90° derece ilerle → 100 piksel

Doğru seçeneği işaretleyiniz.

- A
- B
- C
- D

**Soru 5. Pac-Man'in işaretli yoldan hayaletle ulaşması için hangi komutları takip etmesi gerekmektedir?**

	Seçenek A	Seçenek B
	Seçenek C	Seçenek D

Doğru seçeneği işaretleyiniz.

- A
- B
- C
- D

**Soru 6. Pac-Man'in işaretli yoldan hayaletle ulaşması için aşağıda verilen**

	Seçenek A
	× 2
	Seçenek B
	× 1
Seçenek C	
× 4	
Seçenek D	
× 3	

komut bütünü kaç kere tekrar etmesi gerekmektedir? Doğru seçeneği işaretleyiniz.

- A
- B
- C
- D

**Soru 7. Ressamın aşağıdaki dikdörtgeni çizebilmesi için (kısa kenarlar 50 piksel, uzun kenarlar 100 piksel olacak şekilde) verilen komutlar içerisindeki hangi adımda hata vardır?**

	<p>Adım A</p> <p>bu işlemleri 4 kez tekrarla</p> <p>yap</p> <ul style="list-style-type: none"> <li>ileriye taşı 50 piksel</li> <li>kadar sola dön 90 derece</li> <li>ileriye taşı 100 piksel</li> <li>kadar sola dön 90 derece</li> </ul> <p>Adım B</p> <p>Adım C</p> <p>Adım D</p>
--	---

Doğru seçeneği işaretleyiniz.

- A
- B
- C
- D



**Soru 8. Pac-Man'in işaretli yoldan hayaletle ulaşması için hangi komutları takip etmesi gerekmektedir?**

	<p>Seçenek A</p> <pre>         Bu işlemleri 3 kez tekrarla         yap         Bu işlemleri 3 kez tekrarla         yap         ileri         sağa dön 90         ileri       </pre>	<p>Seçenek B</p> <pre>         Bu işlemleri 3 kez tekrarla         yap         Bu işlemleri 3 kez tekrarla         yap         ileri         sağa dön 90         ileri       </pre>
	<p>Seçenek C</p> <pre>         Bu işlemleri 3 kez tekrarla         yap         Bu işlemleri 3 kez tekrarla         yap         ileri         sağa dön 90         ileri       </pre>	<p>Seçenek D</p> <pre>         Bu işlemleri 3 kez tekrarla         yap         ileri         Bu işlemleri 3 kez tekrarla         yap         sağa dön 90         ileri       </pre>

Doğru işaretleyiniz. seçeneği

- A
- B
- C
- D

**Soru 9. Pac-Man'in işaretli yoldan hayaletle ulaşması için hangi komutları takip etmesi gerekmektedir?**

	<p>Seçenek A</p>	<p>Seçenek B</p>
	<p>Seçenek C</p>	<p>Seçenek D</p>

Doğru işaretleyiniz. seçeneği

- A
- B
- C
- D

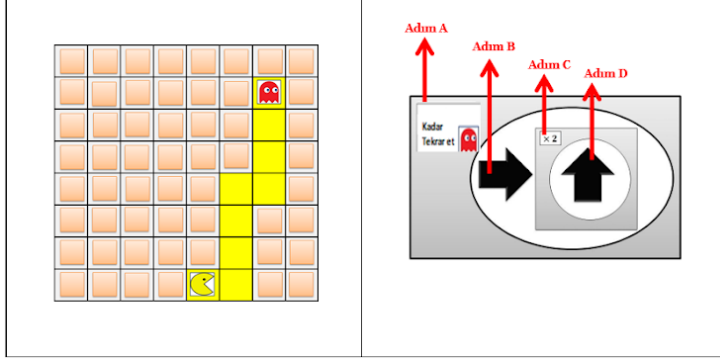
**Soru 10. Pac-Man'in işaretli yoldan hayaletle ulaşması için aşağıda verilen komutlar içindeki eksik adım hangisidir?**

<pre>         kadar tekrarla         yap         sola dön 90         ileri         ileri         sağa dön 90         ileri       </pre>	<p>Seçenek A</p>	<p>Seçenek B</p>
	<p>Seçenek C</p>	<p>Seçenek D</p> <p>Eksik Adım Yok</p>

