



**ONDOKUZ MAYIS ÜNİVERSİTESİ**

**EĞİTİM BİLİMLERİ ENSTİTÜSÜ**

**BİLGİSAYAR VE ÖĞRETİM TEKNOLOJİLERİ EĞİTİMİ**

**ANABİLİM DALI**

**PROGRAMLAMA BAŞARISINI ETKİLEYEN BAZI  
FAKTÖRLERİN İNCELENMESİ**

**Samet ATABAŞ**

**Danışman**

**Doç. Dr. Ufuk ÇORUH**

**YÜKSEK LİSANS TEZİ**

**OCAK, 2018**

## TELİF HAKKI

Bu tezin tüm hakları saklıdır. Kaynak göstermek koşuluyla tezin teslim tarihinden itibaren tezden fotokopi çekilebilir.

### YAZARIN

Adı : Samet

Soyadı : ATABAŞ

Bölümü : Bilgisayar ve Öğretim Teknolojileri Eğitimi

İmza :

Teslim Tarihi :

### TEZİN

Türkçe Adı : Programlama Başarısını Etkileyen Bazı Faktörlerin İncelenmesi

İngilizce Adı : Investigation of Some Factors Affecting Programming Success

## ETİK İLKELERE UYGUNLUK BEYANI

Tez yazma sürecinde bilimsel ve etik ilkelere uyduđumu, yararlandıđım tüm kaynakları kaynak gösterme ilkelerine uygun olarak kaynakçada belirttiđimi ve bu bölümler dışındaki tüm ifadelerin şahsıma ait olduđunu beyan ederim.

Yazar Adı Soyadı: Samet ATABAŞ

İmza:.....

## KABUL VE ONAY

Samet ATABAŞ tarafından hazırlanan “Programlama Başarısını Etkileyen Bazı Faktörlerin İncelenmesi” adlı tez çalışması aşağıdaki jüri tarafından oy birliği/oy çokluğu ile Ondokuz Mayıs Üniversitesi Bilgisayar ve Öğretim Teknolojileri Eğitimi Anabilim Dalı’nda Yüksek Lisans / Doktora tezi olarak kabul edilmiştir.

**Danışman:** (Unvanı Adı Soyadı)

(Anabilim Dalı, Üniversite Adı)

.....

**Başkan:** (Unvanı Adı Soyadı)

(Anabilim Dalı, Üniversite Adı)

.....

**Üye:** (Unvanı Adı Soyadı)

(Anabilim Dalı, Üniversite Adı)

.....

**Üye:** (Unvanı Adı Soyadı)

(Anabilim Dalı, Üniversite Adı)

.....

**Üye:** (Unvanı Adı Soyadı)

(Anabilim Dalı, Üniversite Adı)

.....

Bu tezin **Bilgisayar ve Öğretim Teknolojileri Eğitimi** Anabilim Dalı’nda Yüksek Lisans/ Doktora tezi olması için şartları yerine getirdiğini onaylıyorum.

Tarihi: \_\_/\_\_/\_\_

Eğitim Bilimleri Enstitüsü Müdürü

(İmza ve Mühür)



*Aileme ...*

## TEŐEKKÖRLER

Başlangıç seviyesinde programlama eğitimi alan öğrencilerin programlama başarılarına etki eden bazı faktörlerin incelenmesi amacıyla yapılan bu çalışma Ondokuz Mayıs Üniversitesi Eğitim Bilimleri Enstitüsü Bilgisayar ve Öğretim Teknolojileri Eğitimi Ana Bilim dalında yüksek lisans tez çalışması olarak yürütölmüştür. Bu çalışma Ondokuz Mayıs Üniversitesi Lisansüstü Tezleri Destekleme Programı kapsamında desteklenmiştir.

Çalışma boyunca her türlü desteęi esirgemeyen aileme, danışmanım Doç. Dr. Ufuk ÇORUH'a, veri toplama süreci ve dięer süreçlerde yardımını esirgemeyen değerli hocam Yrd. Doç. Dr. Polat ŐENDURUR'a teşekkürlerimi sunuyorum.

# **PROGRAMLAMA BAŞARISINI ETKİLEYEN BAZI FAKTÖRLERİN İNCELENMESİ**

**Yüksek Lisans Tezi**

**Samet ATABAŞ**

**ONDOKUZ MAYIS ÜNİVERSİTESİ**

**EĞİTİM BİLİMLERİ ENSTİTÜSÜ**

**Ocak, 2018**

**ÖZ**

Araştırma öğrencilerin bilgisayar programlama dersinde başarısız olması problemine yönelik olarak programlama başarısını etkileyen bazı faktörlerin belirlenmesi ve incelenmesi amacıyla yapılmıştır. Araştırma 2016-2017 eğitim öğretim yılında Ondokuz Mayıs Üniversitesi Bilgisayar ve Öğretim Teknolojileri Eğitimi ikinci sınıf öğrencileri ile yürütülmüştür. Çalışmaya Programlama Dilleri I dersini alan 67 öğrenci katılmıştır. Araştırmada programlama yeteneği, mantıksal düşünme yeteneği, öğrenme stili, öz yeterlilik, bilgisayar deneyimi ve programlama deneyimi değişkenleri ile programlama başarısı arasındaki ilişkiler incelenmiştir. Araştırmada Bilgisayar Programlama Yetenek Testi, Mantıksal Düşünme Yeteneği Testi, Kolb Öğrenme Stilleri Envanteri (III), Programlamaya İlişkin Öz yeterlilik Ölçeği ve Programlama Başarı testleri kullanılarak veri toplanmıştır. Programlama yeteneği, mantıksal düşünme yeteneği, programlama öz yeterliliği, bilgisayar deneyimi ve programlama deneyimi ile programlama başarısı arasındaki ilişkiyi belirlemek için korelasyon analizi yapılmıştır. Öğrencilerin sahip oldukları öğrenme stiline göre programlama başarıları arasında bir fark olup olmadığını anlamak için ilişkisiz örneklem için tek yönlü varyans analizi yapılmıştır. Programlama yeteneği, mantıksal düşünme yeteneği, öğrenme stili ve programlama öz yeterliliği birlikte programlama başarısını hangi önem sırasında ve ne düzeyde açıklayabildiğini belirlemek için çoklu regresyon analizi yapılmıştır. Araştırma sonucunda programlama yeteneği, mantıksal düşünme yeteneği ve bilgisayar deneyimi ile programlama başarısı arasında istatistiksel olarak anlamlı bir ilişki bulunamamıştır. Programlama öz yeterliliği ile programlama başarısı arasında orta seviyede bir ilişki bulunmuştur. Öğrencilerin programlama ile uğraştıkları yıl baz alınarak hesaplanan programlama deneyim ile programlama başarısı arasında düşük seviyede bir ilişki

bulunmuştur. Çalışmaya katılan öğrenciler arasında en çok tercih edilen öğrenme stili Özümseme, ikinci sıradaki stil ise Ayrıştırma stilidir. En başarılı stil Ayrıştırma stili olurken, en başarısız stil Özümseme olarak belirlenmiştir. Yapılan ANOVA testi sonucunda öğrenme stilleri ortalamaları arasında anlamlı bir fark olduğu bulunmuştur. Yapılan regresyon analizine göre programlama başarısının %41'i ele alınan bağımsız değişkenler tarafından açıklanabilmektedir.

**Anahtar Kelimeler : programlama yeteneği, mantıksal düşünme yeteneği, öğrenme stilleri, programlama öz yeterliliği, programlama başarısı**

**Sayfa Sayısı : 108**

**Danışman : Doç. Dr. Ufuk ÇORUH**



# **INVESTIGATION OF SOME FACTORS AFFECTING PROGRAMMING SUCCESS**

**MS Thesis**

**Samet ATABAŞ**

**ONDOKUZ MAYIS UNIVERSITY**

**GRADUATE SCHOOL OF EDUCATIONAL SCIENCES**

**January, 2018**

## **ABSTRACT**

This research is conducted to define and investigate the factors that impact the programming success which comes from the problem of students being unsuccessful in programming lessons. The research is undertaken by Computer Education and Instructional Technology department, second year students in 2016-2017 academic year at Ondokuz Mayıs University. 67 students from the class of Programming Languages 1 contributed to the study. In this research, relationship between programming ability, logical thinking ability, learning style, computer programming self-efficacy, computer experience, programming experience and programming success are studied. Data was collected by using computer programming ability test, logical thinking ability test, Kolb learning styles inventory (III), computer programming self-efficacy and programming success tests. Correlation analysis has been applied to define the relation between programming ability, logical thinking ability, programming self-efficacy, computer experience, programming experience and programming success. In order to understand how programming success varies according to learning styles, one way variance analysis is conducted for unrelated random samples. Multiple regression analysis was applied to describe the programming success in which level and importance with programming ability, logical thinking ability, learning style, programming self-efficacy. According to the results of the study, a significant relation between programming ability, logical thinking ability, computer experience and programming success was not found. However, there was a mid-level relation between programming self-efficacy and programming success. There was a low level of relationship between programming experience and programming success, calculated on the basis of the years students were involved with programming. The most preferred learning style was Assimilating followed by Converging, whereas the most successful one was

Converging and the less successful was Assimilating. A significantly difference between training styles' averages was found by ANOVA test. According to regression analysis 41% of the programming success can be explained by independent variances.

**Key Words** : programming ability, logical thinking ability, learning styles, programming self-efficacy, programming success

**Number of Pages** : 108

**Advisor** : Assoc. Dr. Ufuk ÇORUH



# İÇİNDEKİLER

TELİF HAKKI.....	II
ETİK İLKELERE UYGUNLUK BEYANI.....	III
KABUL VE ONAY .....	IV
TEŞEKKÜRLER .....	VI
ÖZ.....	VII
ABSTRACT .....	IX
İÇİNDEKİLER .....	XI
TABLolar LİSTESİ.....	XIV
ŞEKİLLER LİSTESİ.....	XV
BİRİNCİ BÖLÜM.....	1
1 GİRİŞ.....	1
1.1 Problem Durumu.....	3
1.2 Çalışmanın Amacı .....	5
1.3 Çalışmanın Önemi.....	5
1.4 Varsayımlar .....	8
1.5 Sınırlılıklar .....	8
İKİNCİ BÖLÜM.....	9
2 Kuramsal Çerçeve ve İlgili Araştırmalar .....	9
2.1 Programlama Nedir .....	9
2.2 Neden Programlama Öğrenmeliyiz .....	11
2.3 Bilgi ve Bilgisayar Okur Yazarlığı.....	12
2.4 Programlama Öğrenmek Neden Öğrencilere Zor Geliyor.....	14
2.5 Programlama Başarısı Üzerine Yapılan Çalışmalar .....	16
2.6 Programlama Yetenek Testi.....	23
2.7 Mantıksal Düşünme Yeteneği.....	25
2.8 Öğrenme Stilleri .....	26
2.8.1 Kolb Öğrenme Stilleri Modeli .....	28
2.8.2 Kolb'un Deneyimsel Öğrenme Kuramına Dayalı Eğitim .....	29
2.9 Programlama Öz Yeterlilik Seviyesi.....	34
ÜÇÜNCÜ BÖLÜM .....	37
3 Yöntem.....	37
3.1 Araştırma Modeli .....	37

3.2 Çalışma Grubu .....	38
3.3 Veri Toplama Araçları.....	39
3.3.1 Programlama Yetenek Testi .....	39
3.3.2 Mantıksal Düşünme Yetenek Testi .....	39
3.3.3 Programlamaya İlişkin Öz Yeterlilik Ölçeği.....	40
3.3.4 Kolb Öğrenme Stilleri Envanteri III.....	40
3.3.5 Programlama Başarı Testi .....	42
3.4 Veri Toplama Süreci .....	42
3.5 Kullanılan Analiz Yöntemleri .....	43
3.5.1 Korelasyon Analizi.....	43
3.5.2 Regresyon .....	44
3.5.3 İlişkisiz Örneklemeler İçin Tek Yönlü Varyans Analizi.....	45
<b>DÖRDÜNCÜ BÖLÜM .....</b>	<b>46</b>
<b>4    Bulgular .....</b>	<b>46</b>
4.1 Programlama Yetenek Puanı ile Programlama Başarısı Arasında Bir İlişki Var Mıdır ?.....	46
4.2 Mantıksal Düşünme Yeteneği ile Programlama Başarısı Arasında İlişki Var Mıdır? 47	
4.3 Öğrencilerin Programlama Öz Yeterlilik Algıları ile Programlama Başarısı Arasında Bir İlişki Var Mıdır?.....	49
4.4 Öğrenme Stillerine Göre Öğrencilerin Programlama Başarısı Farklılık Göstermekte Midir? .....	49
4.5 Bilgisayar Deneyimi ve Programlama Deneyiminin Programlama Başarısına Bir Etkisi Var Mıdır? .....	52
4.6 Programlama Başarısı Üzerinde Etkisi Olduğu Düşünülen, Programlama Yetenek Puanı, Mantıksal Düşünme Yeteneği, Öğrenme Stilleri ve Programlamaya İlişkin Öz yeterlilik Düzeyi Programlama Başarısını Hangi Öncelikte ve Derecede Açıklamaktadır? .....	53
<b>BEŞİNCİ BÖLÜM .....</b>	<b>55</b>
<b>5    Tartışma ve Sonuç .....</b>	<b>55</b>
5.1 Programlama Yetenek Puanı ile Programlama Başarısı Arasında Bir İlişki Var Mıdır ? Problemine İlişkin Sonuçlar.....	55
5.2 Mantıksal Düşünme Yeteneği ile Programlama Başarısı Arasında İlişki Var Mıdır? Problemine İlişkin Sonuçlar .....	56
5.3 Öğrencilerin Programlama Öz Yeterlilik Algıları ile Programlama Başarısı Arasında Bir İlişki Var Mıdır? Problemine İlişkin Sonuçlar.....	56
5.4 Öğrenme Stillerine Göre Öğrencilerin Programlama Başarısı Farklılık Göstermekte Midir? Problemine İlişkin Sonuçlar .....	57
5.5 Bilgisayar Deneyimi ve Programlama Deneyiminin Programlama Başarısına Bir Etkisi Var Mıdır? Problemine İlişkin Sonuçlar .....	58

<b>5.6 Programlama Başarısı Üzerinde Etkisi Olduğu Düşünülen, Programlama Yetenek Puanı, Mantıksal Düşünme Yeteneği, Öğrenme Stilleri ve Programlamaya İlişkin Öz Yeterlilik Düzeyi Programlama Başarısını Hangi Öncelikte ve Derecede Açıklamaktadır? .....</b>	<b>59</b>
<b>5.7 Öneriler</b>	<b>59</b>
<b>Kaynakça .....</b>	<b>63</b>
<b>Ekler .....</b>	<b>73</b>
<b>Ek 1 Programlama Yetenek Testi .....</b>	<b>73</b>
<b>Ek 2 Mantıksal Düşünme Yeteneği Testi .....</b>	<b>78</b>
<b>Ek 3 Kolb Öğrenme Stilleri Envanteri III.....</b>	<b>86</b>
<b>Ek 4 Programlamaya İlişkin Öz Yeterlilik Ölçeği.....</b>	<b>91</b>



## TABLolar LİSTESİ

<b>Tablo 1</b> Ölçeklerin uygulandıđı öğrenci sayıları .....	39
<b>Tablo 2</b> Programlama yetenek ve programlama başarı testi normallik analizi .....	46
<b>Tablo 3</b> Programlama yetenek puanı ve başarı puanı korelasyon sonuçları .....	47
<b>Tablo 4</b> Mantıksal düşünme yetenek ve programlama başarı testleri normallik analizi .....	48
<b>Tablo 5</b> Mantıksal düşünme yeteneđi ve programlama başarısı arasındaki korelasyon .....	48
<b>Tablo 6</b> Programlama öz yeterliliđi ve programlama başarısı normallik analizi .....	49
<b>Tablo 7</b> Öz yeterlilik ile programlama başarısı arasındaki ilişki.....	49
<b>Tablo 8</b> Öğrenme stilleri gruplarına göre programlama başarısı açıklayıcı istatistikleri.....	50
<b>Tablo 9</b> Öğrenme stilleri grupları ANOVA sonucu .....	51
<b>Tablo 10</b> Öğrenme stilleri gruplarının programlama başarısı ortalaması Scheffe testi sonucu .....	51
<b>Tablo 11</b> Bilgisayar deneyimi, programlama deneyimi ve programlama başarısı verilerinin normallik analizi sonuçları .....	52
<b>Tablo 12</b> Bilgisayar ve programlama deneyim ile programlama başarısı arasındaki korelasyon analizi .....	53
<b>Tablo 13</b> Programlama başarısının yordanmasına ilişkin çoklu regresyon analizi ...	53

## ŞEKİLLER LİSTESİ

Şekil 1 Türkiye imalat sanayisinin küresel değer zincirindeki konumu (TÜSİAD, 2016, s. 34).....	6
Şekil 2 Geleneksel üretim modellerinin Sanayi 4.0 ile değişimi (TÜSİAD, 2016,s. 21).....	7
Şekil 3 Program yazma ve okuma görevleri sonucunda ortaya çıkan ortak örüntü (Özdiñ ve Altun 2014, s. 1539).....	18
Şekil 4 Kolb öğrenme stilleri modelinin iki boyutu (Gencel, 2006, s. 39) .....	29
Şekil 5 Deneysel öğrenme döngüsüne uygun etkinlikler (Gencel, 2006, s. 46) ....	31
Şekil 6 Kolb'un öğrenme stilleri sınıflaması (Gencel, 2006, s. 40).....	32
Şekil 7 KÖSE III koordinat sistemi (Kolb, 1999: 6 Akt. Gencel 2007,s. 132).....	42
Şekil 8 Mantıksal düşünme yetenek testi puan grafiđi .....	47
Şekil 9 PHPStorm tümleşik geliştirme ortamı .....	61



## **SİMGELER VE KISALTMALAR**

BÖTE	Bilgisayar ve Öğretim Teknolojileri Eğitimi
TGO	Tümleşik geliştirme Ortamı
MDYP	Mantıksal Düşünme Yeteneği Puanı
PBP	Programlama Başarı Puanı
PYP	Bilgisayar Programlama Yetenek Puanı
PÖYA	Programlama Öz Yeterlilik Algısı



# BİRİNCİ BÖLÜM

## 1 GİRİŞ

Bilişim teknolojilerinin yoğun olarak kullanıldığı bir çağda yaşıyoruz. Savunma sanayi, eğitim, sağlık, ulaşım gibi neredeyse her alanda bilişim teknolojilerinin imkânlarından yararlanıyoruz. Hayatımızda bu kadar çok yer tutan bu alanı daha iyi anlamamız ve daha etkin kullanmamız gerekmektedir. Bu nedenle bilişim teknolojileri araçlarının en önemlisi olan bilgisayarların nasıl çalıştığını ve nasıl daha etkin kullanabileceğimizi öğrenmemiz ve gelecek nesillere öğretmemiz gerekmektedir.

Bilişim teknolojilerini tanımlamadan önce “Teknoloji nedir?” sorusuna değinelim. Türk Dil Kurumu teknolojiyi “İnsanın maddi çevresini denetlemek ve değiştirmek amacıyla geliştirdiği araç gereçlerle bunlara ilişkin bilgilerin tümü” şeklinde tanımlamıştır. Teknoloji hayatımızdaki etki alanını genişlettikçe teknoloji tanımı da gelişme göstermiştir. Teknolojinin tanımı ilk zamanlarda somut, maddi ürünleri kapsamaktayken zamanla bu ürünlerin oluşum süreci ve sonuçlarını da kapsayacak şekilde güncellenmiştir (Aydın ve Kara, 2013). Deniz, Görgen ve Şeker (2006, s. 63) teknolojiyi “insanların problemlerine yardımcı olacak araçların, makinelerin, materyallerin ve süreçlerin geliştirilmesi ve kullanılması” olarak tanımlamıştır. Benzer şekilde MEB (2006,s. 3) “Teknoloji, insan hayatının kalitesini artırmak amacıyla yaratıcılık ve zekânın; bilim, sanat, mühendislik, ekonomi ve sosyal çalışmayla oluşturulan bir bireşimidir. Herhangi bir şeyi daha iyi, daha hızlı, daha kolay, daha ekonomik ve daha verimli yapma girişimidir” şeklinde bir teknoloji tanımı yapmıştır. Günümüzde teknoloji denildiğinde ilk akla gelen ürünler, bilgisayarlar olmaktadır. Bilgisayarlar denilince de aklımıza ilk olarak evlerde ve iş yerlerinde kullandığımız kişisel bilgisayarlar gelmektedir. Fakat bilgisayarlar bunlardan ibaret değildir, evde kullandığımız bulaşık makinesi, buzdolabı, çamaşır makinesi gibi eşyalarda, toplu ulaşım araçlarındaki kart okuyucularda ve bankamatiklerde farklı türlerde bilgisayarlar kullanılmaktadır. Günlük hayatın her

alanında karşılaştığımız bu ürünlerin ortak amacı, hayatımızı kolaylaştırmak ve daha kaliteli hale getirmektir.

Türk Dil Kurumu'nun tanımına göre bilişim, insanların birçok alanda kullandığı ve bilimin dayanağı olan bilginin özellikle elektronik ortamda düzenli ve akla uygun işlenmesi bilimi olarak tanımlanmaktadır. Bilişim, insanların her gün kullandıkları bilgiye ulaşmalarını sağlayan teknolojileri üretmeyi ve geliştirmeyi amaçlayan bilim dalı olarak tanımlanabilir. Bilişim bilimi, bilgi teknolojileri ile insanlar arasındaki etkileşim üzerine odaklanmıştır. Bilişim hızlı gelişen bir alandır ve bilgisayarları kullanarak gizlilik, güvenlik, yoksulluk, sağlık hizmetleri, eğitim ve çevre sorunları gibi büyük problemlere çözüm arar ("Informatics Defined", t.y). Bilişim, insanların teknoloji kullanarak bilgiye en hızlı ve güvenilir yoldan nasıl ulaşacağı ile ilgilenir. İnsanlar bilgisayar kullanarak bilgiye ulaşmak ve bilgiyi işleyebilmek için bilgisayar yazılımlarına ihtiyaç duyarlar. Bu yazılımları geliştirmek ve daha iyi kullanılmasını sağlamak, insanların bilgiye en kaliteli yoldan ulaştırmak bilişim sektörü çalışanlarının görevidir.