Doğru işaretleyiniz. seçeneği

- A
- B
- C
- D

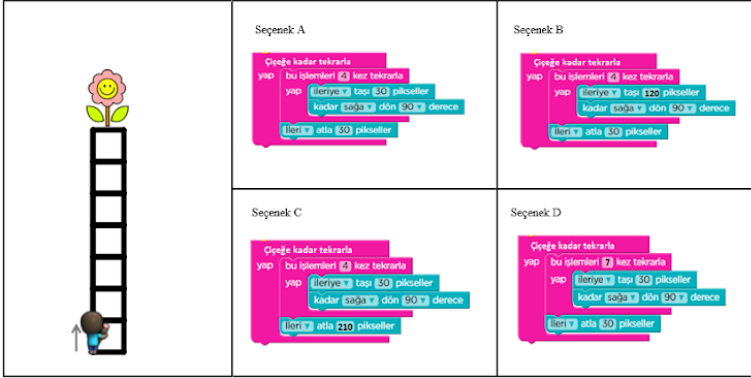
**Soru 11. Pac-Man'in işaretli yoldan hayaletle ulaşması için gerekli komutlar aşağıda verilmiştir. Komutların hangi adımında hata vardır?**



Doğru seçeneği işaretleyiniz.

- A
- B
- C
- D

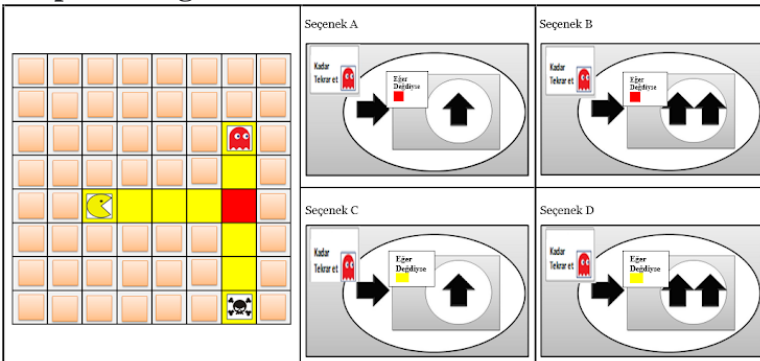
**Soru 12. Ressamın resimdeki çiçeğe ulaşan merdiveni çizebilmesi için hangi komutları takip etmesi gerekmektedir? Her bir basamak 30 pikseldir.**



Doğru seçeneği işaretleyiniz.

- A
- B
- C
- D

**Soru 13. Pac-Man'in işaretli yoldan hayaletle ulaşması için hangi komutları takip etmesi gerekmektedir?**



Doğru seçeneği işaretleyiniz.

- A
- B
- C
- D

**Soru 14. Pac Man'in işaretli yoldan hayaletle ulaşması için hangi komutları takip etmesi gerekmektedir?**

	<p>Seçenek A</p> <pre> kadar tekrarla yap ileri eğer sağa doğru yol varsa döndür yap sağa dön </pre>	<p>Seçenek B</p> <pre> kadar tekrarla yap sağa dön eğer sağa doğru yol varsa döndür yap ileri </pre>
	<p>Seçenek C</p> <pre> kadar tekrarla yap ileri eğer sağa doğru yol varsa döndür yap sola dön </pre>	<p>Seçenek D</p> <pre> kadar tekrarla yap ileri eğer sola doğru yol varsa döndür yap sola dön </pre>

Doğru seçeneği işaretleyiniz.

- A
- B
- C
- D

**Soru 15. Pac-Man'in işaretli yoldan hayaletle ulaşabilmesi için aşağıda verilen**

	<p>Seçenek A</p> <p>Yalnız Mavi Kare</p>
	<p>Seçenek B</p> <p>Yalnız Sarı Kare</p>
	<p>Seçenek C</p> <p>Yalnız Kırmızı Kare</p>
	<p>Seçenek D</p> <p>Mavi Kare Veya Kırmızı Kare</p>

komutlar içindeki soru işaretli kısma hangi renkli kare veya kareler gelebilir?

Doğru seçeneği işaretleyiniz.

- A
- B
- C
- D

**Soru 16. Aşağıda Pac-Man'in işaretli yoldan hayaletle ulaşması için gerekli komutlar verilmiştir. Komutların hangi adımında hata bulunmaktadır?**

	<pre> kadar tekrarla yap ileri eğer sola doğru yol varsa döndür yap sola dön eğer sağa doğru yol varsa döndür yap ileri </pre>
	<p>Adım A</p> <p>Adım B</p> <p>Adım C</p> <p>Adım D</p>

Doğru seçeneği işaretleyiniz.

- A
- B
- C
- D

**Soru 17. Pac-Man'in işaretli yoldan hayaletle ulaşması için hangi komutları takip etmesi gerekmektedir?**

	<p>Seçenek A</p>	<p>Seçenek B</p>
	<p>Seçenek C</p>	<p>Seçenek D</p>

Doğru seçeneği işaretleyiniz.

- A
- B
- C
- D

**Soru 18. Pac-Man'in işaretli yoldan hayaletle ulaşması için hangi komutları takip etmesi gerekmektedir?**

	<p>Seçenek A</p>	<p>Seçenek B</p>
	<p>Seçenek C</p>	<p>Seçenek D</p>

Doğru seçeneği işaretleyiniz.

- A
- B
- C
- D

**Soru 19. Pac-Man'in işaretli yoldan hayaletle ulaşması için verilen komutlar içindeki hangi adım hatalıdır?**

--	--

Doğru seçeneği işaretleyiniz.

- A
- B
- C
- D

Soru 20. Pac-Man'in işaretli yoldan hayaletle ulaşması için verilen komutlar

içindeki soru işaretli kısma hangi komut gelmelidir?

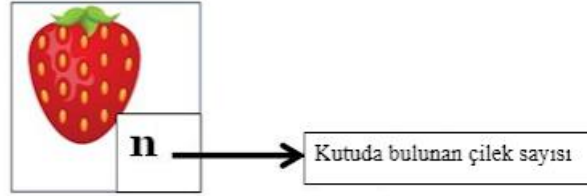
Doğru seçeneği işaretleyiniz.

	Seçenek A ilerle	Seçenek B sağa dön
	Seçenek C sola dön	Seçenek D Eksik Komut bulunmamaktadır

- A
- B
- C
- D

ÖNEMLİ: DİKKATLİCE OKUYUNUZ

Bu gruptaki sorularda, içinde çilek resmi bulunan kutularla karşılaşabilirsiniz. Kutunun sağ alt köşesindeki sayı, kutuda bulunan çilek sayısını göstermektedir.



Soru 21. Pac-Man'in işaretli yoldan çileklere ulaşması ve kutuda gösterilen çileklerin tamamını yemesi için hangi komutları takip etmesi gerekmektedir?

Doğru seçeneği işaretleyiniz.

	Seçenek A ileride yol olduğu sürece yap ilerle bu işlemleri 3 kez tekrarla yap 1 çilek ye	Seçenek B ileride yol olduğu sürece yap ilerle bu işlemleri 4 kez tekrarla yap 1 çilek ye
	Seçenek C ileride yol olduğu sürece yap ilerle bu işlemleri 3 kez tekrarla yap 1 çilek ye	Seçenek D ileride yol olduğu sürece yap ilerle bu işlemleri 3 kez tekrarla yap 1 çilek ye

- A
- B
- C
- D

**Soru 22. Pac-Man'in işaretli yoldan çileklere ulaşması ve kutularda gösterilen çileklerin tamamını yemesi için hangi komutları takip etmesi gerekmektedir?**

	<p>Seçenek A</p> <pre> ileride yol olduğu sürece yap bu işlemleri 3 kez tekrarla yap ilerle bu işlemleri 3 kez tekrarla yap 1 çilek ye         </pre>	<p>Seçenek B</p> <pre> ileride yol olduğu sürece yap ilerle bu işlemleri 3 kez tekrarla yap 1 çilek ye         </pre>
	<p>Seçenek C</p> <pre> ileride yol olduğu sürece yap bu işlemleri 3 kez tekrarla yap ilerle bu işlemleri 3 kez tekrarla yap 1 çilek ye         </pre>	<p>Seçenek D</p> <pre> ileride yol olduğu sürece yap ilerle bu işlemleri 3 kez tekrarla yap 1 çilek ye         </pre>

Doğru seçeneği işaretleyiniz.