Bir yazılım üretmek, o yazılım için gerekli olan programlama diline hâkim olmayı gerektirmektedir. Programlama dili, bilgisayarlar gibi elektronik cihazları programlamak için kullanılan, belli söz dizim kuralları ve deyimleri olan komutlar bütünüdür. Programlama dillerinin kullanılması sürecine programlama, programlama sonucunda cihazların amaca uygun şekilde çalışmasını sağlayan son ürüne de program ya da yazılım adı verilmektedir. (Ersoy, Madran ve Gülbahar, 2011). İlk yazılımlar makine dili ve delikli kartlar gibi bilişim teknolojisi açısından ilkel yöntemlerle programlanıyordu. Bilişim teknolojilerinin gelişmesi ile birlikte günümüzde konuşma dilinde kullandığımız kelimeleri kullanarak yazılımlar programlanabilmektedir. Günümüzde yazılımlarda kullanılan nesnelere hazır olarak bulunduğu yazılım geliştirme ortamları ile eskisine göre çok daha kolay bir şekilde yazılımlar programlanmaktadır.

Yazılım sektörü doğal kaynakları yetersiz olan ve yeterli istihdam sağlanamayan ülkelerde, ülke ekonomisine katkı sağlayan önemli bir sektördür. 21.yy da Sanayi 4.0 ile birlikte yazılım sektörü gittikçe daha çok önem kazanmaktadır. Yazılım sektörü diğer sektörlerle de katkı sağladığı için, bu sektöre önem verilmesi ülke ekonomisine

ve gelişmişliğine önemli katkılar sağlayacaktır (Yılmaz, 2013). Gelecekte güçlü bir devlet olabilmek için yazılım sektörüne önem verilmesi doğru bir karar olacaktır.

Ülkemizde orta öğretim, ön lisans ve lisans düzeylerinde birçok bölümde programlama eğitimi verilmektedir. Yazılım sektöründe başarılı bir ülke olmanın temel şartı kaliteli ürünler ortaya koyabilecek profesyonel yazılımcıların yetiştirilmesidir. Bu nedenle programlama eğitimine önem verilmeli ve daha kaliteli eğitimler verebilmek için çalışmalar yapılmalıdır.

### **1.1 Problem Durumu**

Günümüzde özel eğitim kurumlarındaki eğitimleri de göz önünde bulundurursak her eğitim kademesinde programlama eğitimi verilmektedir. İlköğretim kademesinde verilen programlama eğitimi öğrencilere bilgisayarca düşünme yeteneği kazandırmak, problem çözme yeteneklerini geliştirmek, yaratıcılıklarını arttırmak ve gelecekte ihtiyaç duyacağı temel bilgileri kazandırmak amacıyla verilmektedir. Ortaöğretim kademesinde bilgisayarda karşılaşılabilecekleri sorunlara çözüm üretebilmelerini, teknolojik cihazları daha etkin kullanabilmelerini sağlamak ve bilişim sektöründe ihtiyaç duyulan alt seviye elemanları yetiştirmek için programlama eğitimi verilmektedir. Yükseköğretim kademesinde ise ülkenin ihtiyaç duyduğu orta ve üst kademe bilişim elemanları yetiştirmek, ülke ekonomisine katkı sağlayacak yeni teknolojiler, yazılımlar ve sistemler üretebilecek elemanlar yetiştirilmek amaçlanmaktadır.

Bilgisayarların ilk üretildiği zamanlarda programlama bilgisi sadece bilgisayar alanında yükseköğretim alan öğrenciler için gerekli görülmekteydi. Bilişim teknolojilerindeki hızlı gelişim sonucunda bilgisayar ve programlama eğitimi giderek alt kademelere doğru genişledi. Kişisel bilgisayar kullanımının yayılması, bilgi ve iletişim teknolojilerindeki hızlı gelişim, toplum yapısında ve ihtiyaçlarında da değişikliklere neden oldu. Bu değişiklikler ile toplum yeniden şekillendi ve “bilgi toplumu” olarak tanımlanan bir toplum yapısı oluştu. Oluşan bu toplumu oluşturan bireylerin sahip olması gereken nitelikler de değişim gösterdi (Akpınar ve Altun, 2014; Yıldırım ve Öner, 2004).

Yapılan çalışmalar programlama eğitiminin öğrenciye, bilgi toplumunda ihtiyacı olan niteliklerin birçoğunu kazandırdığını göstermektedir (Akpınar ve Altun, 2014). Bu

durum, programlama eğitiminin ne denli önemli olduğuna ilişkin büyük bir göstergedir. Bu yüzden programlama eğitiminin küçük yaşlardan itibaren öğretilmesi büyük önem taşımaktadır.

Programlama eğitiminin bireye sağladığı faydalar araştırmacıları bu alanda araştırma yapmaya yönlendirmiştir. Yapılan bu araştırmalar sonucu başlangıç seviyesinde programlama dersi alan birçok öğrencinin bu derste zorlandığı ve başarısız olduğu görülmüştür (Bennedsen ve Caspersen, 2007; Bornat, Dehnadi ve Simon, 2008; Dehnadi ve Bornat, 2006; Jenkins, 2002; Mazman, 2013; Özmen ve Altun, 2014). Bu başarısızlığın kaynağı ve nasıl çözüme ulaştırılacağı üzerine birçok çalışma yapılmış fakat yapılan çalışmalar kanıtlanmış bir sonuca ulaşamamıştır (Meštrović ve Martinčić-Ipšić, 2012). Etkili bir programlama eğitimi verilebilmesi için programlama yeteneğine etki eden faktörlerin de bilinmesi gerekmektedir.

Bu çalışmada konu olarak ele alınan problem durumu, programlama dersinde öğrencilerin başarısız olmaları ve bu başarısızlığa neden olan faktörlerin tam olarak belirlenememiş olmasıdır. Bu problem doğrultusunda öğrencilerin programlama başarılarını etkileyen bazı faktörlerin incelenmesi planlanmıştır.

Çalışmada ele alınacak alt problemler:

1. Programlama yetenek puanı ile programlama başarısı arasında bir ilişki var mıdır?
2. Mantıksal düşünme yeteneği ile programlama başarısı arasında ilişki var mıdır?
3. Öğrencilerin programlama öz yeterlilik algıları ile programlama başarısı arasında bir ilişki var mıdır?
4. Öğrenme stillerine göre öğrencilerin programlama başarısı farklılık göstermekte midir?
5. Bilgisayar deneyimi ve programlama deneyiminin programlama başarısına bir etkisi var mıdır?

6. Programlama başarısı üzerinde etkisi olduğu düşünölen, Programlama yetenek puanı, Mantıksal Düşünme Yeteneđi, Öğrenme Stilleri ve Programlamaya İlişkin Öz yeterlilik düzeyi Programlama başarısını hangi öncelikte ve derecede açıklamaktadır?

## **1.2 Çalışmanın Amacı**

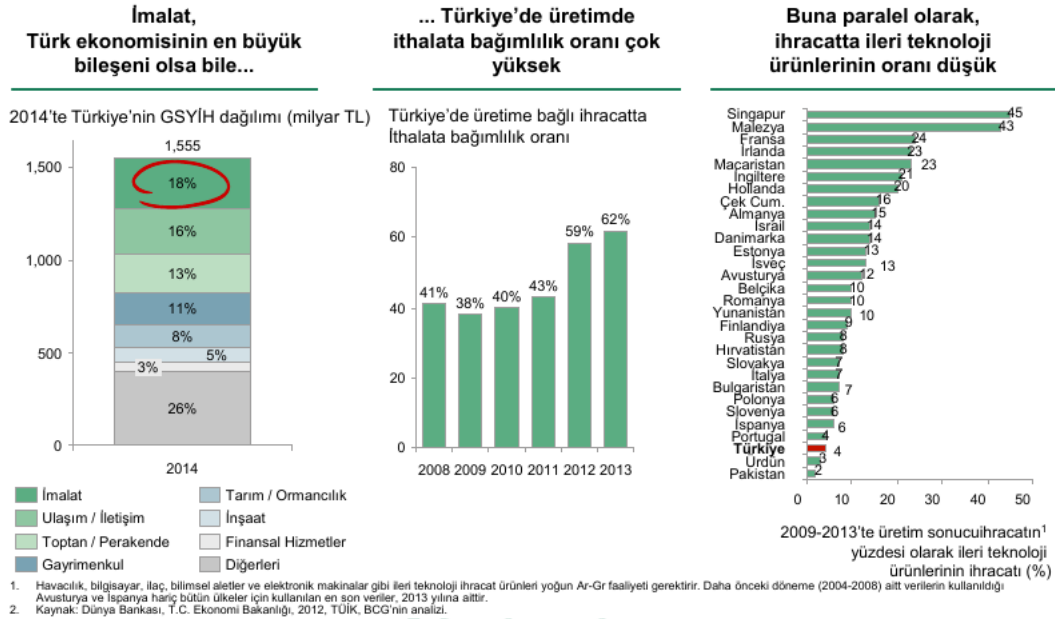
Programlama başarısını etkileyen faktörler üzerine yapılan akademik çalışmalar 1950'li yıllara kadar uzanmaktadır. Bu çalışmaların yöntemleri, ele aldığı değişkenler ve ölçme araçları değişse de ortak amaçları iyi bir programcı olmayı etkileyen faktörleri belirlemektir (Mazman, 2013). Bu çalışmada programlama başarısı ile ilgili yapılan çalışmalara katkı sağlamak ve öğrencilerin programlama derslerinde daha başarılı olmaları için ders işlenişinde ne tür geliştirmelerin yapılabileceđi konusunda önerilerde bulunmak amaçlanmıştır. Çalışmada, Bilgisayar Programlama Yetenek Testi, Mantıksal Düşünme Yetenek Testi, Kolb'un Öğrenme Stilleri Envanteri III, Programlamaya İlişkin Öz Yeterlilik Ölçeđi sonuçları ile programlama dersi başarı puanları arasındaki ilişkiler incelenecektir.

## **1.3 Çalışmanın Önemi**

Günümüzde insansız uçak, otonom araçlar, akıllı telefon, akıllı ev, akıllı fabrika gibi birçok kavram kullanılmaktadır. Bu kavramların hepsinin arkasında gelişen bilişim teknolojileri ve yazılımlar bulunmaktadır. Bu sistemler savunma sanayi, eğitim, sağlık, ulaşım, beslenme gibi birçok alanında etkili bir şekilde kullanılıyorlar ve giderek yaygınlaşıyorlar. Hayatımızda bu kadar geniş yer tutan bu araçları kendi ihtiyaçlarımıza göre geliştirebilmek ve yeni ürünler ortaya çıkartabilmek için bu ürünlerin arkasında bulunan yazılım gücüne hâkim olmamız gerekmektedir. Bu teknolojilerin arkasında bulunan yazılım bilgisini geleceđimiz olan çocuklarımıza ve gençlerimize programlama eğitimi ile aktarmalıyız. Etkili bir şekilde verilen eğitim sonucu programlama bilgisi ile donatılmış olan yeni nesiller geliştirecekleri yazılım ve teknolojilerle ülkemizi daha ileri teknolojik seviyelere taşıyacak ve daha refah bir toplum düzeyine ulaşmamızı sağlayacaklardır.

Günümüzde bilişim teknolojilerini etkin bir şekilde kullanan toplumlar bilgi toplumu olarak adlandırılmaktadır. Akpınar'a (1999) göre bilgi toplumunda bireyler plan yapabilme, olayları yorumlayabilme, yeni bilgiler oluşturup karşılaştıkları sorunların

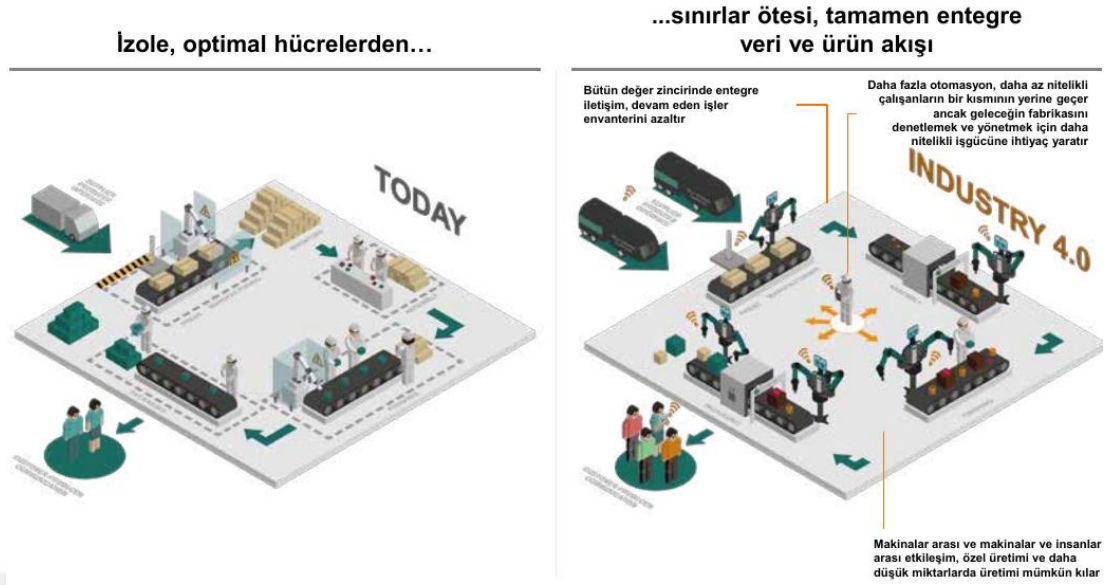
çözümünde kullanabilme, teknik sorunlara çözüm bulabilme gibi niteliklere sahip olması gerekmektedir (Akt. Gündüz ve Odabaşı, 2004). Yapılan çalışmalar programlama eğitiminin öğrenciye bilgi toplumunda ihtiyacı olan niteliklerin birçoğunu kazandırdığını göstermektedir (Akpınar ve Altun, 2014). Bu durum programlama eğitiminin geleceğimiz için önemli olduğunu göstermektedir.



Şekil 1 Türkiye imalat sanayisinin küresel değer zincirindeki konumu (TÜSİAD, 2016, s. 34)

Geçmiş dönemlerde sanayi devrimine öncülük eden toplumlar daha sonraları diğer toplumlara üstünlük sağlamışlardır. Günümüzde ise, üretim zincirinde bulunan tüm unsurların bilgi - iletişim teknolojileri ile birbirine bağlı çalıştığı bir üretim modeli vizyonu olarak tanımlanan, Sanayi 4.0 devrimine adım adım ilerlemekteyiz. Türkiye ekonomisinde imalat sektörü en büyük paya sahip olmasına rağmen üretimde teknoloji kullanımı sıralamasında birçok ülkenin gerisinde yer almaktadır (Şekil 1). Bu durum Sanayi 4.0 devrimini yakalamak için teknolojik gelişmelere daha fazla önem vermemiz gerektiğinin bir göstergesidir.

Sanayi 4.0 ile fabrikalarda insanların yerini robotlar almaya başlayacak. Bu durum az nitelikli eleman ihtiyacını azaltırken, robotların denetimini ve kontrolünü sağlayabilecek yüksek nitelikli eleman ihtiyacını arttıracaktır (Şekil 2).



Şekil 2 Geleneksel üretim modellerinin Sanayi 4.0 ile değişimi (TÜSİAD, 2016,s. 21)

Sanayi devriminde öncü topluluklar arasına girememenin zararlarını hissettiğimiz bu günlerde Sanayi 4.0 dönemini kaçırmamak için bilişim teknolojilerine ve yazılım eğitimine önem ve öncelik vermemiz gerekmektedir.

Sanayi 4.0 da ihtiyacımız olacak olan daha nitelikli iş gücüne sahip olabilmek için verilmekte olan eğitimi daha nitelikli bir hale getirmek zorundayız. Ülkemizin geleceği için Bilişim Teknolojileri eğitimi ile birlikte programlama eğitimi yaygınlaştırmalı ve daha kaliteli bir eğitim sunmalıyız.

Çalışmada Bilgisayar ve Öğretim Teknolojileri Eğitimi öğrencilerinin aldığı programlama dersi incelenecektir. BÖTE bölümü mezunları geleceğin programlama eğitimcileri olduğu için onların aldığı programlama eğitiminin kalitesi doğrudan kendi verecekleri programlama eğitimi etkileyecektir. Ülkemizde ilköğretim seviyesinde okutulan Bilişim Teknolojileri ve Yazılım dersi ve ortaöğretim seviyesinde okutulan Bilgisayar Bilimi derslerini BÖTE bölümü mezunları vermektedir. Bu nedenden dolayı BÖTE bölümü öğrencilerinin aldığı programlama eğitimi oldukça önemlidir. Ülkemizde programlama sektörüne eleman yetiştiren yükseköğretim kurumlarındaki öğrenci sayıları incelendiğinde tüm öğrencilerin %78'i Meslek Yüksek Okulu, %22 si ise Bilgisayar ve Yazılım Mühendisliği bölümlerinde okuduğu görülmektedir (Yılmaz, 2013). BÖTE bölümü mezunlarının

%78 'lik dilime sahip MYO 'larda programlama eğitimi veren Öğretim üyesi kadrolarında görev aldığı düşünüldüğünde BÖTE mezunlarının önemi açıkça görülmektedir.

Bu çalışma sonucunda programlama eğitimimizdeki aksaklıklar giderilmeye çalışılacak ve daha kaliteli bir programlama eğitimi için yapılması gerekenler belirlenecektir. Eğitim ne kadar kaliteli ve başarılı olursa bu eğitimi tamamlamış bireylerde o düzeyde kaliteli ve başarılı işlere imza atacaktır. Bu nedenle programlama eğitimi başarısı üzerinde çalışmalar yaparak eğitim sistemimizi daha üst seviyelere çıkartmak için katkı sağlanmaya çalışılacaktır.

#### **1.4 Varsayımlar**

Çalışmada aşağıda listelenen varsayımlar kabul edilmiştir:

- Çalışmaya katılan katılımcıların uygulanan ölçeklerdeki sorulara içtenlikle ve gerçeği yansıtacak şekilde cevap verdikleri
- Ölçeklerin ölçtükleri değeri doğru ölçtüğü
- Araştırmada seçilen örneklemin evreni temsil ettiği

#### **1.5 Sınırlılıklar**

Bu çalışma verileri Ondokuz Mayıs Üniversitesi Eğitim Fakültesi Bilgisayar ve Öğretim Teknolojileri Eğitimi (BÖTE) Bölümü Programlama dilleri dersini alan öğrencilerden çalışmaya katılanlar ile sınırlıdır.



## İKİNCİ BÖLÜM

### 2 KURAMSAL ÇERÇEVE VE İLGİLİ ARAŞTIRMALAR

#### 2.1 Programlama Nedir

Elektronik cihazlar hayatımızın her alanında kullandığımız cihazlardır. Elektronik cihazların işlerini yapabilmesi için öncesinde programlanmaları gerekmektedir. Geçmişte yapılan tanımlara göre, bir makinenin çalışmaya hazırlanması iki aşamadan oluşur birincisi “programlama” ikincisi ise “kodlama”dır. Programlama makinenin çalışma amacına uygun olacak şekilde yapacağı işlemlerin planlanması ve görselleştirmesini yani algoritma oluşturma sürecini kapsayan bir süreçtir. Kodlama ise yapılan programlamanın (algoritmanın) bilgisayar diline aktarılması sürecidir (Blackwell, 2002). Günümüzde programlama kavramı bu iki süreci de kapsayacak şekilde kullanılmaktadır. Yani programlama amaca uygun bir algoritma oluşturup programlama dilleri yardımı ile oluşturulan algoritmanın bilgisayar diline aktarılması sürecidir. Programlama dili elektronik cihazları programlamak için kullanılan, belli söz dizim kuralları ve deyimleri olan komutlar bütünüdür. Programlama dilleri ile yazılan bu komutlar elektronik cihazların ne yapması gerektiğini belirleyen yazılımları oluşturur (Ersoy ve diğerleri, 2011). Programlama dilleri bilgisayarların ilk üretildiği günden buyana var olmuşlardır. Programlama dilinin oluşturan komutlar ilk zamanlarda 1 ve 0 dizilerinden oluşmaktaydı, teknolojinin geliştirilmesi ile birlikte artık günlük hayatta kullandığımız kelimeleri kullanarak bilgisayarlara komutlar verebildiğimiz programlama dilleri geliştirilmiştir. Bilgisayarlar programlama dillerini kullanarak yazdığımız komutları kendi işleyecekleri şekil olan 1 ve 0 dan oluşan diziler haline getirerek yapılması gereken işlemi yerine getirirler. Programlama dilleri ile düşündüğümüzü ve yapmak istediğimizi bilgisayarlara aktararak bilgisayarların bizim istediğimiz şekilde çalışmasını sağlayabiliriz. Bilgisayarların hayatımıza kattığı değer, onların çalışmasını sağlayan yazılımların kalitesi ile orantılıdır. Bu yazılımlar ne kadar iyi programlanırlarsa bize sağladıkları katkı o derecede iyi olacaktır.

İlk zamanlarda sadece bilgisayar üzerinde yüksek eğitim alanların bildiği programlama, zamanla bilgisayarların gelişmesi ile birlikte her eğitim seviyesindeki kişilerin bildiği veya bilmesi gerektiği bir bilgi durumuna gelmiştir. Günümüzde

programlama dilleri çocukların bile kavrayabileceği bir seviyeye ulaşmıştır. Evimizde, iş yerimizde hatta cebimizde kullandığımız bilgisayarların kullanım alanlarının yaygınlaşması programlama bilgisinin daha geniş bir kitleye yayılmasına olanak sağlamıştır. Gelişmiş ülkelerde neredeyse herkes bir akıllı telefona (bilgisayara) sahip durumda. Bilgisayarın bu kadar yayılmasının en büyük nedeni donanımın küçülmesi ve kullanım alanlarına uygun olacak şekilde yazılım çeşitlerinin artması olarak görülmektedir (Karabak ve Güneş, 2013).