- A  
B  
C  
D

**Soru 23. Pac-Man'in işaretli yoldan çileklere ulaşması ve kutularda gösterilen çileklerin tamamını yemesi için verilen komut dizisinde soru işaretli kısma hangisi gelmelidir?**

<pre> ileride yol olduğu sürece yap bu işlemleri 222 kez tekrarla yap ilerle eğer çilek varsa yap 1 çilek ye         </pre>	<p>Seçenek A</p> <p>1 kez</p>
	<p>Seçenek B</p> <p>2 kez</p>
	<p>Seçenek C</p> <p>3 kez</p>
	<p>Seçenek D</p> <p>5 kez</p>

Doğru seçeneği işaretleyiniz.

- A  
B  
C  
D

**Soru 24. Pac-Man'in işaretli yoldan çileklere ulaşması ve kutularda gösterilen çileklerin tamamını (çilek sayısı bilinmiyor) yemesi için verilen komut dizisinde soru işaretli kısma hangisi gelmelidir?**

<pre> ileride yol olduğu sürece yap ilerle eğer çilek varsa yap ????????????????????????????????? yap 1 çilek ye         </pre>	<p>ileride yol olduğu sürece</p> <p>Seçenek A</p>
	<p>ileride yol olmadığı sürece</p> <p>Seçenek B</p>
	<p>çilek olduğu sürece</p> <p>Seçenek C</p>
	<p>çilek olmadığı sürece</p> <p>Seçenek D</p>

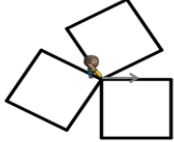
Doğru seçeneği işaretleyiniz.

Doğru seçeneği işaretleyiniz.

- A  
B  
C  
D

**Soru 25. Ressamın aşağıdaki her bir kenarı 100 piksel olan karelerden oluşan**


**şekli çizmesi için hangi komutları takip etmesi gerekmektedir? Doğru seçeneği işaretleyiniz.**

<p>Aşağıdaki komutlar her bir kenarı 100 piksel olan bir kare çizmektedir ve bu komutlar "kare çiz" olarak adlandırılmaktadır.</p> <pre> Fonksiyon kare çiz bu işlemleri 3 kez tekrarla yap ileriye 1 taşı 100 piksel kadar sağa dön 90 derece         </pre> <p>Ressamın aşağıdaki her bir kenarı 100 piksel olan karelerden oluşan şekli çizmesi için hangi komutları takip etmesi gerekmektedir?</p> 	<p>Seçenek A</p> <pre> bu işlemleri 3 kez tekrarla yap kare çiz kadar sağa dön 120 derece         </pre>	<p>Seçenek B</p> <pre> bu işlemleri 3 kez tekrarla yap kare çiz kadar sağa dön 120 derece         </pre>
	<p>Seçenek C</p> <pre> bu işlemleri 4 kez tekrarla yap kare çiz kadar sağa dön 90 derece         </pre>	<p>Seçenek D</p> <pre> bu işlemleri 4 kez tekrarla yap kare çiz kadar sağa dön 90 derece         </pre>

- A
- B
- C
- D

**Soru 26. Ressamın aşağıdaki her bir kenarı 50 piksel olan üçgenlerden oluşan**

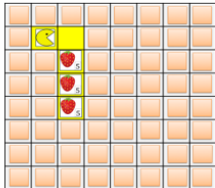
**şekli çizmesi için verilen komutlar içindeki soru işaretli kısma hangisi gelmelidir? Doğru seçeneği işaretleyiniz.**

<p>Aşağıdaki komutlar her bir kenarı 50 piksel olan bir kare çizmektedir ve bu komutlar "üçgen çiz" olarak adlandırılmaktadır.</p> <pre> Fonksiyon üçgen çiz bu işlemleri 3 kez tekrarla yap ileriye 1 taşı 50 piksel kadar sola dön 120 derece         </pre> <p>Ressamın aşağıdaki her bir kenarı 50 piksel olan üçgenlerden oluşan şekli çizmesi için verilen komutlar içindeki soru işaretli kısma hangisi gelmelidir?</p> <pre> bu işlemleri ??? kez tekrarla yap üçgen çiz ileri atla 50 piksel         </pre> 	<p>Seçenek A</p> <p>15</p>	<p>Seçenek B</p> <p>5</p>
	<p>Seçenek C</p> <p>4</p>	<p>Seçenek D</p> <p>3</p>