Programla süreci ilk olarak çözülmesi planlanan problemin tanımlanması ve iyice kavranması ile başlar. İkinci adım, problemin çözüm yollarını belirlemek ve çözüm yoluna uygun algoritma hazırlanmasıdır. Üçüncü adım, seçilen programlama dili ile programın kodlanmasıdır. Son adım olarak da programın derlenip varsa hatalarının tespit edilip giderilmesidir (Casey, 1997; Keskinsoy, 2010; Yiğit, 2016). Bu adımları tamamlayabilmek için kişi birden fazla bilgiye sahip olmalıdır. Programlama bilgisi üç çeşit bilgiye sahip olmayı gerektirir. Bunlar, (1) Sözdizimsel (Syntactic) bilgi, (2) Kavramsal (Conceptual) bilgi, (3) Stratejik (Strategic) bilgidir.

*Sözdizimsel (Syntactic) Bilgi:* Derleyicilerin programlama dilini makine diline çevirebilmesi için yazılan programların belli kurallara uygun şekilde yazılması gerekmektedir. Bir değişkenin tanımının nasıl yapılacağı, fonksiyonları nasıl tanımlanacağı, satır sonlarına noktalı virgül konulup konulmayacağı gibi bilgiler sözdizimsel bilgilerdir (Bayman ve Mayer, 1988; Yiğit, 2016).

*Kavramsal (Conceptual) Bilgi:* Programlama dilinde kullanılan komutların yaptığı işlemlerin bilgisine sahip olmak olarak tanımlanabilir. Programlama dilleri genel olarak aynı kavramları kullanırlar, değişken kullanma, fonksiyon oluşturma, döngü ve karar yapıları bütün programlama dilinde vardır. Programlama yapacak bireyin bu kavramları iyi kavramış olması gerekmektedir.

*Stratejik (Strategic) Bilgi:* Sözdizimsel bilgi ve kavramsal bilgiyi bilmek bir program yazmak için yeterli değildir. Bu bilgilerin doğru bir strateji ile ne şekilde kullanılacağı bilgisine de sahip olmak gerekmektedir. Problem çözme aşamasında doğru algoritmayı oluşturmak stratejik bilgi gerektirir (Bayman ve Mayer, 1988; Yiğit, 2016).

Programlamayı kitap yazmaya benzetirsek, sözdizimsel bilgi yazım kurallarını bilmeyi, kavramsal bilgi, kullanılan kelimelerin anlamlarını bilmeyi, stratejik bilgi ise kitabın giriş gelişme ve sonuç bölümlerini uygun bir şekilde belirleyebilmeyi temsil eder.

## 2.2 Neden Programlama Öğrenmeliyiz

Alınan bilgisayar programlama eğitimi ile yazılan programlar, günlük hayatta işimizi kolaylaştırmanın yanı sıra zihinsel yapımıza da olumlu katkılar sağlamaktadır. Birçok ülkede olduğu gibi ülkemizde de programlama eğitimi ağırlıklı olarak lisans düzeyinde verilmektedir. Son zamanlarda, yazılımı sevdirmek, programlamanın sağladığı faydaları çocuklara erken yaşta kazandırmak ve sonraki yıllarda alacağı programlama eğitimine ön hazırlık olması için ilk ve orta öğretim düzeyinde de programlama eğitimi verilmeye başlanmıştır (Karabak ve Güneş, 2013).

Bilgisayar programlamanın özellikle küçük yaştaki çocukların bilişsel gelişimine yaptığı katkı üzerine yapılan araştırmalar 1980'li yıllara kadar uzanmaktadır (Akpınar ve Altun, 2014). Programlama eğitimi çocuklara, kural öğrenme (Gorman ve Bourne, 1983), yansıtıcılık ve farklı düşünce (yaratıcılık) yetileri ile üst biliş yetenekleri ve yön tarif etme yeteneği (Clements ve Gullo, 1984) gibi yetenekler kazandırmaktadır. Programlama büyük bir problemi alt parçalara ayırarak genellenebilir bir çözüm üretme sürecini barındığı için öğrenciye problem çözme analiz yeteneği kazandırır (Saedi, Perrenet, Jochems ve Zwaneveld, 2011). Programlama eğitimi diğer disiplinler ile birlikte planlandığında diğer disiplinlerin daha etkili öğretilmesine olanak sağlayacaktır. Örneğin matematik dersinde tek ve çift sayıları öğrenen bir çocuk, LEGO kullanarak hazırladığı bir aracı bir döngü oluşturarak her döngü sonunda yönünü tersine çevirecek şekilde programlıyor. Çocuk denemelerinde döngü sayısına çift sayılar ve tek sayılar vererek sonucu gözlemliyor. Bu gözlem sonucunda şu cümleleri kuruyor:

Döngüde çift sayıları kullandığımda araç harekete başladığı yerde duruyor. Tek sayı kullandığımda ise başladığı yerden uzakta duruyor (Resnick ve Ocko, 1990, s. 6).

Ve sonrasında ekliyor:

Buda çift ve tek sayıların neden var olduğunu gösteriyor (Resnick ve Ocko, 1990, s. 6).

Daha önce matematik dersinde öğrenilen çift ve tek sayıları kullanarak yapılan bu etkinlik sonucunda öğrenci kendi deneyimleri ile çift ve tek sayılara yeni bir bakış açısı kazanmış oluyor (Resnick ve Ocko, 1990). Ayrıca öğrenciler birlikte programlar yazarak, yaptıkları projeleri birbiri ile paylaşarak işbirlikçi öğrenme yeteneklerini geliştirebilirler (Demirer ve Sak, 2016; Resnick, 2013).

Teknolojik gelişmelerle birlikte günlük hayatta yaptığımız birçok işlem giderek daha da hızlı bir şekilde sonuca ulaşmaktadır. Programlama bilgisi günlük hayatta sık sık yapmamız gereken ve her seferinde aynı işlemleri yaptığımız işlerimizi kolaylaştırmamıza imkân sağlar. Kendi ufak programlarımızı yazarak bilgisayar işlemlerimizi otomatikleştirebilir ve kendimize zaman kazandırabiliriz. Yazacağımız ufak bir programı çalıştırarak 5 dk. sürebilecek bir işi saniyeler içerisinde halletmemiz mümkün.

İçinde yaşadığımız bilgi toplumunda bireylerin plan yapabilme, olayları yorumlayabilme, yeni bilgiler oluşturup karşılaştıkları sorunların çözümünde kullanabilme, teknik sorunlara çözüm bulabilme gibi niteliklere sahip olunması gerekmektedir (Gündüz ve Odabaşı, 2004) Yapılan çalışmalar programlama eğitimi ile bu niteliklerin öğrencilere kolaylıkla kazandırılabilceğini göstermektedir (Akpınar ve Altun, 2014). Genç neslin dijital yerli olarak tanımlandığı 21.yy da bilgi ve bilgisayar okuryazarlığı kavramlarının oldukça önem kazandığı görülmektedir.

### **2.3 Bilgi ve Bilgisayar Okur Yazarlığı**

Kellner'e (2000) göre okuryazarlık toplum tarafından oluşturulan ve sık kullanılan iletişim simgelerinin etkili bir şekilde kullanma becerisidir. Okuryazar olan insan toplumun ürettiği bilgiye kolay ulaşır ve o bilgiyi kullanarak yeni bilgiler ve ürünler üretebilir. Günümüzde birbirinden farklı alanlarda birçok okuryazarlık terimi bulunmaktadır. Bilgi okuryazarlığı ve bilgisayar okuryazarlığı bu terimlerden yaygın olarak kullanılanlarındandır.

Bilgi okuryazarlığı günümüzde sıkça karşılaştığımız bir kavram haline gelmiştir. Son zamanlarda yayımlanan birçok akademik çalışmada bilgi toplumu ve bilgi

okuryazarlığı kavramlarına sık sık rastlanılmaktadır. Bilgi okuryazarlığı kavramını daha iyi anlayabilmek için insanoğlunun tarihsel gelişimine bir göz atmak gerekir. Tarihsel süreç içerisinde insanlar sanayi devrimi gerçekleşmeden önce daha çok tarım ile ilgileniyorlardı. Bu dönemlerde okuryazarlık yeteri kadar yaygın değildi. Zaman geçtikçe okuryazarlığa verilen önem arttı ve daha çok insan bilgiye ulaşır ve bilgi üretir hale geldi. Üretilen bu bilgi peşinden sanayi devrimini getirdi. Sanayi devrimini yaşayan insanlar tarım toplumu insanlarını geride bırakarak, teknolojisi, eğitimi ve ekonomisi tamamen farklı yeni bir toplum yapısı oluşturdular. Sanayi devrimi ile teknolojiye yaşanan gelişmeler ile bilginin üretilmesi ve paylaşılması son derece kolaylaştı, bu imkanlarla artan bilgi birikimi yeni bir toplum yapısı olan bilgi toplumunu oluşturdu (Altun, 2003; Kurbanoglu ve Akkoyunlu, 2001; Yıldırım ve Öner, 2004). İçinde yaşadığımız bilgi toplumu, iş yaşamında, okulda ve kişisel yaşamda başarıya ulaşmak için iyi bir bilgi okuryazarı olmayı gerektirmektedir. Bilgi okuryazarı olmak, bilgiyi bulma, kullanma ve iletme becerilerine sahip olmayı gerektirir (Kurbanoglu ve Akkoyunlu, 2001). Sekizinci Beş Yıllık Kalkınma Planı Bilişim Teknolojileri ve Politikaları Özel İhtisas Komisyonu Raporu'nda (2001) bilgi okuryazarlığı, gerekirse bilgisayar ve ağları kullanarak basılı ve elektronik bilgi kaynaklarından ihtiyacı olana ulaşabilme yeteneği olarak tanımlanmıştır. İyi bir bilgi okuryazarı olabilmek için iyi bir bilgisayar kullanıcısı yani bilgisayar okuryazarı da olmak gerekmektedir.

Yazıcı'ya (2006) göre bilgisayar okuryazarlığı, kişinin temel bilgisayar bilgilerini bilmesi, bilgisayar ile günlük hayatta yaşam kalitesini arttırabilmek, bilgiye ulaşma ve eğlence amacıyla bilgisayar kullanabilmek, bilişim teknolojilerini belli bir seviyede takip edebilmek ve kendi için uygun bilişim teknolojilerini seçebilmek gibi yeteneklere sahip olması olarak tanımlanabilir. Yani kısaca bilgisayar kullanabilme becerisidir. Bilgisayar okuryazarlığı ömür boyu süren bir süreçtir ve kişinin bilgisayar konusundaki deneyimi arttıkça bilgisayar okuryazarlık seviyesi de artacaktır (Kılınç ve Salman, 2006).

Bilgisayar okuryazarlığı programlama becerisini de içinde barındırır. Bilgisayar okuryazarı olan bir kişi temel seviyede programlama bilgisine sahip olmalıdır. Ülkemizde ilkokul seviyesinden başlayarak öğrencilere bilgisayar okuryazarlığı ile ilgili dersler verilmektedir. Fakat bilgisayar okuryazarlığı için oldukça önemli olan

programlama bilgisi yükseköğretim seviyesinin altında sadece meslek liselerinin bilişim teknolojileri bölümünde verilmektedir.

#### **2.4 Programlama Öğrenmek Neden Öğrencilere Zor Geliyor**

Programlama eğitimi ile ilgili birçok akademik çalışma öğrencilerin programlama dersini zor olarak tanımladıklarını vurgulamaktadır (Başer, 2013; Bennedsen ve Caspersen, 2007; Bornat ve diğerleri, 2008; Dehnadi ve Bornat, 2006; Jenkins, 2002; Mazman, 2013; Özmen ve Altun, 2014). Bu ön kabul başarısızlığa ve kalitenin düşmesine neden olmaktadır. Günümüzde birçok yazılım şirketi yazılımcılarının yeterli bilgiye sahip olmaması nedeniyle başarısız olmaktadır. Bu durum yazılım sektörünün önem kazandığı geleceğimizde, programlama eğitimine daha çok önem vermemiz gerektiğinin bir göstergesi niteliğindedir (Bosse ve Gerosa, 2017).

Programlama süreci soyut bir süreç olduğu için birçok öğrenci programlamayı anlamakta güçlük çekmektedir. Anlamakta zorluk çektikleri için ve programlama süreci uzun süre bilgisayar başında zaman harcanmasını gerektirdiğinden dolayı can sıkıcı bir durum oluşturmaktadır. Öğrencilerin soyut programlama kavramlarını anlamakta güçlük çekmeleri ve buna ek olarak bilgisayar başında uzun, sıkıcı vakitler geçirmeleri, programlama öz yeterliliklerini ve tutumlarını olumsuz olarak etkilemektedir (Başer, 2013; Ersoy ve diğerleri, 2011; Özyurt ve Özyurt, 2016). Ayrıca programlama eğitiminin kariyerlerinde kendilerine fayda sağlamayacağını düşünen öğrencilerde programlamaya karşı olumsuz tutumlar görülmektedir (Ersoy ve diğerleri, 2011; Thomas, Ratcliffe, Woodbury ve Jarman, 2002). Bu tip durumlar öğrencilerin motivasyonunu olumsuz etkilemekte ve derse olan ilgiyi azaltarak başarısız olunmasına neden olmaktadır. Jenkins (2002)'e göre programlama dersinin üniversitenin üçüncü ya da daha sonraki dönemlerde verilmesi, ilk dönemlerde verilmesine göre daha doğru olacaktır. Üniversiteye yeni başlamış, ailesinden yeni ayrılmış ve tek başına yeni bir hayata uyum sağlamaya çalışan öğrencilerin, bu sıkıntılarının üstüne bir de programlama dersi eklendiğinde programlama dersindeki başarısızlık daha da artmaktadır.

Programlama başarısına etki eden en önemli etkenlerden biri de acemi programcılar için seçilen ilk programlama dilinin belirlenmesidir. Programlama dilinin piyasada yaygın olarak kullanılıyor olması o programlama dilini seçmek için doğru bir

argüman olmayacaktır (Jenkins, 2002; Mannila, Peltomäki ve Salakoski, 2006). Programlama dili seçimi yapılırken basitlik, okunabilirlik, yazılabilirlik, standartlık, kaynak zenginliği ve maliyet gibi argümanlara göre seçim yapılması daha doğru olacaktır (Brusilovsky, Calabrese, Hvorecky, Kouchnirenko ve Miller, 1997; Güvenir ve Yazıcı, 2015)

Bosse ve Gerosa (2017) 'ya göre öğrenciler en çok, yazdıkları programlarda karşılına çıkan söz dizimi (Syntax) hatalarından şikâyet ediyorlar. İkinci sırada ise programlarında kullanacakları değişkenleri belirlemede sıkıntı yaşadıklarını belirtiyorlar. Öğrenciler hangi değişken türünü (Örn. Float, Integer) ne zaman kullanacaklarını belirleyemediklerinden şikâyet ediyorlar. Üçüncü sırada ise öğrencilerin programda bir hata oluştuğunda karşılına çıkan hata mesajlarını anlamadıklarından ve kafalarının karıştığından şikâyet ettiği belirtilmiştir. Benzer şekilde Özmen ve Altun (2014) üniversite öğrencileri ile yaptığı çalışmada programlama sürecinde karşılaşılan zorluklar (i) Söz dizimi,(ii) Programlama dili ile ilgili kavramlar ve ilkelerin bilinmesi, (iii) Fonksiyonların ve parametrelerinin hatırlanması, (iv) Değişkenlerin belirlenmesi ve kullanılacak döngü ve karar yapılarına karar verilmesi şeklinde sıralamıştır.

Programlama yapabilmenin ön şartı iyi bir algoritma oluşturmaktır. Eğer programlama öğretmek için seçilmiş olan programlama dili katı kurallara sahipse, yazım ortamı öğrenciye anlık destek sağlamıyor ve yapılan yazım hatalarını anlık olarak göstermiyorsa öğrenci kodlama sürecindeki zorluklardan dolayı algoritmaya odaklanamayabilir. Bu tip olumsuzluklar karşısında öğrenci problemi analiz etmekten çok kodları ezberlemek ve hatırlamaya çalışmak ile vakit harcar, sonuç olarak öğrencinin motivasyonu azalır ve bu durum programlama başarısını olumsuz olarak etkiler (Ersoy ve diğerleri, 2011).

Programlama eğitimde tekrar ve pratik yapmak oldukça önemlidir. Derste işlenen konu ders sonrasında ufak projeler geliştirilerek tekrar edilmelidir. Özyurt ve Özyurt (2016)'un yaptığı çalışmada öğrenciler yeterince pratik yapmadıklarını ve bu yüzden programlamadan başarısız olduklarını belirtmişlerdir.

Programla gibi soyut bir dersi her öğrenci aynı hızda anlayamaya bilmektedir. Bu nedenle derste işlenen konuyu, ders sonrasında tekrar edeceği bir kaynak bulamayan

öğrenciler dersten geri kalıp dersten soğumaktadır. Bu durum öğrencinin dersten başarısız olmasına neden olabilmektedir (Jenkins, 2002).

Programlama dersi anlatımında kullanılan araç ve yöntemler de programlama başarısı üzerinde etkilidir (Ersoy ve diğerleri, 2011). Programlama dersindeki başarısızlığa çözüm üretebilmek için farklı araç ve yöntemler geliştirilmiştir. Jiau, Chen ve Su (2009) programlama eğitimini somutlaştırmak için araçlar geliştirmişlerdir. Rajala ve diğerleri (2008) java kullanarak VILLE isminde görselleştirme aracı geliştirmişlerdir. Arabacıoğlu, Bülbül ve Filiz (2007) programlama mantığının öğretilmesini kolaylaştırmak için görsel ve Türkçe bir uygulama dili ve uygulama geliştirmişlerdir.

## **2.5 Programlama Başarısı Üzerine Yapılan Çalışmalar**

Hangi özelliklere sahip bireylerin programlama performansının daha iyi olduğu ve programlama başarısını arttıran uygulamaların neler olduğu konusu yıllardır araştırmacıların dikkatini çekmiştir. Programlama performansını etkileyen iç ve dış faktörler üzerinde yapılan araştırmalarda birbirinden farklı sonuçlara ulaşılmıştır. Programlama başarısı üzerine yapılan çalışmalar 1950'li yıllarda başlamış ve günümüzde hala devam etmektedir. Bu çalışmalarda kullanılan ölçme araçları ve yöntemler değişse de temelde ele alınan konu, iyi bir programcı olmayı etkileyen faktörlerin tespit edilmesi olmuştur (Mazman, 2013). Bu zamana kadar yapılan çalışmalarda daha çok, demografik bilgiler (cinsiyet, yaş), geçmiş akademik başarı, bilgisayar deneyimi, öğrenme stilleri, matematik başarısı ve yeteneği, öz yeterlilik algısı, programlama deneyimi, problem çözme yeteneği, kişilik özellikleri, bilgisayara karşı tutum ve programlama yetenek puanı gibi faktörler incelenmiştir.

Yiğit (2016), yaptığı çalışmada programlama eğitiminde görsel programlama araçlarının kullanılmasının öğrencilerin programlama başarısına ve tutumuna nasıl etki ettiğini araştırmıştır. Bilgisayar ve Öğretim Teknolojileri Eğitimi bölümünde eğitim gören 42 ikinci sınıf öğrencisinin katıldığı çalışma sonucunda, görsel programlama araçları ile öğrenim gören öğrencilerin geleneksel programlama yaklaşımı ile öğrenim gören öğrencilerden daha başarılı olduğu ve programlamaya karşı tutumlarının daha olumlu olduğu tespit edilmiştir.

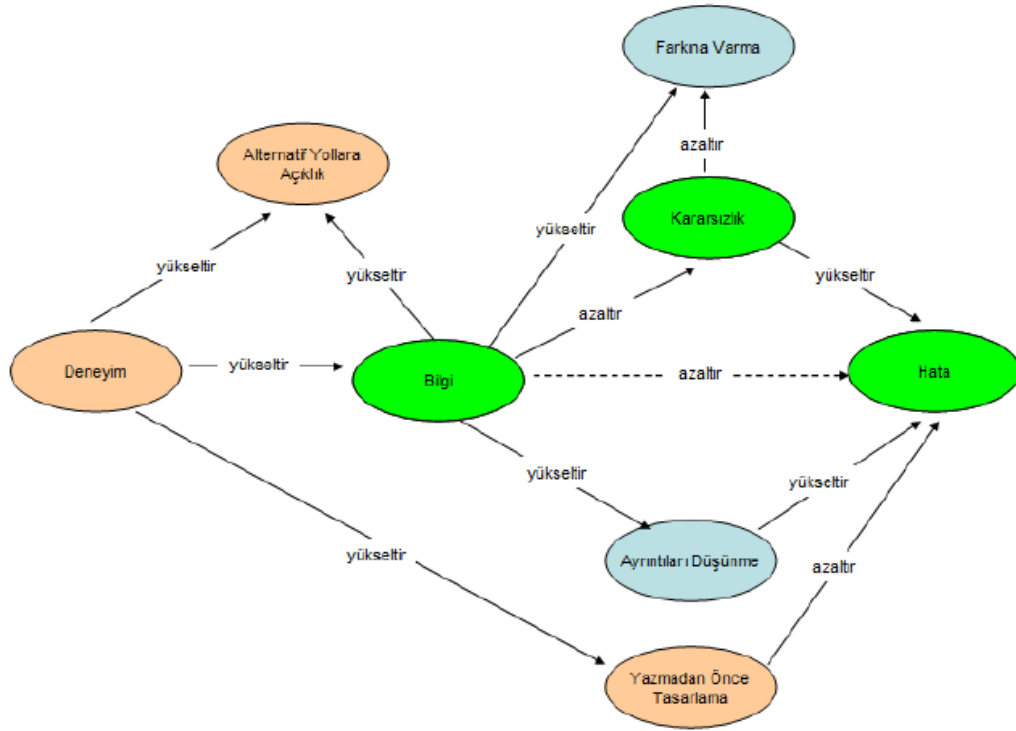


Demir (2015), programlama öğretiminin önündeki zorlukların aşılmasına katkı sağlamak amacıyla, eğitsel programlama dili, yani Scratch ve benzeri programlama dillerinin, Bilgisayar Programlama bölümü programlama dersinde kullanımının öğrencilerin akademik başarısına, performansına ve bilgisayar programlamaya yönelik kaygısına olan etkisini incelemiştir. Kontrol grupsuz deneysel yöntemi kullanılan çalışmada 87 öğrenci rastgele örneklem yöntemi ile üç gruba ayrılmıştır. Eğitsel programlama dili birinci grupta dersin uygulama kısmına, ikinci grupta dersin teorik kısmına, üçüncü grupta ise her iki kısma da entegre edilmiştir. Çalışma sonucunda elde edilen bilgilere göre eğitsel programlama dilinin hem uygulama hem de teorik kısmına entegre edilmesi öğrencilerin akademik başarılarını, kaygılarını ve performansını olumlu yönde etkilediği görülmüştür.

Sebetci ve Aksu (2014), öğrencilerin mantıksal düşünme ve analitik düşünme yeteneklerinin programlama başarısına etkisini incelemiştir. Çalışmaya 142 meslek yüksekokulu öğrencisi katılmıştır. Öğrencilerin mantıksal düşünme yeteneklerini ölçmek için Mantıksal Düşünme Yeteneği Testi, analitik düşünme yeteneklerini ölçmek için de Bütüncül ve Analitik Düşünme Ölçeği kullanılmıştır. Çalışma sonuçlarına göre analitik düşünme ile programlama başarısı arasında yüksek ve pozitif yönde bir ilişki bulunurken, mantıksal düşünme yeteneği ile programlama başarısı arasında orta düzeyde ve pozitif bir ilişki bulunmuştur. Ayrıca analitik düşünme ve mantıksal düşünme yetenekleri ikisi birden incelendiğinde programlama başarısı varyansının %54'ünü açıkladığı sonucuna ulaşılmıştır.

Özdiñç ve Altun (2014), programlama başarısını etkileyen faktörleri incelerken sadece süreç sonundaki testlerin kullanılmasının yeterli olmayacağını ve programlama sürecinin de incelenmesi gerektiğini vurgulamışlardır. Bu doğrultuda yaptıkları çalışmada bilişim teknolojileri öğretmen adaylarının programlama süreçlerinin izlemek ve programlama performansını etkileyen faktörleri ve bu faktörler arasındaki ilişkiyi incelemeyi amaçlamışlardır. Çalışmayı bilişsel görev analizi yöntemi kullanarak sürdürmüşlerdir. Çalışmaya İnternet Tabanlı Programlama dersine devam eden 6 bilişim teknolojileri öğretmen adayı katılmıştır. Çalışma, program yazma ve program okuma olmak üzere iki görevin katılımcılar tarafından yapılaş süreci sesli düşünme yöntemi kullanılarak incelenmesi şeklinde

yapılmıştır. Çalışma sonucunda elde edilen nitel verilerin analizi sonucu Şekil 3 de gösterilen örüntü elde edilmiştir.



Şekil 3 Program yazma ve okuma görevleri sonucunda ortaya çıkan ortak örüntü  
(Özdiñ ve Altun 2014, s. 1539)

Solmaz (2014) Bilgisayar ve Öğretim Teknolojileri Eğitimi öğrencisi olan 39 kişinin katılımı ile yaptığı çalışmasında Alice üç boyutlu yazılım geliştirme ortamının öğrencilerin eleştirel düşünme eğilimleri, problem çözme becerileri, ders başarıları ve üst bilişsel farkındalık düzeyleri üzerindeki etkileri incelemiştir. Çalışma Programlama Temelleri I dersini alan öğrencilerin, birinci şubesinin deney grubu, ikinci şubesinin de kontrol grubu olarak ayırarak oluşturulan gruplar ile ön test - son test kontrol gruplu desen kullanılarak sürdürülmüştür. Deney grubuna Alice uygulaması kullanılarak PHP programlama dili anlatılırken, kontrol grubunda herhangi bir yardımcı araç kullanılmadan PHP programlama dili anlatılmıştır. Çalışmada elde edilen verilere göre Alice üç boyutlu yazılım geliştirme ortamının PHP programlama dili öğretimine, eleştirel düşünme eğilimine, problem çözme becerisine ve üst bilişsel farkındalık düzeylerine anlamlı bir katkı yapmadığı sonucuna ulaşılmıştır.

Mazman (2013), “Programlama Performansını Etkileyen Faktörlerin Bilişsel Tabanlı Bireysel Farklılıklar Temelinde Modellenmesi” başlıklı çalışmada uzamsal beceriler ve çalışma belleği ile öz yeterlilik, cinsiyet, ön deneyim ve üniversite değişkenleri birlikte programlama performansının ne kadarını yordadığı ve bu değişkenlerin önem sıralarını ortaya koymuştur. Mazman yaptığı analizler sonucunda yukarıda sayılan faktörlerin hepsi birlikte programlama başarısının yaklaşık olarak %60 - %65 ini tahmin edebildiği sonucuna ulaşmıştır. Ayrıca ön deneyim ve cinsiyet değişkenlerinin programlama performansı üzerinde anlamlı bir etkisi bulunamamıştır.

Durak (2013), çalışmada Programlama Dilleri I dersini uzaktan eğitim ile alan öğrenciler ile örgün eğitim ile alan öğrenciler arasındaki ders başarısı ve programlamaya karşı tutum farklarını incelemiştir. 38 öğrencinin katıldığı çalışmada dersi tamamen çevrim içi ortamda alan deney grubu ile geleneksel yöntemle dersi alan kontrol grubu oluşturulmuştur. Deney grubu öğrencileri sanal sınıf uygulamalarıyla önceden belirlenen zamanlarda ve kontrol grubu ile birebir aynı ders saati süresince eğitim görmüşlerdir. Araştırma sonucunda çevrim içi öğrenme ortamında dersi alan öğrencilerin daha başarılı oldukları, derse diğer gruptan daha çok devam ettikleri ve programlamaya karşı tutumlarının daha olumlu olduğu sonucuna ulaşılmıştır.

Yılmaz (2013), meslek yüksekokulu öğrencilerinin programlama başarısını etkileyen faktörleri ortaya koymayı amaçlayan çalışmada, öğrenci demografik bilgileri, problem çözme ve analitik düşünme becerisi, kişisel öz yeterlilik ve motivasyonları ve öz düzenleyici öğrenme stratejileri ile programlama başarısı arasında bir ilişki olup olmadığını incelemiştir. Araştırma ilişkisel tarama yöntemi kullanılarak yapılmıştır. Araştırmaya üç farklı üniversiteden 2. sınıf Meslek yüksekokulu öğrencisi katılmıştır. Araştırma sonucunda, cinsiyet, genel akademik not ortalaması, okul türü, motivasyon ve öz yeterlilik ile programlama başarısı arasında anlamlı bir ilişki bulunmuştur. Çalışma sonuçlarına göre programlama başarısının toplam varyansının %18’ini öz düzenleyici öğrenme stratejileri ile açıklanırken, %23’ü demografik özellikler, analitik düşünme ve problem çözme becerileri, kişisel öz yeterlilik ve motivasyonla açıklanmaktadır.

Lau ve Yuen (2011), programlama performansını etkileyen faktörleri ortaya koymak için 131 lise öğrencisi ile bir çalışma yapmıştır. Çalışmada cinsiyet, öğrenme stili, zihinsel modeller, ön akademik beceri ve öğretim ortamı gibi faktörlerin programlama başarısı üzerindeki etkisi birlikte incelenmiştir. Çalışmada incelenen değişkenler programlama başarısına ilişkin varyansın %43,6'sını açıklamıştır. Öğrencilerin bireysel farklılıkları olan cinsiyet, zihinsel modeller ve öğrenme stiline yanı sıra öğretim ortamı ve akademik becerinin öğrenme çıktıları üzerinde istatistiksel olarak anlamlı etkiye sahip olduğu ortaya konulmuştur.

Keskinsoy (2010), meslek lisesi öğrencileri ile yaptığı çalışmasında, genel akademik başarı, matematik başarısı, genel yetenek, yaratıcılık, problem çözme ve bilgisayara karşı tutum değişkenlerinin programlama başarısına etkisini incelemiştir. Çalışma sonucunda yaratıcılık ile programlama başarısı arasında bir ilişki bulunamazken, diğer değişkenler programlama başarısının, endüstri meslek lisesinde %51,8 ini, Anadolu teknik lisesinde %41,2 sini ve teknik lisede %84,7 sini açıklamıştır.

Davidson, Larzon ve Ljunggren (2010), “Programlama öz yeterliliği bir yıl süren giriş seviyesinde programlama dersi süresince nasıl değişir?” sorusuna cevap aramışlardır. Çalışmaya dersi ilk defa alan 53 öğrenci ile daha önce aynı dersi almış 24 öğrenci katılmıştır. Çalışma sonucunda ders sonunda öğrencilerin öz yeterlilik algılarında bir değişiklik bulunamamıştır.

Aşkar ve Davenport (2009), yaptıkları çalışmada cinsiyet, bölüm, bilgisayar deneyimi, bilgisayar kullanım sıklığı ve ailenin bilgisayar kullanımı gibi faktörlerin programlamaya karşı öz yeterlilik algısına etkilerini araştırmışlardır. Çalışmada gerekli bilgileri toplamak için hazırlanan ve İnternet ortamında öğrencilere ulaştırılan bir anket kullanılmıştır. Çalışmaya Bilkent Üniversitesi'nde okuyan farklı bölümlerden Java Programlama dersine kayıt olmuş toplam 326 öğrenci katılmıştır. Çalışma sonunda programlamaya ilişkin öz yeterlilik algısının erkeklerde kadınlara kıyasla daha yüksek olduğu ve bilgisayar mühendisliği öğrencilerinin programlamaya ilişkin öz yeterlilik algılarının diğer bölümlere kıyasla daha yüksek olduğu sonuçlarına ulaşılmıştır. Ayrıca bilgisayar deneyimi, bilgisayar kullanım sıklığı ve aile üyelerinin bilgisayar kullanımının programlama öz yeterlilik algısını anlamlı bir şekilde yordadığı sonucu elde edilmiştir.

Erdoğan (2005), üniversite öğrencilerinin programlama başarısına etki eden faktörlerin belirlemek amacıyla 60 öğrencinin katılımıyla yaptığı çalışmada akademik başarı, bilgisayara karşı tutum, genel yetenek, cinsiyet ve lise türü değişkenleri ile programlama başarısı arasındaki ilişkiyi incelemiştir. Çalışmada ilişkisel tarama yöntemi kullanılmıştır. Yapılan analizler sonucunda matematik ve genel akademik başarısı ile programlama başarısı arasında istatistiksel olarak anlamlı bir ilişki bulunmuş, fakat diğer değişkenler ile programlama başarısı arasında ilişki bulunmamıştır.

Ramalingam, Labelle ve Wiedenbeck (2004), öğrencilerin sahip oldukları zihinsel modeller, programlama öz yeterliği ve önceki deneyimler ile programlama başarısı arasındaki ilişkiyi incelemiştir. 75 öğrencinin katıldığı çalışmanın sonucunda, önceki deneyimlerin ve zihinsel modellerin programlama öz yeterliliği ile doğrudan ilişkili olduğu ve yüksek öz yeterlilik ve iyi oluşturulmuş zihinsel modellere sahip öğrencilerin programlamadan daha başarılı olduğu sonuçlarına ulaşılmıştır.

Grant (2003), çalışmasında öğrenme stillerinin ve cinsiyetin programlama başarısına etkisi olup olmadığını incelemiştir. Ayrıca programlama dersinin öğrencilerin eleştirel düşünme yeteneklerini geliştirip geliştirmediğini incelemiştir. İki ayrı programlama dersinden biri C programlama dili diğeri ise C++ programlama dili kullanılarak işlenmiştir. Çalışma toplamda 41 öğrencinin katılımı ile yapılmıştır. Dönem başında dersi alan öğrencilerin ön bilgileri, eleştirel düşünme becerileri ve öğrenme stilleri belirlenmiştir. Dönem sonunda öğrencilerin programlama ders notu alınmış ve eleştirel düşünme yetenekleri karşılaştırma yapmak için tekrar ölçülmüştür. Çalışma sonucunda programlama dersinin eleştirel düşünmeye anlamlı bir katkı sağlanmadığı sonucuna ulaşılmıştır. Ayrıca cinsiyet ve öğrenme stilleri değişkenlerinin programlama başarısı üzerinde etkisi anlamlı olarak bulunamamıştır.

Fowler, Campbell, McGill ve Roy (2003), öğrenme stillerinin programlama dersindeki başarı ile bir ilişkisinin olup olmadığını araştırmıştır. Öğrencilerin öğrenme stillerini tespit etmek için Kolb'un öğrenme stilleri envanterini kullanmıştır. Bu envanter öğrencileri Değiştiren (Divergent), Özümseyen (Assimilator), Yerleştiren (Accommodator) ve Ayırıştırıcı (Converger) olmak üzere 4 çeşit öğrenme stiline göre sınıflandırmaktadır. Çalışma sonucunda Ayırıştırıcı (Converger) ve

Özümseyen (Assimilator) öğrenme stiline sahip öğrencilerin en yüksek ortalamayı elde ettikleri sonucuna ulaşılmıştır.

Holden ve Weeden (2003) öğrencilerin sahip olduğu bilgisayar deneyiminin programlama başarısına etkisini belirlemeyi amaçlamıştır. Çalışmaya 159 öğrenci katılmıştır. Çalışma başlangıcında öğrencilerin bilgisayar deneyiminin olup olmadığı eğer varsa ne derecede olduğu belirlenmiştir. Çalışma sonunda bilgisayar deneyimi olan öğrencilerin ilk programlama dersinde deneyim olmayan öğrencilere göre daha başarılı olduğu fakat sonraki dönemlerdeki derslerde deneyimin etkisinin olmadığı sonucuna ulaşılmış. Ayrıca deneyimi olan öğrenciler arasındaki deneyim farkının programlama başarısı üzerinde bir katkısı olmadığı gözlenmiştir.

Byrne ve Lyons (2001), ilk yıl programlama dersi ile cinsiyet, akademik başarı, öğrenme stili ve bilgisayar deneyimi arasındaki ilişkiyi incelemişlerdir. Çalışma sonuçlarında cinsiyet açısından programlama başarısında anlamlı bir fark bulunamamıştır. Öğrenme stilleri sonuçları istatistiksel olarak anlamlı olmasa da, “Ayrıştırma” stiline sahip olan öğrencilerin programlamadan daha başarılı olduğu tespit edilmiştir. Çalışmaya katılan öğrencilerin üniversite giriş sınavında aldıkları fen ve matematik puanları ile programlama başarısı arasında istatistiksel olarak anlamlı korelasyon bulunmuştur. Ayrıca öğrencilerden daha fazla bilgisayar deneyimine sahip olanların programlama dersinden daha başarılı oldukları tespit edilmiştir.

Wilson ve Shrock (2001), 12 bağımsız değişkeni ele alarak bilgisayar bilimlerine giriş dersindeki başarıya etki eden faktörleri araştırmıştır. Çalışmada kullanılan bağımsız değişkenler; cinsiyet, programlama deneyimi, dersteki rahatlık düzeyi, çalışma stili, matematik alt yapısı, programlama öz yeterliliği, cesaretlendirme ve başarı/başarısızlık niteliğidir. Araştırmaya 105 öğrenci katılmıştır. Çalışma sonucunda ders başarısını ön görebilecek değişkenler önem sırasına göre, dersteki rahatlık derecesi, matematik alt yapısı ve başarı/başarısızlık niteliği olarak sıralanmıştır.

Austin (1987), programlama başarısını tahmin etmek için kullanılacak değişkenleri belirlemeyi amaçlayan bir çalışma yapmıştır. 76 öğrencinin katıldığı çalışmada araştırmacının geliştirdiği bir test ile öğrencilerin demografik bilgileri,

akademik geçmişi ve bilgisayar geçmişi gibi veriler toplanmıştır. Ayrıca bilgisayara karşı tutum, kişilik özellikleri, programlama yeteneği, bilişsel stiller ve matematik yeteneği gibi verileri elde etmek için dönem boyunca testler uygulanmıştır. Araştırma sonucunda öğrencilerin bilişsel stilleri, matematik yetenekleri, programlama yetenekleri ve bilgisayara karşı tutumları ile ders başarısı arasında istatistiksel olarak anlamlı bir korelasyon bulunmuştur.

Alspaugh (1972) daha önce programlama deneyimi olmayan 50 öğrenci ile yapılan çalışmada öğrencilerin mizacı, IBM programlama yetenek testi puanları, eleştirel düşünme yeteneği ve matematik başarıları ile programlama dersi başarısı arasındaki ilişki incelenmiştir. Çalışma sonucunda matematik başarısının programlama başarısı için önemli bir kriter olduğu sonucuna ulaşılmıştır. Yapılan regresyon analizi sonuçlarına göre kullanılan bağımsız değişkenler programlama başarısının çok küçük bir yüzdesini açıklamaktadır. Bu durum da henüz keşfedilmeyen birçok faktörün programlama başarısına etki ettiğinin göstergesi olarak yorumlanmıştır.

Yapılan çalışmalar incelendiğinde geçmiş yıllarda programlama eğitimi alan kişilerin özellikleri araştırılırken, güncel çalışmalarda programlama öğretim süreci, yöntemi ve kullanılan araçlar üzerine çalışmalar yapıldığı görülmektedir. Programlama temellerinin eğitimi için özel olarak geliştirilen programlama dillerinin kullanımının programlama eğitime çoğunlukla katkı yaptığı görülmektedir. Öğrencilerin özelliklerini inceleyen çalışmalara bakıldığında mantıksal düşünme, analitik düşünme, öz yeterlilik, problem çözme, matematik başarısı, öğrenme stilleri, zihinsel modeller, ön akademik beceriler ve bilgisayar deneyimi gibi faktörlerin programlama başarısı üzerinde etkili olduğu görülmektedir.

## **2.6 Programlama Yetenek Testi**

Programlama yetenek testleri bir kişinin programlamada başarılı olup olmayacağını önceden tahmin etmeyi amaçlayan testlerdir. Yetenek testleri programlama eğitiminin ve mesleğinin varoluşuyla aynı dönemde var olmuştur (Tukiainen ve Mönkkönen, 2002). Geliştirilen ilk yetenek testleri, sayı dizilerini tamamlama, rakamlarla temsil edilen analogileri anlama ve aritmetik problemleri çözme üzerine odaklanmıştır. Daha sonra geliştirilen testlere mantıksal yetenek, tanımlamaların yorumlanması, dikkat gibi özellikler de eklenmiştir.

Geliştirilen ilk programlama yetenek testi IBM Programlama Yetenek Testi olarak bilinmektedir (Tukiainen ve Mönkkönen, 2002). İlerleyen zamanlarda Wolfe (1971) eski testlerin eksikliklerini tamamlamak amacıyla bir test serisi geliştirmiştir. Geliştirilen bu testler ile programlama başarısını belirlemek iki saatten uzun bir zaman alıyordu. Sonraki yıllarda sadece programlama yeteneklerini ölçmenin yeterli olmadığı ve problemleri belirleme, yorumlama ve çözüme yeteneklerinin de teste dahil edilmesi gerektiği anlaşılmıştır (Winrow, 1999).

Programlama yetenek testleri, standart testler, karma testler ve programlama simülasyonları olarak üç gruba ayrılırlar (“Computer Programming Aptitude Test”, t.y). *Standart Testler*, temel seviye bilgi işlem görevlerinde gerekli olan sayısal akıl yürütme yeteneği, mantıksal akıl yürütme ve sözel olmayan akıl yürütme yeteneklerini ölçer. *Karma testler*, mantıksal akıl yürütme, sayısal akıl yürütme, desenleri kavrama, karmaşık işlemleri takip etme ve dikkat yeteneğini ölçer. Bu iki test hiçbir programlama bilgisi olmayan kişiler için uygun testlerdir. *Programlama simülasyonları* ise sahte kodlar, kontrol yapıları, tablolar, tanımlamalar, diziler, döngüler ve diğer programlama yapılarını içermektedir. Bu tür testler programlama bilgisi olan kişilerin seviyesini belirlemek için kullanılmaktadır. Programlama yetenek testlerinin kullanıldığı birçok çalışma yapılmıştır. Bunlardan bazıları: Haris (2014) geliştirmiş olduğu Programlama Yetenek Dili (PAL) ile hazırladığı Programlama Yetenek Testi ve Öğreticisi (PATT)’ni kullanarak yaptığı test ile programlama başarısını arasında yüksek seviyede anlamlı bir ilişki olduğunu belirtmiştir. Lacher ve Lewis (2015) Kent Üniversitesi tarafından geliştirilen programlama yetenek testini kullanarak yaptığı çalışmada yetenek testinin öğrenci başarısını doğru bir şekilde tahmin edebildiği sonucuna ulaşmıştır. Dehnadi (2006) ise zihinsel modelleri temel alarak yaptığı çalışmada geliştirmiş olduğu programlama yetenek testinin programlama başarısını tahmin etmede etkili olduğu sonucuna ulaşmıştır. Fakat sonraki uygulamalarda öğrencilerin programlama başarısını aynı oranda tahmin edemediğini belirtmiştir. (Bornat ve diğerleri, 2008).

Bu çalışmada yukarıda bahsedilen test türlerinden karma test grubuna giren ve 26 sorudan oluşan Kent Üniversitesi Kariyer Danışma Biriminin hazırladığı, Bilgisayar Programlama Yetenek Testi kullanılacaktır. Testin Türkçe ’ye çevrilmesi Yiğit (2016) tarafından yapılmıştır.