- A
- B
- C
- D

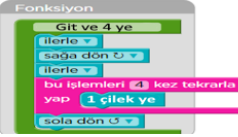
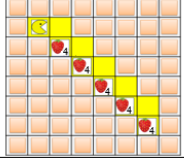
**Soru 27. Pac-Man'in işaretli yoldan çileklere ulaşması ve kutularda gösterilen çileklerin tamamını yemesi için hangi komutları takip etmesi gerekmektedir?**

**Doğru seçeneği işaretleyiniz.**

<p>Aşağıdaki komutları "5 ye" olarak adlandıralım</p> <pre> Fonksiyon 5 ye bu işlemleri 5 kez tekrarla yap 1 çilek ye         </pre> <p>Pac-Man'in işaretli yoldan çileklere ulaşması ve kutularda gösterilen çileklerin tamamını yemesi için hangi komutları takip etmesi gerekmektedir?</p> 	<p>Seçenek A</p> <pre> ileri ye sağa dön 90 bu işlemleri 3 kez tekrarla yap ileri ye 5 ye         </pre>	<p>Seçenek B</p> <pre> ileri ye sağa dön 90 bu işlemleri 3 kez tekrarla yap 5 ye ileri ye         </pre>
	<p>Seçenek C</p> <pre> ileri ye sağa dön 90 bu işlemleri 5 kez tekrarla yap ileri ye 5 ye         </pre>	<p>Seçenek D</p> <pre> ileri ye sağa dön 90 bu işlemleri 5 kez tekrarla yap 5 ye ileri ye         </pre>

- A
- B
- C
- D

**Soru 28. Pac-Man'in işaretli yoldan çileklere ulaşması ve kutularda gösterilen çileklerin tamamını yemesi için aşağıda verilen komutlar içindeki soru işaretli kısma hangisi gelmelidir?**

<p>Aşağıdaki komutları "Git ve 4 ye" olarak adlandıralım</p>  <p>Pac-Man'in işaretli yoldan çileklere ulaşması ve kutularda gösterilen çileklerin tamamını yemesi için aşağıda verilen komutlar içindeki soru işaretli kısma hangisi gelmelidir?</p>	Seçenek A <b>3</b>	Seçenek B <b>4</b>
<p>bu işlemleri ??? kez tekrarla yap Git ve 4 ye</p> 	Seçenek C <b>5</b>	Seçenek D <b>6</b>

Doğru işaretleyiniz. seçeneği

- A
- B
- C
- D



## Ek 2. Ortaokul Öğrencileri İçin Bilgi İşlemsel Düşünme Becerisine Yönelik Öz Yeterlik Algısı Ölçeği

Değerli katılımcı, aşağıda yer alan ifadelere ilişkin, her madde içerisinde sunulan 3 seçenekten (Evet-1, Kısmen-2, Hayır-3) size en uygun olanı işaretleyiniz. Tercihlerinizin doğru ya da yanlış olarak bir değerlendirilmesi yapılmayacaktır. İfadelere, düşünerek ve içtenlikle vereceğiniz cevaplar için teşekkür ederiz.