## 2.7 Mantıksal Düşünme Yeteneği

Mantıksal düşünme yeteneği üzerine ilk çalışmaları Jean Piaget ve ekibi başlatmıştır (Tobin ve Capie, 1981). Piaget, yaptığı çalışmalar neticesinde insanların bilişsel gelişim aşamalarını dört aşama olarak tanımlamıştır. Bunlar 0-2 yaş Duyu – motor (sensori motor) dönemi, 2-7 yaş İşlem öncesi (preoperational) dönem, 7-11 yaş Somut işlemler (concrete operational) dönemi, 11-18 yaş Soyut işlemler (formal operational) dönemidir (Yüzüak ve Dökme, 2015). Piaget'in kuramına göre mantıksal düşünme becerileri somut işlemler döneminde gelişmeye başlamaktadır. Soyut işlemler döneminde ise bu beceriler gelişerek devam etmektedir. Birey soyut işlemler döneminde hipotetik düşünme, kombinasyonel düşünme, değişkenleri değiştirme ve kontrol etme, korelasyonel düşünme, oransal ve olasılıklı düşünme gibi mantıksal düşünme yetenekleri gelişmektedir (Lawson, 1978; Senemoğlu, 2012; Tortop, 2016; Yüzüak ve Dökme, 2015). Soyut işlemler dönemindeki bir birey üst düzey matematiksel düşünme becerisi gerektiren problemleri çözmeyi kolayca öğrenir ve edindiği kazanımları yeni problemlerde uygulayabilir. Ayrıca problemin çözümünde kendine bir plan yapabilmekte ve olasılıkları değerlendirebilmektedir.

Formal operasyonel düşünme olarak da adlandırılan mantıksal düşünme yeteneği, bireylerin karşılaştıkları problemler karşısında kullandıkları zihinsel işlemlerdir (Yüzüak ve Dökme, 2015). Mantıksal düşünme bir sonuca varmak için kararlı bir düşünme sürecini gerektirir. Bu da problemle ilgili tüm gerekçeleri, fikirleri ve sonuçları zincirleme bir biçimde düzenlemeyi gerektirir (Bozdoğan, 2007; Kılıç ve Sağlam, 2009). Programlamada bir problemin çözümü için ilk olarak algoritma oluşturulması yani çözümün birbiri ardına sırayla planlanması, mantıksal düşünme yeteneğindeki problemin çözümü için gerekli olan zincirleme düşünme biçimi ile paralellik göstermektedir. Bu nedenle programlama başarısı için mantıksal düşünme yeteneği oldukça önemlidir. Sebetci ve Aksu (2014), yaptıkları çalışmada öğrencilerin analitik düşünme ve mantıksal düşünme yeteneklerinin programlama başarısına etkisini incelemiştir. Çalışma sonucunda analitik düşünme ve mantıksal düşünme becerileri yüksek olan öğrencilerin programlamadan daha başarılı olduğu sonucuna ulaşmışlardır. Tortop (2016) üstün zekâlı öğrenciler ile yaptığı çalışmada programlama eğitiminin öğrencilerin mantıksal düşünme yeteneğine bir katkısı olup olmadığını araştırmıştır. Araştırmasının sonucunda programlama öğrenmenin

öğrencilerin mantıksal düşünme yeteneklerinde istatistiksel olarak anlamlı bir artışa neden olduğu görülmüştür. Bu iki çalışma göz önüne alındığında programlama eğitimi ve mantıksal düşünme yeteneği birbirini destekleyen özellikler olduğu görülmektedir.

## 2.8 Öğrenme Stilleri

Her öğrencinin bilgiyi edinme yolları ve bilgiyi işleyişi birbirinden farklıdır. Nitelikli bir eğitim için öğrencilerin farklılıkları dikkate alınmalıdır. Öğrenciye bilgiyi kazandırmanın en etkili yolu onun öğrenme stiline uygun şekilde bilgiyi aktarmak olacaktır. Öğrenme bilim adamlarının yıllardır üzerinde çalıştığı, farklı yaklaşımlarla tanımlamaya çalıştığı bir kavramdır. Öğrenme ile ilgili yapılan çalışmalar incelendiğinde ilk olarak davranışçı kuramlar ile tanımlandığı, davranışçı kuramların eksiklerinin farkına varılıp bilişselci kuramların yaygınlaşması ile bilişselci kuramlara göre tanımlamalar getirildiği görülmektedir (Açıkgöz, 2003; Gencel, 2006). Davranışçı kuramı benimseyen araştırmacılar ile bilişselci kuramı benimseyen araştırmacılar arasındaki temel fark öğrenme ürünü üzerindedir. Davranışçılar öğrenme ürünü olarak davranış değişikliğini görürken, bilişselciler davranış potansiyelini yani gözlenemeyen içsel, bilişsel süreçlerdeki değişimi de öğrenme ürünü olarak görmüşlerdir (Açıkgöz, 2003, s. 9). Davranışçı ve bilişselci kuramların öğrenme konusunda görüş birliği içinde olduğu noktalar, öğrenmenin kalıcılığı, yaşantı ürünü olması, kişide bir değişmeye neden olması ve hem çevresel hem de içsel etkenlerden etkileniyor olması şeklinde sıralanabilir. Bu iki görüşün ayrıldığı noktalar ise; (i) davranışın öğrenmenin kendisi (davranışçı) olarak görülmesi ile davranışın öğrenmenin bir sonucu (bilişselci) olarak görülmesi, (ii) öğrenme sürecinin öğrenenden bağımsız, çevre ile ilgili kabul edilmesi (davranışçı) ile öğrenme sürecinin öğrenenin etkin olduğu bir süreç olarak kabul edilmesi (bilişselci) şeklinde özetlenebilir (Açıkgöz, 2003, ss. 8–11; Gencel, 2006, s. 26).

Herkesin uzlaştığı bir öğrenme tanımı bulunmamaktadır (Illeris, 2009; Senemoğlu, 2012, s. 88). Günümüze kadar yapılan tanımlardan öne çıkan tanımlardan bazıları şu şekildedir: Öğrenme “kişinin performansında veya performans potansiyelinde meydana gelen kalıcı değişimlerdir” (Driscoll, 2000, s. 11). Mayer'e (1982) göre öğrenme, “kişilerin eski tecrübelerine bağlı olarak bilgi ve davranışlarında oluşan kısmen göreceli kalıcı değişimlerdir.”(Akt. Özden ve Şimşek, 1998, s. 3). Shuell'e

(1988) göre ise öğrenme, “bireyin aktif, bilinçli olarak bazı stratejiler kullanarak çevresinden bilgi edinme ve hafızasında var olan bilgilerle birleştirerek kendine kalıcı davranış oluşturma süreci” olarak tanımlanmıştır (Akt. Arı, 2011, s. 6).

Davranışçı kuramın yerini bilişselci kuramların alması ile eğitim ve öğretim anlayışında da değişiklikler olmuştur. Öğretmen merkezli eğitimden öğrenci merkezli eğitim anlayışına geçilmiştir. Öğrenci merkezli eğitim anlayışı öğrenenlerin bireysel farklılıklarına daha çok dikkat çekmiştir. Eğitimcilerin unutmaması gereken önemli bir nokta her öğrencinin farklı şekilde öğrendiği gerçeğidir (Lukow, 2002, s. 20). Beyin üzerindeki çalışmalar ve öğrencilerin öğrenme sürecindeki farklılıklar öğrenme stilleri konusuna temel olmuştur (Gencel, 2006, s. 27). Öğrenme stili kavramının tanımı birçok araştırmacı tarafından yapılmıştır. Altun (2016, s. 24) öğrenme stilini “Öğrencilerin, öğrenme çevresini nasıl algıladıkları, bu çevreyle nasıl etkileşim kurduklarını, nasıl tepki verdiklerini ortaya koyan bireysel özellikler ve tercihler” şeklinde tanımlamıştır. Bir başka çalışmada öğrenme stilleri öğrencinin imzası olarak vurgulanmıştır (Dunn, Beudury, Klavas ve Babadoğan, 1991). Kolb’a (1984, ss. 24–29) göre öğrenme stili, bilgiyi alma ve işlemede bireyin tercih ettiği yöntemdir (Akt. Gencel, 2006). Öğrenme stilleri bilgiyi zihinde yerleştirme, geçmiş yaşantılar, algılama, çevre etkisi ve kalıtsal özellikler gibi konularla ilgili çok boyutlu bir kavram olduğu için öğrenme stillerine farklı bakış açıları ile yaklaşılmıştır (Gencel, 2006).

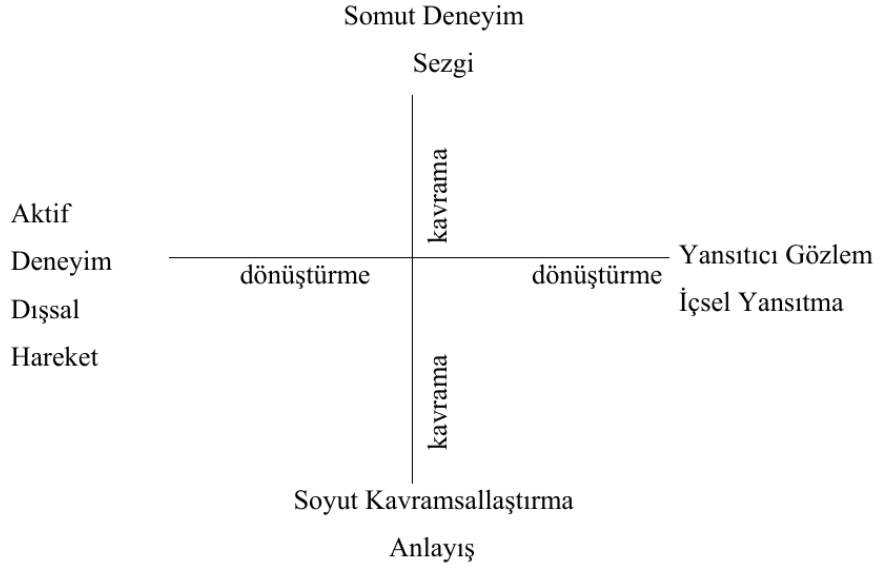
Öğrenme stilleri üzerinde yapılan araştırmalar; kapsamı, değişkenliği, öğrenme üzerindeki etkisi ve belirlenmesinde kullanılacak araçlar gibi konularda farklılıklar göstermektedir. Bu farklılıklar farklı öğrenme stilleri modelleri oluşturulmasını sağlamıştır. Bunların başlıcaları Hill’in Bilişsel Harita Modeli, Hermann’ın Beyinsel Baskınlık Modeli, Myers ve Brings’in Tür Gösteri Modeli, Grasha ve Rirchman’ın Öğrenme Stilleri Modeli, Dunn ve Dunn’ın Öğrenme Stilleri Modeli, Gregorc’un Düşünme Stilleri Modeli, Felder ve Sicerman’ın Öğrenme stilleri boyutlarına ilişkin Modeli ve Kolb’un Deneysel Öğrenme Modeli’dir. Howard Gardner’in Çoklu Zekâ Kuramı ise, doğrudan bir öğrenme Stili modeli olmamakla birlikte, öğrenme stillerinin temel teorisi olarak algılanmaktadır (Kılıç, 2002, s. 11)

### 2.8.1 Kolb Öğrenme Stilleri Modeli

Öğrenme stilleri üzerine yapılan çalışmalardan en bilinenlerden birisi de Kolb tarafından geliştirilen ve kendi geliştirdiği Deneysel Öğrenme Kuramı'nın bir uzantısı olarak ortaya koyduğu öğrenme stilleridir (Altun, 2016; Gencel, 2006). Kolb deneysel öğrenme kuramını öğrenmede deneyimi ele alan Dewey, bireyin etkin katılımını vurgulayan Lewin ve zekayı bireyin çevresi ile etkileşimi ve doğuştan gelen özelliklerin ortak ürünü olarak kabul eden Piaget'in çalışmalarına dayandırmaktadır (Aşkar ve Akkoyunlu, 1993; Gencel, 2006; Kolb ve Kolb, 2005). 1960'lı yıllarda deneysel öğrenme çalışmalarına başlayan Kolb, öğrenmeyi dört adımdan oluşan bir süreç olarak açıklamaktadır. Kolb'a göre bireyler yaşadıkları çevre ile etkileşimde bulunarak somut yaşantılara sahip olurlar. Kolb bireylerin sahip oldukları bu yaşantıları farklı şekillerde gözlemleyerek yansıttıklarına dikkat çekmektedir. Buna ek olarak yapılan yansıtıcı gözlemlerin, soyut kavramsallaştırma yapılmasında, ilke ve genellemelerin oluşmasında etkili olduğu vurgulanmaktadır. Sonuç olarak bireyler oluşturdukları soyut kavramları, ilkeleri ve genellemeleri daha sonraki etkinliklerinde ve ileri seviye öğrenmelerinde kaynak olarak kullanacaklardır. Bu süreç Somut Yaşantı (Concrete Experience), Yansıtıcı Gözlem (Reflective Observation), Soyut Kavramsallaştırma (Abstract Conceptualization) ve Aktif Deneyimler (Active Experimentation) olarak isimlendirilen dört öğrenme biçimini barındıran bir döngü şeklinde devam etmektedir (Altun, 2016, ss. 76–77; Aşkar ve Akkoyunlu, 1993, s. 37; Gencel, 2006, s. 38). Bu süreç Somut Yaşantı (Concrete Experience), Yansıtıcı Gözlem (Reflective Observation), Soyut Kavramsallaştırma (Abstract Conceptualization) ve Aktif Deneyimler (Active Experimentation) olarak isimlendirilen dört öğrenme biçimini barındıran bir döngü şeklinde devam etmektedir (Altun, 2016, ss. 76–77; Aşkar ve Akkoyunlu, 1993, s. 37; Gencel, 2006, s. 38). Her öğrenme biçimi farklı bir öğrenme yolunu simgeler. Somut yaşantı biçimini *hissederek* öğrenme, yansıtıcı gözlem biçimini *izleyerek* öğrenme, soyut kavramsallaştırma biçimini *düşünerek* öğrenme, aktif deneyim biçimini de *yaparak* öğrenme simgelemektedir (Altun, 2016, s. 77; Aşkar ve Akkoyunlu, 1993, s. 38).

Kolb öğrenme stilleri ile ilgili olarak iki temel boyut ortaya koymuştur. Bunlar *kavrama* ve *dönüştürme* boyutlarıdır. Şekil 4' de görüldüğü gibi kavrama boyutunun iki ucunda somut deneyim ile soyut kavramsallaştırma yer almaktadır ve bireyin

çevresi ve yaşadıklarını algılama tercihlerini göstermektedir. Kavrama boyutu bireyin somut deneyimlerden sezgisel yollarla öğrenmesi ya da deneyimlerden sembolik anlamlar oluşturmasını içermektedir.



Şekil 4 Kolb öğrenme stilleri modelinin iki boyutu (Gencel, 2006, s. 39)

Dönüştürme boyutunda ise aktif deneyim ve yansıtıcı gözlem bulunmaktadır. Bu boyut bireyin bilgi dönüştürme ve işleme tercihlerini temsil eder. Dönüştürme dışsal hareket ya da içsel yansıtma yolu ile gerçekleşir (Gencel, 2006, ss. 39–40).

Bireyler birden fazla öğrenme biçimini seçebilir ya da bunları bir arada kullanabilir. Öğrenme öğretme sürecinde somut deneyim, yansıtıcı gözlem, soyut kavramsallaştırma ve aktif deneyim öğrenme biçimlerine uygun öğrenme etkinliklerinin düzenlenmesi, deneysel öğrenme kuramına dayalı eğitimin temellerini oluşturmuştur (Gencel, 2006, s. 42).

### 2.8.2 Kolb'un Deneysel Öğrenme Kuramına Dayalı Eğitim

Deneysel öğrenme kuramına dayalı eğitim, temelde tüm öğrenme biçimlerine ve öğrenme stillerine uygun öğrenme etkinliklerinin düzenlenmesini gerektirir. Kolb'a göre öğrenme ilk olarak somut deneyimden yansıtıcı gözleme, daha sonra soyut kavramsallaştırmadan aktif deneyime doğru yapılandırılmalıdır (Kolb ve Kolb, 2005).

Somut deneyim yolu ile öğrenen öğrenciler, öğrenme konusu ile ilgili özel durumlara, örneklere ve olay içinde yer almaya gereksinim duymaktadırlar. Bu

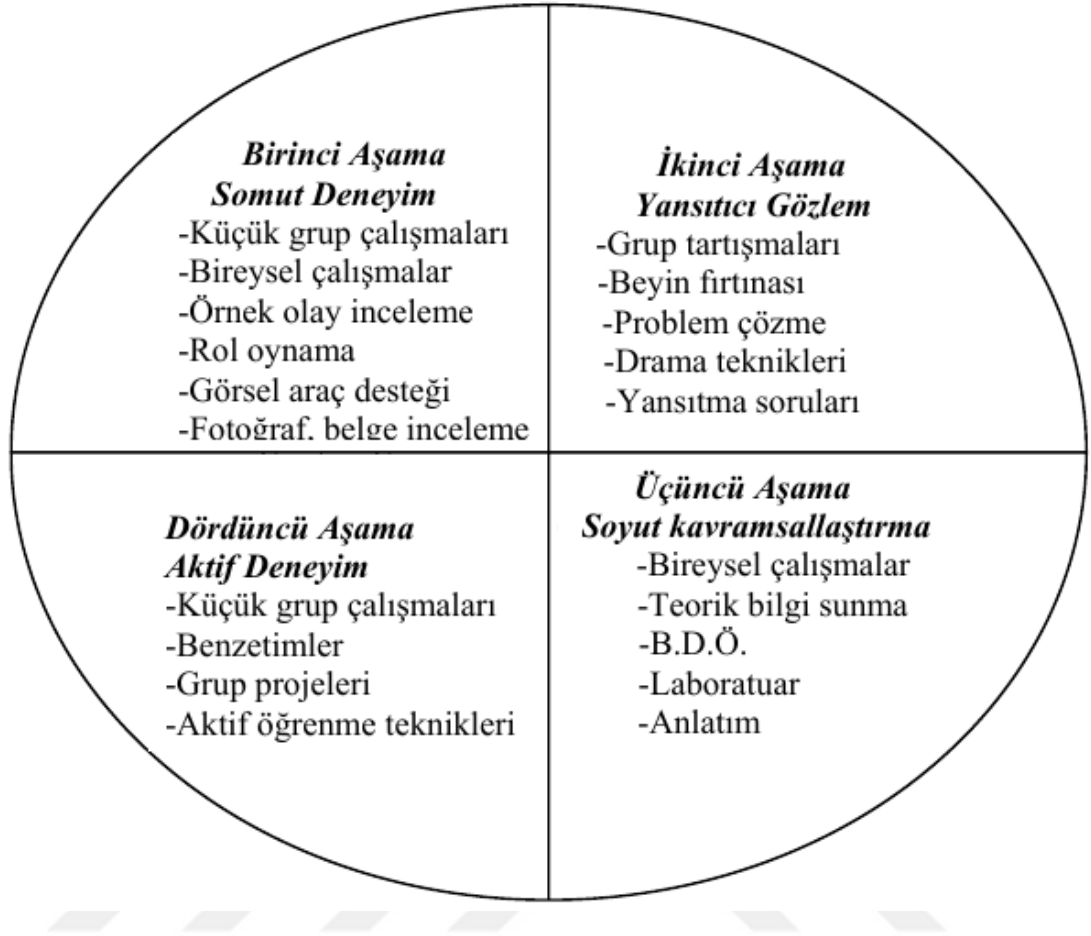
nedenle konuların gerçek yaşamla ilişkilendirilmesi gerekmektedir. Örnek olay ve rol oynama etkinliklerinin bu öğrenme yoluna uygun olduğu belirtilmektedir (Gencel, 2006, s. 43).

Yansıtıcı gözlem yolu ile öğrenen öğrenciler, öğrendikleri ve gözledikleri durumların üzerinde düşünmeyi tercih ederler ve farklı bakış açıları ile olay ve olguların temelindeki düşünceyi anlamaya çabalarlar. Bu öğrenme yolunda neden ve niçin soruları önem kazanmaktadır. Öğrenme konusuna ilişkin farklı bakış açılarını ortaya koymaya olanak veren tartışma etkinlikleri bu yolla öğrenenler için uygun olacaktır. Yansıtıcı gözlem etkinlikleri somut deneyim etkinliklerinin devamı şeklinde olmalıdır (Gencel, 2006, s. 44).

Yansıtıcı gözlem etkinliklerinden sonra, soyut kavramlaştırma öğrenme yoluna uygun etkinlikler yapılması öğrenme sürecini olumlu etkileyecektir. Bu aşamada mantık, düşünce ve kavramlara odaklanma söz konusudur. Bu yolla öğrenen öğrenciler bilgilerin mantıksal olarak bir düzen içerisinde sunulmasına ihtiyaç duymaktadırlar. Bir diğer deyişle konunun teorik bilgilerinin düzenli bir şekilde verilmesi gerekmektedir. Öğrenciye bireysel çalışma olanağı verilmesi öğrenme sürecini olumlu etkileyecektir. Ayrıca laboratuvar çalışmaları, projeler ve bilgisayar destekli öğretim öğrenmeyi kolaylaştıracaktır.

Aktif deneyim yoluyla öğrenen öğrenciler, uygulamaya dayalı etkinliklerle öğrenmeyi tercih etmektedir. Gözlem yapmak, dinlemek yerine etkinliğe katılarak öğrenmek önem kazanmaktadır. Bu yolla öğrenen öğrenciler daha önce öğrendikleri bilgileri uygulamaktan, gerçek hayatta kullanmaktan zevk alırlar.

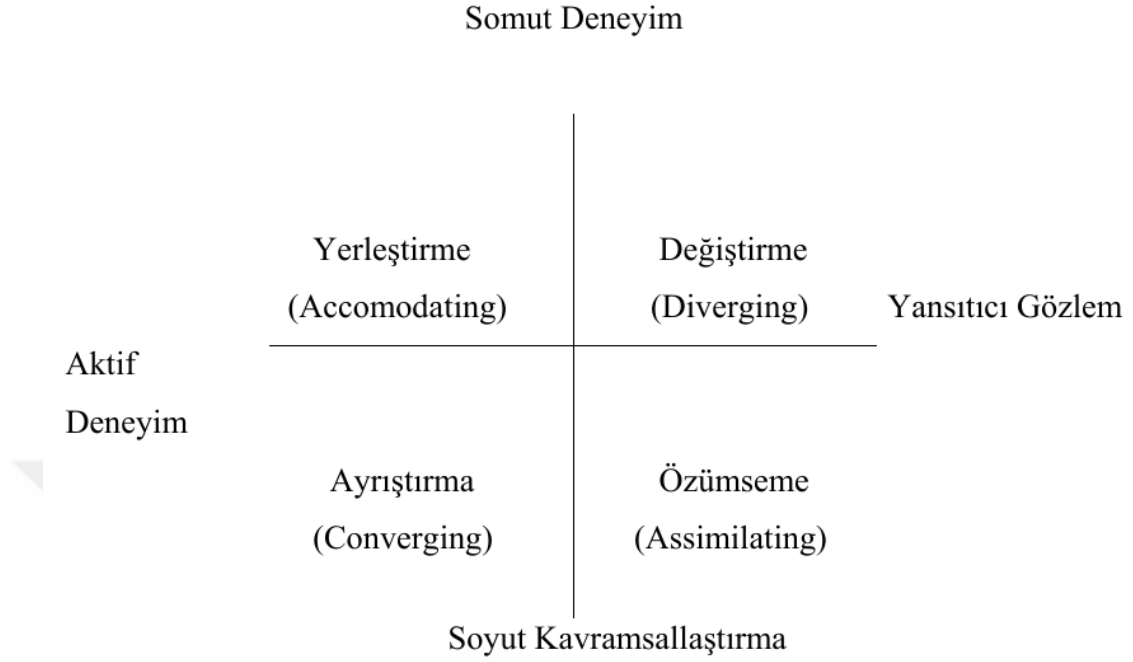
Deneyisel öğrenme kuramında aşama aşama birbirini takip eden öğrenme yolları ve bu yollara uygun öğrenme etkinlikleri Şekil 5 de bir arada gösterilmiştir.



Şekil 5 Deneyimsel öğrenme döngüsüne uygun etkinlikler (Gencel, 2006, s. 46)

Kolb öğrenme stilleri modelinde öğrenme bir döngü şeklinde devam etmektedir. Öğrenme sürecinde ilk olarak somut deneyimler yolu ile konuya ilişkin somut deneyim kazanan öğrenciler, ikinci aşamada edindiği bilgileri kullanarak konuya yeni bakış açıları getirecektir. Üçüncü aşamada ise edindiği bilgilerin mantıksal yapısını kavrayacaktır. Son aşamada ise edindiği bilgileri uygulama yaparak aktif deneyimler kazanacak ve öğrenme döngüsünü tamamlayacaktır (Gencel, 2006, s. 45).

Deneyimsel öğrenme kuramında öğrenme stilleri bahsedilen bu dört öğrenme yolunun bileşenleri olarak tanımlanmıştır. Bunlar *Yerleştirme*, *Değiştirme*, *Ayrıştırma* ve *Özümseme* olarak isimlendirilmişlerdir.



Şekil 6 Kolb'un öğrenme stilleri sınıflaması (Gencel, 2006, s. 40)

Yerleştirme öğrenme stili, aktif deneyimler ve somut yaşantı öğrenme biçimlerinin içerisinde yer almaktadır. Bu öğrenme stiline sahip olan öğrencilerin en önemli özellikleri geçmiş deneyimlerden faydalanarak öğrenme becerisine sahip olmalarıdır (Kolb, 1984, s. 78). Liderlik özelliğine sahip olan bu öğrenciler, teknik çözümler yerine diğer bireylerin kişisel bilgilerine başvurmayı tercih ederler. Meraklı ve araştırmacıdırlar. Kararları yürütme, plan yapma ve yeni deneyimler içerisinde yer alma belli başlı özellikleridir (Altun, 2016, s. 78; Aşkar ve Akkoyunlu, 1993, s. 39; Gencel, 2006, s. 49).

Değiştirme öğrenme stili, yansıtıcı gözlem ve somut yaşantı öğrenme biçimlerini kapsar. Hayal gücünü kullanma, problemleri tanımlama, algılama ve farklı bakış açıları ile değerlendirme ve düşünme yeteneği bu öğrenme stiline sahip olanların en önemli özellikleridir. Bu öğrenme stiline sahip bireyler somut durumları birçok açıdan gözden geçirebilir ve ilişkileri anlamlı bir şekilde organize edebilir (Altun, 2016, s. 78; Aşkar ve Akkoyunlu, 1993, s. 39; Gencel, 2006, ss. 46–47).

Ayrıştırma öğrenme stili, soyut kavramsallaştırma ve yansıtıcı gözlem öğrenme biçimlerini kapsar. Mantıksal çözümleme, problem çözme becerileri ve doğru karar



verme becerileri gelişmiştir. Bu stile sahip olan öğrenciler sosyal ve kişiler arası etkinlikler yerine teknik konuları barındıran etkinlikleri tercih ederler (Kılıç, 2002, s. 24). Fikirlerin mantıksal analizi, sistematik planlama, problem çözme ve karar verme belli başlı özellikleridir. Bu öğrenciler test ederler, sınırlar, nasıl çalıştığına bakarlar ve işlem basamaklarını sıra ile takip ederek uygularlar (Altun, 2016, s. 78; Aşkar ve Akkoyunlu, 1993, s. 39; Gencel, 2006, s. 48; Kılıç, 2002, s. 24).

Özümseyen öğrenme stili, yansıtıcı gözlem ve soyut kavramsallaştırma öğrenme biçimlerini kapsar. Bu öğrenme stiline sahip olan öğrenciler öğrenirken soyut kavramlar ve fikirler üzerine odaklanır. Farklı kaynaklardan edindikleri bilgileri mantıksal bir bütün haline getirme konusunda başarılıdırlar. En kuvvetli yönleri planlama yapma, problemleri belirleme ve kuramlar geliştirmedir (Altun, 2016, s. 78; Aşkar ve Akkoyunlu, 1993, s. 39; Gencel, 2006, s. 47; Kılıç, 2002, s. 23).

Fowler, Campbell, McGill ve Roy (2003) yaptıkları çalışmada Kolb Öğrenme Stilleri Envanterini kullanmışlardır. Programlama eğitimi gören mühendislik öğrencilerinin öğrenme stilleri ile programlama başarıları arasındaki fark incelenmiştir. Çalışmaya göre mühendislik öğrencilerinin %46'sı Ayırıştırma (Converging) ve %42'si Özümseme (Assimilating) öğrenme stiline sahip olduğu belirlenmiştir.

Byrne ve Lyons (2001) ilk yıl programlama başarısı üzerine yaptığı araştırmada öğrencilerin öğrenme stillerini öğrenmek için Kolb'un öğrenme stilleri envanterini kullanmıştır. Çalışma sonucunda Ayırıştırma (Converging) stiline sahip olan öğrencilerin daha başarılı olduğu sonucuna ulaşmıştır.

Kolb ve Kolb (2005, s. 6) mühendislik alanının Ayırıştırma öğrenme stiline uygun bir alan olduğunu belirtmiştir. Yukarıda bahsedilen çalışmaların sonuçları da bu düşünceyi doğrulamaktadır. Programlama eğitimi genel olarak mühendislik alanında veriliyor olsa da diğer alanlarda da yaygın bir şekilde verilmektedir. Yukarıda bahsedilen çalışmalar her ne kadar mühendislik öğrencileri ile yapılmış olsa da asıl araştırılan konu öğrenme stilleri ile programlama başarısı ilişkisidir. Bu açıdan baktığımızda programlama dersine uygun olan öğrenme stiline Ayırıştırma olduğu görülmektedir.

## 2.9 Programlama Öz Yeterlilik Seviyesi

Bandura (1986)'ya göre öz yeterlilik, insanların belirli bir amaca ulaşmak için kapasitelerini ve sahip oldukları yetenekleri organize edebilme ve eyleme sokabilme yargısıdır. (Akt. Aşkar ve Davenport, 2009, s. 1). Bir başka deyişle bireyin belirli bir konuda başarı gösterme kapasitesine olan inancıdır (Aşkar ve Davenport, 2009, s. 1). Bandura 'ya göre öz yeterlilik dört kaynaktan beslenmektedir.(i) Kişisel deneyimler, (ii) Başkalarından edinilen deneyim. Örneğin rol model olarak alınan kişiden gözlemlenen deneyimler. (iii) Toplumsal ikna, bir durumla ilgili olarak bireyin çevresinden aldığı öneriler. (iv) Duygu kontrolü, kişinin mutsuzluklarını, stresini kontrol edebilme yeteneği (Aşkar ve Umay, 2001, s. 2; Bandura, 1994, ss. 2-3). Kişinin sahip olduğu öz yeterlilik algısı, etkinliklerin seçimini, güçlükler karşısındaki sabrını, sonuca ulaşmak için harcadığı emeği ve performansını etkiler (Aşkar ve Umay, 2001, s. 1). Bu açıdan bakıldığında öz yeterlilik eğitim açısından oldukça önemli bir kavramdır. (Aşkar ve Davenport, 2009). Aynı bilişsel becerilere sahip iki kişiden öz yeterliliği yüksek olan diğerine nazaran zor görevleri başarmada daha başarılı olduğu ifade edilmektedir (Altun ve Mazman, 2012; Aşkar ve Davenport, 2009). Konuyla ilgili yapılan araştırmalarda, öz yeterlilikleri yüksek olan öğrencilerin olumsuzluklar karşısında daha dirençli oldukları, bir işi tamamlamak için daha çok çaba harcadıkları, ısrarlı ve sabırlı oldukları belirtilmiştir (Pajares, 1996). Birey her alan için geçerli tek bir öz yeterlilik algısına sahip değildir. Yani bireyin sahip olduğu öz yeterlilik algısı her türlü eylem için etkili değildir. Bu nedenle öz yeterlilik durum, konu ve yapılacak işe göre değişir (Aşkar ve Dönmez, 2004; Bandura, 2012, s. 15). Bir konuda yüksek öz yeterliliğe sahip olan birisi, bir başka konuda düşük öz yeterliliğe sahip olabilir. Bununla birlikte farklı öz yeterlilik alanları birbirini destekleyebilir (Altun ve Mazman, 2012; Ramalingam ve diğerleri, 2004). Örneğin bisiklet sürme konusunda öz yeterliliği yüksek olan bir kişi motosiklet kullanmada başarılı olabilir. Her alana özgü ayrı bir öz yeterlilik algısı olduğu için farklı alanlardaki öz yeterlilik algılarını ölçen testler geliştirilmiştir. Ülkemizde bilgisayar alanında; bilgisayar kullanımına ilişkin öz yeterlilik ölçeği (Aşkar ve Umay, 2001), eğitim yazılımı geliştirme öz yeterlilik algısı ölçeği (Aşkar ve Dönmez, 2004), eğitsel İnternet kullanım öz yeterlilik ölçeği (Şahin, 2009), bilgisayar programlamaya ilişkin öz yeterlilik algısını ölçeği (Altun ve Mazman, 2012) gibi testlerin uyarlandığı ya da geliştirildiği görülmektedir.

Alan yazında Programlamaya ilişkin Öz yeterlilik Ölçeği kullanılan bazı çalışmalar:

Davidson ve diğerleri (2010), programlama öz yeterliliğinin programlama dersi alan öğrencilerde nasıl değiştiğini incelemiştir. 77 öğrencinin katıldığı çalışma sonucunda bir yıl boyunca ders alan öğrencilerin programlama öz yeterliliğinde bir değişim olmadığı gözlemlenmiştir.

Aşkar ve Davenport (2009) yaptıkları çalışmada erkek öğrencilerin kızlara göre daha yüksek programlama öz yeterliliğine sahip olduğu görülmüştür. Ayrıca aile bireylerinin bilgisayar kullanım durumunun ve bireyin bilgisayar deneyiminin programlama öz yeterliliğini anlamlı şekilde yordadığı sonucuna ulaşılmıştır.

Ramalingam ve diğerleri (2004) bireyin bilgisayar deneyimi, programlama öz yeterliliği ve sahip olduğu zihinsel modellerin programlama performansı üzerindeki etkisini incelemişlerdir. Çalışma yüksek öz yeterliliğin ve iyi oluşturulmuş zihinsel modellerin programlama başarısını pozitif yönde etkilediğini göstermiştir.

Jegede (2009), mühendislik öğrencilerinin programlama deneyimi ile programlama öz yeterliliği arasındaki ilişkiyi incelemiştir. Öğrencilerin programlama deneyimi Programlama deneyim anketi ile öz yeterliliği ise Java programlama öz yeterlilik ölçeği ile ölçülmüştür. Çalışmanın bulguları programlama deneyimi ile öz yeterliliği arasında bir ilişki olmadığını göstermektedir. Ek olarak Alınan programlama ders sayısı ve programlama ders ortalamaları programlama başarısının tek göstergesi olarak belirlenmiştir.

Literatür incelendiğinde öğrencilerin programlama dersi hakkındaki düşüncelerinin programlama başarılarını etkilediği sonucuna ulaşılmıştır. Bu nedenle öğrencilerin programlamaya karşı öz yeterliliklerinin programlama başarıları üzerindeki etkisi incelenme gereği duyulmuştur. Öğrenmenin verimliliğini etkileyen bir başka unsur olarak öğrenme stilleri dikkat çekmektedir. Öğrenme stilleri ile ilgili çalışmaların genellikle mühendislik öğrencileri ile yapıldığı görülmektedir. Bu çalışmalarda Kolb öğrenme stillerine göre ayrıştırma öğrenme stiline sahip olan öğrencilerin programlama dersinde daha başarılı olduğu sonucuna ulaşılmıştır. Aynı sonuçların eğitim fakültesi öğrencilerinde de alınıp alınmayacağının incelenmesinin bu alana katkı sağlayacağı düşünülmüştür. Öğrencilerin programlama yeteneğinin önceden

tahmin edilmesine yönelik birçok çalışma yapılmıştır. Fakat bu çalışmalar sonucunda programlama yeteneğini etkili bir şekilde tahmin edebilecek bir test geliştirilememiştir. Bu çalışmada daha önce az sayıda çalışmada kullanılan Kent Üniversitesi Kariyer Danışma Biriminin hazırladığı Bilgisayar Programlama Yetenek Testi (BPYT) kullanılarak bu testin geçerliliğine katkı sağlamaya çalışılmıştır. Programlama sürecinde problemin kavranması ve çözüm için alt parçalara ayrılarak analiz edilmesi gerekmektedir. Bu gereklilik mantıksal düşünme yeteneğinde problemin sahip olduğu tüm alt parçaları, fikirleri ve sonuçları zincirleme bir şekilde düzenleme gerekliliği ile paralellik göstermektedir. Programlama süreci ve mantıksal düşünme yeteneğindeki bu benzerlikten dolayı mantıksal düşünme yeteneği programlama başarısında önemli bir unsur olarak görülerek çalışma kapsamına alınmıştır.

Tüm bu araştırma ve gerekçelerden dolayı çalışmada programlama yeteneği, mantıksal düşünme yeteneği, programlama öz yeterliliği ve öğrenme stilleri ile programlama başarısı arasındaki ilişki incelenmiştir.

## ÜÇÜNCÜ BÖLÜM

### 3 YÖNTEM

Bu bölümde, araştırmanın modeli, çalışma grubunun yapısı, veri toplama araçları ve süreci ve kullanılan analiz yöntemleri detayları ile anlatılmıştır.

#### 3.1 Araştırma Modeli

Araştırma modeli, araştırma sürecinde kullanılacak verilerin toplanması ve çözümlenmesi için, araştırma amacına uygun olarak gerekli koşulların düzenlenmesidir. Bu koşulların düzenlenmesinde tarama ve deneme olmak üzere iki temel yaklaşım vardır (Karasar, 2017, s. 108). Bu çalışma tarama modeline göre modellenmiştir. Araştırmada birden fazla bağımsız değişkenin bir bağımlı değişkenle olan ilişkisi araştırılacağından ilişkisel tarama modeli kullanılmıştır. Çalışmada mantıksal düşünme yeteneği, programlama yeteneği, öğrenme stilleri, programlama öz yeterliliği, programlama deneyimi ve bilgisayar deneyiminin programlama başarısına etkisi incelenmiştir.

Tarama modeli, var olan bir durumu hiç bir şekilde değiştirme ya da etkileme çabası olmadan olduğu gibi tespit etmeyi amaçlayan araştırma modelidir. Tarama modeli ile geçmişteki ya da halen var olan bir olay araştırılabilir. Tarama modelinin veri bulma ve verileri kontrol etme gücü gibi iki temel sınırlılığı vardır (Karasar, 2017, s. 109).

Tarama modelini, genel tarama modelleri ve örnek olay tarama modelleri olarak iki kategoriye ayırabiliriz. Genel tarama modelleri, çok sayıda elemandan oluşan bir evren ya da onun bir bölümü üzerinde genel bir kanıya varmak için yapılan tarama düzenlemeleridir (Karasar, 2017, s. 111). Örnek olay tarama modelleri, evrendeki belli bir ünitenin (birey aile, hastane, dernek ve benzerleri) kendi içinde ve çevresiyle olan ilişkilerin detayları ile araştırılarak tanımlanmasını amaçlayan tarama düzenidir (Karasar, 2017, s. 119). Örnek olay taramalarında elde edilen veriler sadece incelenen olay içindir, genelleme yapılamaz.

Genel tarama modelleri, tekil tarama modeli ve ilişkisel tarama modeli olarak iki grupta incelenebilir. Tekil tarama modelleri, değişkenlerin tek tek incelendiği modellerdir. İlişkisel tarama modelleri ise, birden çok sayıdaki değişken arasındaki

ortak deęişimin varlığını ya da derecesini belirlemeyi arařtıran modellerdir (Karasar, 2017, ss. 111–114).

Tarama yönteminde bulunan iliřki gerçek bir neden sonuç iliřkisini belirtmez, fakat ipucu vererek bir olgunun bilinmesi halinde dięeri hakkında kestirimde bulunulmasına olanak saęlar (Akdaę, 2015, s. 13; Karasar, 2017, s. 114). İliřkisel tarama modelleri korelasyon türü ve karřılařtırma türü olarak iki grupta incelenebilir.

Korelasyon türünde deęişkenlerin birlikte deęişim katsayısı, korelasyon katsayısı, belirlenir. Kullanılacak korelasyon analiz yöntemi deęişkenlerin türüne göre belirlenir. Aralıklı ya da oran ölçeğinde olan iki deęişken için Pearson Momentler Çarpımı ( $r$ ), sıralama ölçeğinde olan iki deęişken arasındaki korelasyon ölçülürken Spearman Brown Sıra Farkları ( $r_s$ ), iki kategorili süreksiz bir deęişken ile en az aralık ölçeğinde olan iki deęişken arasındaki korelasyon ölçülürken Nokta Çift Serili ( $r_{pb}$ ) ve iki kategorili süreksiz iki deęişken arasındaki korelasyon ölçülürken Dörtlü (PHI) korelasyon katsayıları kullanılır (Büyüköztürk, Çokluk ve Köklü, 2015, ss. 91–105).

Karřılařtırma türü, deneme modeline benzeyen fakat deneme yapılmayan bir türdür. Amacı, belli bir durumun oluřma nedenini bire indirmektir. Karřılařtırma türünde en az iki deęişken vardır eldeki veriler sınanmak istenen deęişkene yani bağımsız deęişkene göre gruplara ayrılır ve bağımlı deęişken ile aralarında bir deęişme olup olmadığı incelenir. Korelasyon türündeki gibi sayısal bir sonuç elde edilmez. Karřılařtırma türünde elde edilen sonuç gruplar arası fark vardır ya da yoktur şeklinde belirtilir (Karasar, 2017, ss. 106–108).

Çalıřma iliřkisel tarama modeli ve korelasyon türüne göre modellenmiřtir.

### **3.2 Çalıřma Grubu**

Arařtırmanın evrenini temel (bařlangıç) seviyede programlama dersi alan öęrenciler oluřturmaktadır. Arařtırmada olasılıksız örnekleme yönteminden uygun örnekleme yöntemi ile örneklem belirlenmiřtir (Aziz, 2010, s. 54). Arařtırmada örneklem olarak Ondokuz Mayıs Üniversitesi Eęitim Fakültesi Bilgisayar ve Öęretim Teknolojileri Eęitimi bölümü ikinci sınıf Programlama Temelleri dersini alan öęrenciler oluřturmaktadır. Dersi sečen 100 öęrenci arasından vize ve final sınavlarına giren toplam öęrenci sayısı 86 'dır. Çalıřmada ölçeklerden herhangi birinin uygulandıęı

67, tüm ölçekleri uygulayan 25'i erkek 15'i kadın toplam 40 öğrenci vardır. Tablo 1 de uygulanan ölçeklere katılım sayıları verilmiştir.

**Tablo 1** Ölçeklerin uygulandığı öğrenci sayıları

	<b>Yetenek Testi</b>	<b>MDYT</b>	<b>ÖSE</b>	<b>Programlama Öz Yeterliliği</b>	<b>Başarı Sınavı</b>	<b>Tüm Ölçekler</b>
<b>Geçerli</b>	61	56	50	56	64	40
<b>Eksik</b>	6	11	17	11	3	27
<b>Toplam</b>	67	67	67	67	67	67

### 3.3 Veri Toplama Araçları

Çalışmada kullanılan ölçme araçlarının detayları bu başlık altında belirtilmiştir.