Sıra No	Soru Metni	Evet	Kısmen	Hayır
<b>Algoritma Tasarlama Yeterliği</b>				
1	Algoritmaların hangi amaçla kullanıldığını anlıyorum.	1	2	3
2	Algoritmanın ne olduğunu biliyorum.	1	2	3
3	Basit algoritmalar oluşturabilirim.	1	2	3
4	Koşullu algoritmalar oluşturabilirim.	1	2	3
5	Döngü yapısında algoritmalar oluşturabilirim.	1	2	3
6	Algoritma oluştururken mantıksal sorgulama yapabilirim.	1	2	3
7	Bir algoritmanın çıktısının ne olacağını tahmin edebilirim.	1	2	3
8	Algoritmada bulunan hataları ayıklayabilirim.	1	2	3
9	Algoritmaların dijital araçlar için nasıl koda dönüştürüleceğini anlıyorum.	1	2	3
<b>Problem Çözme Yeterliği</b>				
10	Problemi çözüp sonucunu bulduktan sonra yaptığım işlemleri kontrol ederim.	1	2	3
11	Problemi çözüp sonucunu bulduktan sonra yaptığım işlemleri kontrol eder varsa hataları düzeltirim.	1	2	3
12	Bir problemi okuduğumda, çözüm için hangi bilgiye ihtiyacım olduğunu düşünürüm.	1	2	3
13	Problem çözüm sürecinde işlem önceliklerine dikkat ederim.	1	2	3
14	Bir problemi okuduğumda, çözüm için gerekli ve gereksiz olan bilgiyi ayırt edebilirim.	1	2	3
15	Farklı çözüm yollarını inceleyerek daha iyi bir çözüm bulmaya çalışırım.	1	2	3
16	Bir problemi okuduğumda, daha önce çözdüğüm problemleri düşünerek benzerlik ve farklılıklarına göre aralarında ilişki kurarım.	1	2	3
17	Problem çözerken, hangi işlemi neden yaptığımı sürekli sorgularım.	1	2	3
18	Bir problemi çözebilmem için yeterli veri sunulup sunulmadığına karar verebilirim.	1	2	3
19	Bir problem için ürettiğim çözümü farklı problemlere genelledebilirim.	1	2	3
<b>Veri İşleme Yeterliği</b>				
20	Verinin ne olduğunu biliyorum.	1	2	3
21	Veri toplamının önemini anlıyorum.	1	2	3
22	Verinin farklı türleri olduğunun (sayı ve metin) farkındayım.	1	2	3
23	Veri ve bilgi arasındaki farklı açıklayabilirim.	1	2	3
24	Problemlerin çözümünde farklı veri türleri kullanılabileceğini biliyorum.	1	2	3
25	Verilerin tablo yapısında daha anlamlı sunulabildiğini biliyorum.	1	2	3
26	Dijital verinin farklı biçimlere dönüşebileceğini biliyorum.	1	2	3
<b>Temel Programlama Yeterliği</b>				

27	Değişkenleri tanımlayıp kullanabilirim.	1	2	3
28	Koşullu yapıları ve döngüleri oluştururken aritmetik operatörleri kullanabilirim.	1	2	3
29	Bir döngüyü sonlandırmak için değişken ve ilişkisel operatörleri kullanabilirim.	1	2	3
30	Farklı kontrol durumları için değişik döngüler oluşturabilirim.	1	2	3
31	Belirli işlemler için hazır fonksiyonları kullanabilirim.	1	2	3
<b>Özgüven Yeterliği</b>				
32	Yönergelerin ve işlem adımlarının önemini biliyorum.	1	2	3
33	Çözümleri göstermek için şemalar kullanabilirim.	1	2	3
34	Aynı problem için farklı çözümler üretebileceğimin farkındayım.	1	2	3
35	Problem çözme sürecinde hatalarımı nasıl düzelteceğimi biliyorum.	1	2	3
36	Dijital araçlar tarafından en iyi başarılan işlemlerin ne olduğunun farkındayım.	1	2	3

### Ek 3. Araştırma İzni



T.C.  
SAMSUN VALİLİĞİ  
İl Millî Eğitim Müdürlüğü

Sayı : E-27485554-605.01-50556525  
Konu : Muradiye BOZAL

27.05.2022

#### DAĞITIM YERLERİNE

İlgi : a) Millî Eğitim Bakanlığı Yenilik ve Eğitim Teknolojileri Genel Müdürlüğünün  
21/01/2020 tarihli ve 81576613-10.06.01-E. 1563890- 2020/2 sayılı Genelgesi,  
b) Ondokuz Mayıs Üniversitesi Lisansüstü Eğitim Enstitüsü Müdürlüğünün bila tarihli ve 238739  
sayılı yazısı.

Ondokuz Mayıs Üniversitesi Lisansüstü Eğitim Enstitüsü Bilgisayar ve Öğretim Teknolojileri  
eğitimi Anabilim Dalı Yüksek Lisans Öğrencisi Muradiye BOZAL'ın ; İlimiz İlkadım ilçesindeki  
İlkadım Gazi ortaokulu 6. sınıf öğrencilerine yönelik " Matematik Destekli Temel Programlama Eğitimin  
in Bilgi İşlemsel Düşünme Becerisi Üzerindeki Etkisi " başlıklı tez çalışması yapmak istediğine ilişkin  
ilgi (b) yazı ve ekleri, ilgi (a) genelgeye göre incelenmiş ve komisyon tarafından uygun görülmüştür.