#### 3.3.1 Programlama Yetenek Testi

Öğrencilerin programlama yeteneklerini tespit etmek için Kent Üniversitesi Kariyer Danışma Biriminin hazırladığı Bilgisayar Programlama Yetenek Testi (BPYT) ([EK 1](#)) kullanılmıştır. 26 sorudan oluşan bu testte sorular, mantıksal düşünme ve problem çözme, desen ve söz dizimi tanıma ve karmaşık işlemleri takip edebilme yeteneğini ölçmektedir. Bu test herhangi bir programlama bilgisi gerektirmeden cevaplanabilmektedir. Testin Türkçe 'ye çevrilmesi Yiğit (2016) tarafından yapılmıştır. Testin güvenirlik seviyesini belirlemek için 42 öğrenciye uygulanmış ve güvenirlik katsayısı 0,65 olarak bulunmuştur. Güvenirliği arttırmak için madde-toplam korelasyonu 0,2 den düşük olan, 1., 2., 4. ve 6. maddeler testten çıkartılmıştır. Bu işlemden sonra güvenirlik katsayısı 0,71 olarak bulunmuştur (Yiğit, 2016). Testin uygulanması için 40 dakika süre belirlenmiştir. Teste 26 soru üzerinden 19 ve üzeri doğru yapan öğrenci yüksek programlama yeteneğine, 13-18 doğru yapan öğrenci orta düzey programlama yeteneğine 12 ve aşağısında doğru yapan öğrenci düşük programlama yeteneğine sahip kabul edilmektedir ("Computer Programming Aptitude Test", t.y). Çalışmada test sonuçları 0 ile 100 arasında puanlanmıştır.

#### 3.3.2 Mantıksal Düşünme Yetenek Testi

Öğrencilerin mantıksal düşünme yeteneklerini belirlemek için Mantıksal Düşünme Yetenek Testi (MDYT) ([EK 2](#)) kullanılmıştır. Orijinali Tobin ve Capie (1981) tarafından geliştirilen testin Türkçe 'ye uyarlaması Geban, Aşkar ve Özkan (1992)

tarafından yapılmıştır. Testin Türkçe versiyonunun güvenilirlik katsayısı 0,77 olarak bulunmuştur. Testte toplam 10 soru bulunmaktadır. İlk 8 soru çoktan seçmeli sorulardan oluşurken son iki soruda ise verilen boşluğa cevabın yazılması istenmektedir. Testte her biri iki soru olmak üzere 5 mantıksal işlemi ölçen 10 soru bulunmaktadır. Bu mantıksal işlemler; değişkenleri kontrol etme, orantısal düşünme, ilişkisel düşünme, birleşik düşünme ve olasılıklı düşünmedir (Güler, 2010; Tobin ve Capie, 1981). İlk 8 soruda cevap seçeneklerine ek olarak cevap açıklamaları da verilmiştir. Bir sorunun doğru sayılması için cevabın ve cevap açıklamasının doğru olarak seçilmesi gerekmektedir. Son iki soruda ise her hangi bir eksik bulunması durumunda puan verilmemektedir. Her soru bir puan olarak puanlanmaktadır. Testten 0-3 puan alan öğrenci somut düşünme düzeyinde, 4 - 6 puan alan öğrenci geçiş döneminde ve 7-10 puan alan öğrenci soyut operasyon döneminde kabul edilmektedir (Bilgin ve Ateş, 2004; Güler, 2010). Çalışmada test sonucu 0 ile 10 arasında değerlendirilecektir.

### **3.3.3 Programlamaya İlişkin Öz Yeterlilik Ölçeği**

Çalışmada öğrencilerin programlama öz yeterlilik seviyelerini belirlemek için Programlamaya İlişkin Öz Yeterlilik Ölçeği (EK 4) kullanılmıştır. Ölçek ilk olarak Ramalingam ve Wiedenbeck (1998) tarafından geliştirilmiştir. Geliştirilen ölçeğin güvenilirlik kat sayısı 0,98 olarak hesaplanmıştır. Ölçeğin Türkçeye uyarlanması Altun ve Mazman (2012) tarafından yapılmıştır. Orijinali 32 madde 4 faktörden oluşan ölçek Türkçe 'ye uyarlandığında 9 madde ve iki faktörden oluştuğu bulunmuştur. Türkçe Ölçeğin güvenilirlik kat sayısı 0,928 olarak bulunmuştur (Altun ve Mazman, 2012). Ölçek 7 'li likert tipindedir. Cevaplar 1 den 7 ye doğru, "Kendime hiç güvenmiyorum", "Genellikle güvenmiyorum", "Biraz güveniyorum", "Yarı yarıya güveniyorum", "Oldukça güveniyorum", "Genellikle güveniyorum", "Tamamen güveniyorum" şeklinde isimlendirilmiştir. Ölçek maddelere verilen cevapların toplamı şeklinde 9 ile 63 puan arasında değerlendirilmektedir.

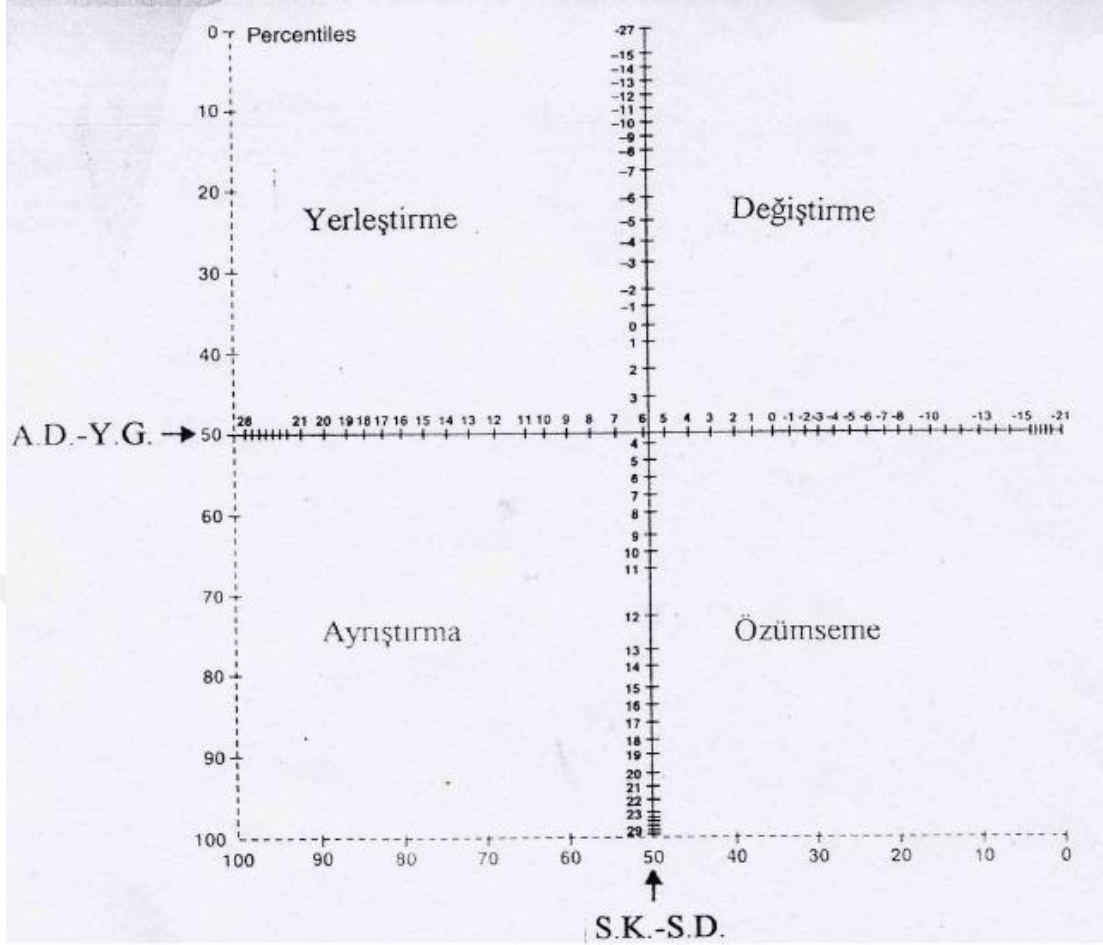
### **3.3.4 Kolb Öğrenme Stilleri Envanteri III**

Çalışmada Öğrencilerin hangi öğrenme stiline sahip olduğunu belirlemek için Kolb Öğrenme Stilleri Envanteri III (EK 3) kullanılmıştır. Ölçeğin üçüncü sürümü 1999 yılında hazırlanmıştır. Üçüncü sürümün Türkçe 'ye uyarlanması Gencel (2006) tarafından yapılmıştır. Ölçekte cümle tamamlama şeklinde 12 soru bulunmaktadır.



Her bir soruda bulunan seçenekler 1 ile 4 arasında puanlanarak sıralanmaktadır. Sorulardaki her bir seçenek Kolb'un deneyimsel öğrenme kuramındaki "Soyut Kavramsallaştırma", "Somut Deneyim", "Aktif Deneyim" ve "Yansıtıcı Gözlem" öğrenme yollarını temsil etmektedir. Her öğrenme yolu en az 12 en fazla 48 puan alabilmektedir.

Öğrenme yollarının toplam puanları üzerinde yapılan işlemler ile öğrencinin birleştirilmiş puanları elde edilmektedir. Bu puanlar soyut Kavramsallaştırma (S.K.) - Somut Deneyim (S.D) ve Aktif Deneyim (A.D.) - Yansıtıcı Gözlem (Y.G.) puanlarıdır. Bu puanlar -36 ile +36 arasında değer almaktadır. S.K. - S.D. işleminden elde edilen puan pozitif ise öğrenmenin soyut, negatif ise öğrenmenin somut olduğu; A.D. - Y.G. işleminden elde edilen sonuç pozitifse öğrenmenin aktif, negatif ise öğrenmenin yansıtıcı olduğu sonucuna ulaşılmaktadır (Gencel, 2007, s. 131). Elde edilen birleştirilmiş puanlar Şekil 7 de bulunan koordinat sistemi üzerine yerleştirilerek öğrencinin öğrenme stili belirlenmektedir. Öğrencinin öğrenme stilini gösteren noktanın x bileşeni A.D. - Y.G işleminden elde edilen puan, y bileşenini ise S.K. - S.D. işlemiyle elde edilen puan ile belirlenmektedir. Bu noktanın bulunduğu alan öğrencinin öğrenme stilini belirtmektedir.



Şekil 7 KÖSE III koordinat sistemi (Kolb, 1999: 6 Akt. Gencil 2007,s. 132)

### 3.3.5 Programlama Başarı Testi

Öğrencilerin dönem sonunda programlama başarılarını ölçmek için dersi veren öğretim üyesinin hazırladığı vize ve final sınavları kullanılmıştır. Öğrencilerin başarı not ortalamaları hesaplanırken vize notunun %40 ı ile final notunun %60 'ının toplamı kullanılmıştır. Başarı not ortalaması 60 'ın altında olan öğrenciler başarısız olarak kabul edilmiştir.

### 3.4 Veri Toplama Süreci

Dönem başında öğrencilerin programlama yeteneklerini belirlemek için geliştirilen Bilgisayar Programlama Yetenek testi, öğrencilerin mantıksal düşünme yeteneklerini belirlemek için Mantıksal Düşünme Yetenek Testi ve öğrencilerin nasıl öğrendiklerini belirlemek için Kolb'un Öğrenme Stilleri Envanteri III testleri öğrencilere uygulanacaktır. Dönem sonuna yaklaşıldığında öğrencilerin programlama öz yeterliliklerini tespit etmek için Programlama Öz Yeterlilik Ölçeği uygulanmıştır.

Öz yeterlilik ölçeğine ek olarak öğrencilerin kaç yaşında olduğu, cinsiyeti, kaç yıldır bilgisayar kullandığı, kaç yıldır programlama yaptığı ve hangi programlama dillerini bildiği gibi veriler toplanmıştır.

### **3.5 Kullanılan Analiz Yöntemleri**

Çalışmanın bağımlı değişkeni programlama başarısı, bağımsız değişkenleri ise, mantıksal düşünme yeteneği, öğrenme stili, programlama öz yeterliliği, programlama yetenek puanı, bilgisayar deneyimi ve programlama deneyimidir.

Değişkenler arasındaki ilişkiyi belirlemek için değişkenlerin ölçek türlerine göre uygun olan korelasyon katsayısı belirlenerek korelasyon analizi yapılmıştır. Öğrenme stilleri gruplarında programlama başarısının farklılık gösterip göstermediğini belirlemek için ilişkisiz örneklemelerde tek yönlü varyans analizi kullanılmıştır. Tüm değişkenlerin programlama başarısı üzerindeki etkisini incelemek için de çoklu regresyon analizi yapılmıştır. Çalışmada yapılan analizler Sosyal Bilimler için İstatistik paket programı (SPSS) kullanılmıştır

#### **3.5.1 Korelasyon Analizi**

Korelasyon katsayısı 0 ile 1 arasında değerler alır. Korelasyon katsayısının işareti ilişkinin yönünü belirtir. Katsayının 1 olması mükemmel pozitif korelasyonu, -1 olması mükemmel negatif korelasyonu belirtir. Katsayının 0 olması hiç bir ilişkinin olmadığı anlamına gelir. Korelasyon katsayısı 0,30'dan az ise düşük düzeyde bir ilişki, 0,30 ile 0,70 arasında ise orta seviye bir ilişkiyi, 0,70'den büyük ise yüksek düzeyde bir ilişkiyi belirtmektedir (Büyüköztürk ve diğerleri, 2015, s. 92). Analizde kullanılan değişkenlerin ölçek türüne göre farklı korelasyon katsayısı formülleri vardır.

*Pearson Momentler Çarpımı korelasyon katsayısı ( r ), en az aralıklı ölçek türünde ölçülen ve sürekli değişken olan iki değişkenin doğrusal bir şekilde ne kadar tutarlı değiştiğini incelemek için kullanılır. Pearson Momentler çarpımını kullanmanın ön koşulu iki değişkeninde normal dağılım gösteriyor olmasıdır (Büyüköztürk ve diğerleri, 2015, s. 91; Can, 2017, s. 370).*

*Spearman Brown Sıra Farkları korelasyon katsayısı ( r<sub>s</sub> ); genellikle üç durumda kullanılır. Birincisi, değişkenlerin sıralama ölçeği ile ölçülmesi gerektiğinde. İkincisi,*

bir sıralama ölçeği ile ölçülmüş değişken ile en az aralık ölçeğinde ölçülen değişken karşılaştırılmak istendiğinde. İkinci durumda aralık ölçeğindeki değişken sanal sıralama ölçeğine çevrilerek işleme alınır. Üçüncüsü, en az aralık türünde olan iki değişken normallik şartını sağlamadığı durumlarda kullanılır. Bu işlem için de değişkenler sanal sıralama ölçeğine dönüştürülür (Büyüköztürk ve diğerleri, 2015, s. 95).

*Nokta Çift Serili korelasyon katsayısı ( $r_{pb}$ )*; iki kategorili gerçek süreksiz bir değişken ile en az aralık ölçeğinde ölçülmüş bir değişken arasındaki ilişki ölçülmek istendiğinde kullanılan korelasyon formülüdür (Büyüköztürk ve diğerleri, 2015, s. 99). Pearson momentler çarpımı korelasyon katsayısının özel bir halidir ve SPSS de pearson hesaplaması ile aynı şekilde yapılır (Can, 2017, s. 377).

*Dörtlü korelasyon katsayısı (PHI)*; iki kategorili süreksiz iki değişken arasındaki korelasyon ölçülürken kullanılan korelasyon katsayısıdır (Büyüköztürk ve diğerleri, 2015, s. 105).

### **3.5.2 Regresyon**

Birbiri ile ilişkili olan değişkenlerden birinde meydana gelen değişikliğin diğerinde ya da diğerlerinde ne düzeyde bir değişiklik meydana getireceğini tahmin etmek için kullanılan analiz yöntemidir. Regresyon kelime anlamı ile bir değişkenin değerinin değişkenin ait olduğu grubun ortalamasına doğru çekilmesidir. Regresyon analizi ise bağımlı değişken ile bağımsız değişkenler arasındaki ilişkinin matematiksel eşitlik ile açıklanması sürecidir (Büyüköztürk ve diğerleri, 2015, s. 122).

Korelasyon analizi değişkenler arasındaki ilişkiyi neden sonuç bağı olmadan gösterirken, regresyon analizi değişkenler arasındaki neden sonuç ilişkisini gösterir (Eymen, 2007, s. 92).

Regresyon analizi, bağımlı değişkenin bir bağımsız değişken tarafından tahmin ediliyorsa basit doğrusal regresyon, birden fazla bağımsız değişken kullanılarak tahmin ediliyorsa çoklu regresyon adını alır. Regresyon analizinde tüm değişkenlerin en az aralık ölçeğinde olması beklenir. Fakat bazı durumlarda sınıflama ve sıralama ölçeklerinin kullanılması gerekebilir. Bu durumda sıralamalı veriler için logaritmik dönüşüm yapılır. Sınıflama verilerinde ise temsili değişken (Dummy Variable)

olarak isimlendirilen yapay bir deęişkene dönüştürerek regresyon analizi yapılmalıdır (Büyüköztürk ve dięerleri, 2015, s. 125; Can, 2017, s. 277). Temsili deęişken oluşturulurken kategorilerden biri dışarıda bırakılarak kategori sayısını bir eksięi kadar kulla deęişken oluşturulur. Oluşturulan kukla deęişkende 1 deęeri temsil ettięi kategoriye ait olmayı, 0 ise dięer kategorilere ait olmayı temsil eder. Bu çalışmada Öğrenme Stilleri deęişkeni için oluşturulan temsili deęişkenlerde Özümseme stili dışarıda tutularak üç tane temsili deęişken oluşturulmuştur. Temsili deęişken ile ilgili yorumlar yapılırken dışarıda bırakılan kategori temel alınarak yapılır (Büyüköztürk, 2011, s. 92).Örneęin Ayırıştırma stilinin regresyon analizinin baęımlı deęişkeni olan programla başarısı ile pozitif ilişkide olması Ayırıştırma stiline sahip olmanın programlama başarılarına katkısı Özümseme stiline sahip olanlara göre daha yüksek olduğunu, negatif olması ise Özümseme stiline sahip olmanın Ayırıştırma stiline sahip olmaktan daha fazla katkı sağladığını göstermektedir.

### **3.5.3 İlişkisiz Örneklemeler İçin Tek Yönlü Varyans Analizi**

İkiden fazla grupta, en az aralıklı ölçek ile ölçülmüş bir ölçüm sonucunun gruplara arasında farklılaşıp farklılaşmadığını ölçmek için İlişkisiz Örneklemeler için Tek Yönlü Varyans Analizi (Tek Yönlü ANOVA) kullanılır. ANOVA grupların en az ikisi arasında istatistiksel olarak anlamlı bir fark olup olmadığını test eder. Bu test için “ortalamlar arasında anlamlı bir fark yoktur.” şeklindeki yokluk hipotezini ( $H_0$ ) sınanır. ANOVA normal dağılım gösteren veriler ile yapılmalıdır. Ancak ANOVA ‘nın güçlü bir analiz olduğu göz önüne alındığında normal dağılım göstermeyen veriler ile de kullanılabilir (Büyüköztürk ve dięerleri, 2015, ss. 178–179).

## DÖRDÜNCÜ BÖLÜM

### 4 BULGULAR

Bu bölümde araştırma problemlerine yönelik yapılan analizler sonucunda elde edilen bulgulara yer verilecektir.

#### 4.1 Programlama Yetenek Puanı ile Programlama Başarısı Arasında Bir İlişki Var Mıdır?

Bu araştırma problemini test etmek için araştırma hipotezi ( $H_A$ ), “Programlama yetenek puanı ile programlama başarısı arasında ilişki vardır.” şeklinde belirlenmiştir. Bu durumda test edilecek olan yokluk hipotezi ( $H_0$ ), “Programlama yetenek puanı ile programlama başarısı arasında ilişki yoktur.” şeklinde olacaktır. Programlama yetenek testi ile programlama başarı testinden elde edilen verileri analiz etmek için kullanılacak korelasyon katsayısı verilerin normallik analizi sonuçlarına göre belirlenecektir. Analizin sonuçları Tablo 2 de gösterilmiştir.

**Tablo 2** Programlama yetenek ve programlama başarı testi normallik analizi

	Kolmogorov-Smirnov <sup>a</sup>			Shapiro-Wilk		
	Statistic	df	p	Statistic	df	p
<b>PBP</b>	0,076	58	0,200	0,971	58	0,187
<b>PYP</b>	0,134	58	0,011	0,961	58	0,061

Analiz sonucunda iki farklı test sonucu gösterilmektedir. Kolmogorov-Smirnov ve Shapiro-Wilk. Büyüköztürk ve diğerlerine (2015, s. 156) göre grup büyüklüğü 50 ve üzeri ise Kolmogorov-Smirnov, küçük ise Shapiro-Wilk test sonuçları dikkate alınmalıdır. Burada grup büyüklüğü 58 olduğu için K-S sonuçları dikkate alınmıştır. K-S anlamlılık değeri 0,05 den küçük ise verilerin normal dağılım göstermediği yorumu yapılır (Can, 2017,s. 89). K-S anlamlılık değerlerine göre programlama başarı puanları normal dağılım gösterirken, programlama yetenek puanları normal dağılım göstermemektedir (Büyüköztürk ve diğerleri, 2015, s. 156; Can, 2017, s. 89). Veriler normal dağılım göstermediği için parametrik olmayan analiz yöntemlerinden Spearman Brown Sıra Farkları korelasyon katsayısı ( $r_s$ ) kullanılarak ilişki analiz edilecektir. Tablo 3 de korelasyon analizi sonuçları gösterilmiştir. Değişkenler sıralama ölçeğine dönüştürülerek analiz yapılmıştır.