Söz konusu çalışmanın komisyon kararı doğrultusunda, uygulama sorularını çalışmayı yapan kişi  
tarafından raporlanarak, Müdürlüğümüz Ar-Ge Birimine gönderilmesine dikkat edilerek, yüz yüze eğitim  
öğretime ara verilmesi gözönüne alınarak online, örgün eğitimin tam olarak başlamasıyla birlikte  
denetimi ilçe millî eğitim müdürlükleri/okul idaresinde olmak üzere, kurum faaliyetlerini aksatmadan,  
gönüllülük esasına göre yapılmasının sağlanması hususunda;

Bilgilerinizi ve gereğini rica ederim.

Murat YIĞIT  
Vali a.  
İl Millî Eğitim Müdürü

Ekler :  
1- İlgi (b) yazı ve ekleri  
2- 20/05/2022 tarihli komisyon kararı

#### DAĞITIM:

Gereği:  
İlkadım İlçe Kaymakamlığına  
( İlçe Millî Eğitim Müdürlüğü )

Bilgi:  
Ondokuz Mayıs Üniversitesi Rektörlüğü  
Lisansüstü Eğitim Enstitüsü Müdürlüğü

Bu belge güvenli elektronik imza ile imzalanmıştır.  
Adres : Kale Mahallesi Atatürk Bulvarı Yeni Hükümet Konağı Kat:3  
55030 SAMSUN  
Telefon No : 0(362)4358063  
E-Posta: yuksekogretimyurtidasi55@meb.gov.tr  
Kep Adresi : meb@hs01.kep.tr

Belge Doğrulama Adresi : <https://www.turkiye.gov.tr/meb-ebys>  
Bilgi için: S.SUZUKI(340)  
Unvan : Bilgisayar İşletmeni  
İnternet Adresi: samsun.meb.gov.tr Faks 3624319376

Bu evrak güvenli elektronik imza ile imzalanmıştır. <https://evrakorgu.meb.gov.tr> adresinden 6c4a-8b2d-333d-83fe-6c64 kodu ile teyit edilebilir.



## Ek 4. Etik Kurul Kararı



**ONDOKUZ MAYIS ÜNİVERSİTESİ**  
**SOSYAL VE BEŞERİ BİLİMLER ARAŞTIRMALARI ETİK KURUL KARARLARI**

KARAR TARİHİ	TOPLANTI SAYISI	KARAR SAYISI
25.06.2021	06	2021/568

**KARAR NO:** 2021-568  
Üniversitemiz Lisansüstü Eğitim Enstitüsü öğrencisi Muradiye BOZAL' ın Doç. Dr. Polat ŞENDURUR danışmanlığında "Matematik Destekli Temel Programlama Eğitiminin Bilgi İşlemsel Düşünme Becerisi Üzerindeki Etkisi" isimli yüksek lisans tezine ilişkin ölçek ve test çalışmalarını içeren 18393 sayılı dilekçesi okunarak görüşüldü.

Üniversitemiz Lisansüstü Eğitim Enstitüsü öğrencisi Muradiye BOZAL' ın Doç. Dr. Polat ŞENDURUR danışmanlığında "Matematik Destekli Temel Programlama Eğitiminin Bilgi İşlemsel Düşünme Becerisi Üzerindeki Etkisi" isimli yüksek lisans tezine ilişkin ölçek ve test çalışmalarının kabulüne oy birliği ile karar verildi.

## ÖZ GEÇMİŞ

Muradiye BOZAL, Samsun Anadolu Kız Meslek Lisesi Bilgisayar Bölümünü bitirdikten sonra Ege Üniversitesi Eğitim Fakültesi BÖTE bölümünden 2006 yılında mezun oldu. 2020 yılında OMÜ LEE Bilgisayar ve Öğretim Teknolojileri Eğitimi Ana Bilim Dalı Yüksek Lisans programına girdi. Mezuniyetinden bu yana MEB’de Bilişim Teknolojileri öğretmeni olarak görev yapmaktadır. Robotik kodlama alanında çeşitli eğitimler almıştır.

### İletişim Bilgileri

ORCID ID : 0000-0002-8616-9223

### Yayınlar:

1. Bozal, M. ve Şendurur, P. (2022). Matematik destekli temel programlama eğitiminin bilgi işlemsel düşünme becerisi üzerindeki etkisi. *International Conference on Educational Technology and Online Learning (2022)*, Balıkesir, Türkiye.