**Tablo 3** Programlama yetenek puanı ve başarı puanı korelasyon sonuçları

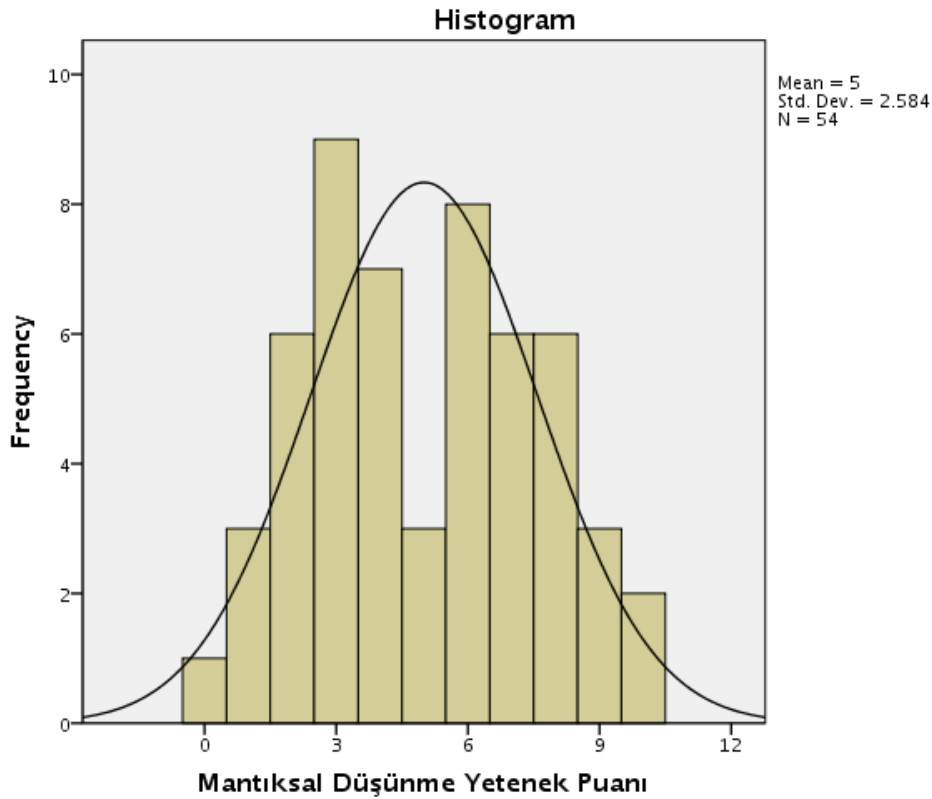
		<b>PYP</b>
<b>PBP</b>	Spearman's rho	.244
	p (2-tailed)	.065
	N	58

Yetenek Puanı ile Programlama Başarısı arasında ilişki olup olmadığını ortaya koymak için yapılan Spearman Sıra Farkları korelasyon testi, bu iki değişken arasında anlamlı bir ilişkinin olmadığını göstermektedir ( $r_s=0.241$ ,  $p>0.05$ ).

Yetenek testinin öğrencilerin programlama başarılarını önceden tespit edebildiğine yönelik öngörünün doğru olmadığı sonucuna ulaşılmıştır.

#### 4.2 Mantıksal Düşünme Yeteneği ile Programlama Başarısı Arasında İlişki Var mıdır?

Mantıksal düşünme yeteneği testinden alınan puanların ortalaması 5 olarak hesaplanmıştır. Katılımcıların %35 (19)'ü somut düşünce düzeyinde, %33 (18)'ü geçiş döneminde, %32 (17)'si soyut operasyon döneminde. Katılımcıların puan



Şekil 8 Mantıksal düşünme yetenek testi puan grafiği

grafiği Şekil 8 de verilmiştir.

Testten 0-3 puan alan öğrenci somut düşünme düzeyinde, 4-6 puan alan öğrenci geçiş döneminde ve 7-10 puan alan öğrenci soyut operasyon döneminde kabul edilmektedir

Araştırmanın bu alt problemi için incelenecek olan hipotez ( $H_A$ ) “Mantıksal düşünme yeteneği ile programlama başarısı arasında ilişki vardır.” şeklinde belirlenmiştir. SPSS ile sınanacak olan yokluk hipotezi ( $H_0$ ) ise “Mantıksal düşünme yeteneği ile programlama başarısı arasında ilişki yoktur.” şeklinde belirlenmiştir. Bu hipotezlerin sınanmasında kullanılacak analiz yöntemini seçmek için ilk önce normallik analizi yapılmalıdır. Tablo 4 de mantıksal düşünme yeteneği puanı ile programlama başarı puanı normallik analizi sonuçları verilmiştir.

**Tablo 4** Mantıksal düşünme yetenek ve programlama başarı testleri normallik analizi

	Kolmogorov-Smirnov <sup>a</sup>			Shapiro-Wilk		
	Statistic	df	p	Statistic	df	p
<b>MDYP</b>	.132	54	.019	.959	54	.065
<b>PBP</b>	.091	54	.200*	.967	54	.149

Tablo 4 de Mantıksal Düşünme Yeteneği Puanı için Kolmogorov-Smirnov testinin anlamlılık düzeyi 0,05 den küçük olduğu için test normal dağılıma sahip olmadığı görülmektedir. Bu nedenle iki verinin karşılaştırılması için parametrik olmayan korelasyon yöntemlerinden Spearman Brown Sıra Farkları korelasyon katsayısı ( $r_s$ ) kullanılmıştır. Bu işlem için iki değişken de sanal sınıflama ölçeğine dönüştürülerek analiz yapılmıştır. Tablo 5 de yapılan analizin sonuçları verilmiştir.

**Tablo 5** Mantıksal düşünme yeteneği ve programlama başarısı arasındaki korelasyon

		<b>PBP</b>
	Spearman's rho	.241
<b>MDYP</b>	p (2-tailed)	.079
	N	54

Analiz sonucuna göre MDYP ile PBP arasında pozitif yönde düşük seviyede bir ilişki vardır. Fakat analizin anlamlılık değeri 0,05 den büyük olduğu için elde edilen sonuç istatistiksel olarak anlamlı değildir. Bu durumda “Mantıksal düşünme yeteneği



ile programlama başarısı arasında bir ilişki yoktur.” hipotezi kabul edilmiş olmaktadır

#### 4.3 Öğrencilerin Programlama Öz Yeterlilik Algıları ile Programlama Başarısı Arasında Bir İlişki Var mıdır?

Programlama öz yeterlilik algısı ve programlama başarısı verilerinin normallik analizi sonuçları Tablo 6 da verilmiştir. Bu sonuçlara göre öz yeterlilik ve başarı verilerinin her ikisi de normal dağılım göstermektedir. Bu durumda bu veriler parametrik analizler yapmak için uygundur. Bu araştırma problemi için araştırma hipotezi ( $H_A$ ) “Programlama Öz yeterlilik algısı ile Programlama başarısı arasında ilişki vardır.” olarak belirlenmiştir.

**Tablo 6** Programlama öz yeterliliği ve programlama başarısı normallik analizi

	Kolmogorov-Smirnov <sup>a</sup>			Shapiro-Wilk		
	Statistic	df	p	Statistic	df	p
<b>PÖYA</b>	.076	55	.200*	.979	55	.431
<b>PBP</b>	.085	55	.200*	.971	55	.209

Öğrencilerin programlama başarıları ile programlama öz yeterlilik algıları arasında bir ilişki olup olmadığını incelemek için Pearson Momentler Çarpımı korelasyon katsayısı hesaplanmıştır. Tablo 7’de de gözüktüğü gibi Programlama Öz Yeterliliği ile Programlama başarısı arasında pozitif yönde orta seviyede bir ilişki bulunmaktadır. Yapılan analiz sonucunda anlamlılık seviyesi 0,01 den küçük olduğu için oluşturulan araştırma hipotezi ( $H_A$ ) kabul edilmiştir.

**Tablo 7** Öz yeterlilik ile programlama başarısı arasındaki ilişki

		Programlama Başarısı
	Pearson Correlation	.436
<b>Öz yeterlilik puanı</b>	p (2-tailed)	.001
	N	55

#### 4.4 Öğrenme Stillere Göre Öğrencilerin Programlama Başarısı Farklılık Göstermekte Midir?

Dört farklı öğrenme stiline sahip öğrencilerin programlama başarıları arasında fark olup olmadığını sınamak için öğrenme stillerine göre oluşturulan grupların programlama başarı ortalamaları ilişkisiz örneklem için tek yönlü varyans analizi

ile karşılaştırılmıştır. Analizden önce tüm gruplarda programlama başarı puanının normal dağılım gösterdiği belirlenmiştir. ANOVA testine dâhil olan verilerin açıklayıcı istatistikleri Tablo 8’de verilmiştir.

**Tablo 8** Öğrenme stilleri gruplarına göre programlama başarısı açıklayıcı istatistikleri

<b>Programlama Başarısı</b>								
	N	Ortalama	Standart Sapma	Standart Hata	95% Olasılıkla Ortalamaların Güven aralıkları		Min	Max
					Alt Sınır	Üst Sınır		
<b>Ayrıştırma</b>	15	68.760	13.52695	3.49264	61.2690	76.2510	45.80	88.60
<b>Değiştirme</b>	11	53.182	12.39143	3.73616	44.8571	61.5065	30.80	70.00
<b>Özümseme</b>	16	52.175	19.53115	4.88279	41.7676	62.5824	15.80	74.60
<b>Yerleştirme</b>	8	61.125	12.01710	4.24869	51.0785	71.1715	44.00	76.00
<b>Total</b>	50	58.804	16.54355	2.33961	54.1024	63.5056	15.80	88.60

En yüksek mevcuda sahip olan öğrenme stili 16 mevcut ile Özümseme stilidir. En az mevcudu olan stil ise 8 mevcut ile Yerleştirme stilidir. Ayrıştırma stili en yüksek ortalama ( $X_{Ayrıştırma} = 68.760$ ) sahip stildir. En düşük ortalama ise Özümseme stiline sahip öğrencilerin ortalamasıdır ( $X_{Özümseme} = 52,175$ ). Grup ortalamalarına bakıldığında öğrenme stillerinin programlama başarısı sıralaması, en başarılı Ayrıştırma, ikinci sırada Yerleştirme, üçüncü sırada Değiştirme ve son sırada Özümseme öğrenme stili bulunmaktadır. Grupların varyanslarının eşitliğini kontrol etmek için yapılan Levene testinde anlamlılık düzeyi ( $p=0,353$ ) 0,05 den büyük olduğu için grupların varyansları arasında istatistiksel fark olmadığı sonucuna ulaşılmıştır (Can, 2017, s. 154).

Tablo 9 da gösterilen sonuçlara göre Öğrenme stilleri gruplarından en az ikisi arasında istatistiksel olarak anlamlı fark olduğu tespit edilmiştir ( $F_{(3 - 46)} = 3,654, p < 0,05$ ).

**Tablo 9** Öğrenme stilleri grupları ANOVA sonucu

<b>Programlama Başarısı</b>					
	<b>Kareler Toplamı</b>	<b>Sd</b>	<b>Kareler Ortalaması</b>	<b>F</b>	<b>p</b>
<b>Between Groups</b>	2580.722	3	860.241	3.654	.019
<b>Within Groups</b>	10830.037	46	235.436		
<b>Total</b>	13410.759	49			

Hangi ortalamalar arasında fark olduğunu test etmek için kullanılan farklı testler bulunmaktadır. Grup varyanslarının eşit olduğu ve grup örneklem sayıları arasında fark olduğu zaman Scheffe testi tercih edilmelidir (Can, 2017, s. 152). Hangi ortalamalar arasında fark olduğunu belirlemek için Scheffe çoklu karşılaştırmalar testi sonucu Tablo 10 'da gösterilmiştir.

**Tablo 10** Öğrenme stilleri gruplarının programlama başarısı ortalaması Scheffe testi sonucu

<b>Bağımlı değişken: Programlama Başarısı</b>						
<b>(I) Öğrenme Stili</b>	<b>(J) Öğrenme Stili</b>	<b>Ortalama Farkları (I-J)</b>	<b>Standart Hata</b>	<b>p</b>	<b>95% Güven Aralığı</b>	
					<b>Alt sınır</b>	<b>Üst Sınır</b>
	Değiştirme	15.57818	6.09089	.103	-2.0965	33.2528
<b>Ayrıştırma</b>	Özümseme	16.58500*	5.51457	.039	.5827	32.5873
	Yerleştirme	7.63500	6.71753	.732	-11.8580	27.1280
	Ayrıştırma	-15.57818	6.09089	.103	-33.2528	2.0965
<b>Değiştirme</b>	Özümseme	1.00682	6.00982	.999	-16.4326	18.4462
	Yerleştirme	-7.94318	7.12970	.744	-28.6323	12.7459
	Ayrıştırma	-16.58500*	5.51457	.039	-32.5873	-.5827
<b>Özümseme</b>	Değiştirme	-1.00682	6.00982	.999	-18.4462	16.4326
	Yerleştirme	-8.95000	6.64411	.615	-28.2300	10.3300
	Ayrıştırma	-7.63500	6.71753	.732	-27.1280	11.8580
<b>Yerleştirme</b>	Değiştirme	7.94318	7.12970	.744	-12.7459	28.6323
	Özümseme	8.95000	6.64411	.615	-10.3300	28.2300

Çoklu karşılaştırmalar testinde kıyaslanan grupların ortalamaları arasındaki farkın anlamlılık test sonucu 0,05 den büyük ise anlamlı fark yok, küçük ise anlamlı fark var şeklinde yorumlanır (Can, 2017, s. 156). Tablo 10 incelendiğinde p değeri 0,05

den küçük olan gruplar Araştırma ve Özümseme gruplarıdır ( $p=0,039$ ). Bu sonuç Ayırıştırma ile Özümseme stilleri arasındaki farkın istatistiksel olarak anlamlı olduğunu göstermektedir. ANOVA sonucu gruplar arasındaki ortalamaların farklı olup olmadığı bilgisini verir fakat bu farkın büyüklüğü hakkında bilgi vermez. Aradaki farkın büyüklüğünü, Anova sonucundaki gruplar arası varyansın toplam varyansa bölünmesi ile elde edilen eta-kare ( $\eta^2$ ) katsayısı ile ifade edilir. Eta-kare 0 ile 1 arasında değer alır, 0,01 değeri küçük, 0,06 değeri orta, 0,14 değeri geniş etki büyüklüğü olarak yorumlanır (Can, 2017, s. 157). Test sonucu hesaplanan etki büyüklüğü ( $\eta^2 = 0,19$ ) gruplar arası farkın geniş düzeyde olduğunu göstermektedir.

#### 4.5 Bilgisayar Deneyimi ve Programlama Deneyiminin Programlama Başarısına Bir Etkisi Var Mıdır?

Bu araştırma problemi için öz yeterlilik ölçeği içerisinde toplanan “Kaç yıldır bilgisayar kullanıyorsunuz?” ve “Kaç yıldır program yazıyorsunuz?” sorularına verilen cevaplar kullanılmıştır. İlk olarak verilere normallik analizi yapılmış ve bu analizin sonuçlarına göre hangi korelasyon katsayısının kullanılacağı belirlenmiştir. Tablo 11 de gösterilen sonuçlara göre bilgisayar deneyiminin ve programlama deneyimi verilerinin normal dağılım göstermediği görülmektedir.

**Tablo 11** Bilgisayar deneyimi, programlama deneyimi ve programlama başarısı verilerinin normallik analizi sonuçları

	Kolmogorov-Smirnov <sup>a</sup>			Shapiro-Wilk		
	Statistic	df	p	Statistic	df	p
<b>Bilgisayar Deneyimi</b>	.196	55	.000	.948	55	.020
<b>Programlama Deneyimi</b>	.191	55	.000	.922	55	.002
<b>Programlama Başarısı</b>	.085	55	.200*	.971	55	.209

Bu değişkenler arasındaki ilişkiyi incelemek için Spearman Brown Sıra Farkları ( $r_s$ ) korelasyon katsayısı kullanılmıştır. Korelasyon hesabı yapılmadan önce SPSS yazılımı kullanılarak veriler yapay sıralama ölçeğine dönüştürülmüştür.

Analiz sonucunda Programlama Başarı puanı ile bilgisayar deneyimi arasında anlamlı bir ilişki bulunamazken ( $r_s=0,018, p=0,447$ ) programlama deneyimi ile düşük seviyede bir ilişki bulunmuştur ( $r_s=0,273, p=0,022$ ).

**Tablo 12** Bilgisayar ve programlama deneyim ile programlama başarısı arasındaki korelasyon analizi

		<b>Bilgisayar Deneyimi</b>	<b>Programlama Deneyimi</b>
	Spearman's ks	.018	.272*
<b>PBP</b>	p (1-tailed)	.447	.022
	N	55	55

**4.6 Programlama Başarısı Üzerinde Etkisi Olduğu Düşünülen, Programlama Yetenek Puanı, Mantıksal Düşünme Yeteneği, Öğrenme Stilleri ve Programlamaya İlişkin Öz yeterlilik Düzeyi Programlama Başarısını Hangi Öncelikte ve Derecede Açıklamaktadır?**

**Tablo 13** Programlama başarısının yordanmasına ilişkin çoklu regresyon analizi

<b>Değişken</b>	<b>B</b>	<b>Standart Hata B</b>	<b>β</b>	<b>T</b>	<b>p</b>	<b>İkili r</b>	<b>Kısmi r</b>
<b>Sabit</b>	16,746	13,411	-	1,249	0,221	-	-
<b>PÖYA</b>	0,527	0,232	0,321	2,272	0,030	0,402	0,368
<b>PYP</b>	0,287	0,151	0,263	1,898	0,066	0,275	0,314
<b>MDYP</b>	0,689	0,874	0,110	0,788	0,436	0,192	0,104
<b>Temsili Ayrıştırma</b>	11,528	5,350	0,355	2,155	0,039	0,459	0,286
<b>Temsili Değiştirme</b>	-3,594	5,642	-0,102	-0,637	0,529	-0,325	-0,084
<b>Temsili Yerleştirme</b>	-0,564	6,524	-0,013	-0,087	0,932	-0,067	-0,011
<b>R = 0,648</b>	<b>R<sup>2</sup> = 0,419</b>						
<b>F<sub>(6,33)</sub> 3,971</b>	<b>=</b>	<b>P = 0,004</b>					

Programlama başarısını etkilediğini düşünülen programlama öz yeterliliği, mantıksal düşünme yeteneği, programlama yetenek puanı ve öğrenme stilleri değişkenlerinin, birlikte programlama başarısının ne kadarını açıkladığını ve bu değişkenlerin programlama başarısını etkileme önem sıralamasını belirlemek için çoklu regresyon analizi yapılmıştır. Programlama Öz yeterliliği, Programlama Yetenek Puanı, Mantıksal Düşünme Yetenek Puanı ve Öğrenme Stili değişkenlerine göre Programlama Başarısının yordanmasına yönelik regresyon analizi sonucu Tablo 13 de verilmiştir. Elde edilen sonuçlara göre programlama öz yeterliliği, programlama

yetenek puanı, mantıksal düşünme yetenek puanı ve öğrenme stili birlikte öğrencilerin programlama başarıları ile anlamlı bir ilişki ( $R = 0,648$ ,  $R^2 = 0,419$ ) göstermektedir ( $F_{(6,33)} = 3,971$ ,  $p < 0,01$ ). Değerlendirilen bu 6 değişken programlama başarısındaki değişimin %41'ini açıklamaktadır.

Standartlaştırılmış regresyon katsayılarına göre programlama başarısı üzerindeki görece önem sırası, öğrenme stili ( $\beta = 0,355$ ), programlama öz yeterliliği ( $\beta = 0,321$ ), programlama yetenek puanı ( $\beta = 0,263$ ), mantıksal düşünme yetenek puanı ( $\beta = 0,110$ ) şeklindedir.



## BEŞİNCİ BÖLÜM

### 5 TARTIŞMA VE SONUÇ

Bu bölümde çalışma kapsamında toplanan veriler ile yapılan analizlerden elde edilen sonuçlar literatür ile harmanlanarak yorumlanacaktır. Son olarak çalışmada elde edilen sonuçlar doğrultusunda daha iyi bir programlama eğitimi için öneriler yazılacaktır.

Öğrencilerin büyük bir bölümünün programlama dersinde başarısız olması problemini temel alarak yapılan bu çalışmada,

- Bilgisayar başarısını önceden tahmin etmek için geliştirilen programlama yeteneği
- Bilgisayar programlamada önemli bir işlevi olan mantıksal düşünme yeteneği
- Öğrencilerin sahip olduğu öğrenme stilleri
- Öğrencilerin programlama öz yeterlilik seviyeleri
- Bilgisayar ve programlama deneyimi

gibi faktörlerin programlama dersi başarısına etkileri araştırılmıştır.

Çalışmada ele alınan alt problemler için yapılan analizlerden elde edilen sonuçlar başlıklar halinde açıklanacaktır.

#### 5.1 Programlama Yetenek Puanı ile Programlama Başarısı Arasında Bir İlişki

##### Var Mıdır? Problemine İlişkin Sonuçlar

Programlama başarısı ile programlama yeteneği arasındaki ilişkiyi incelemek için yapılan korelasyon analizine göre Programlama Yetenek testinden alınan puanlar ile öğrencilerin programlama başarısı arasında anlamlı bir ilişki bulunamamıştır ( $r_s=0.241, p>0.05$ ). Bu sonuç kullanılan programlama yetenek testinin programlama başarısını doğru bir şekilde yordayamadığını göstermektedir. Literatürde yetenek testlerinin başarıyı ölçmede kullanışlı olmadığını belirten çalışmalar (Dehnadi, 2006; Jenkins, 2002) elde edilen sonucu desteklemektedir. Literatür incelendiğinde elde edilen sonuçtan farklı olarak yetenek testinin programlama başarısını doğru

yordadığı sonucuna ulaşan çalışmalar bulunmaktadır. Lacher ve Lewis (2015, s. 226) yaptığı çalışmada öğrencilerin programlama yeteneklerini ölçmek için Kent Üniversitesi Programlama yetenek testini kullanarak elde ettiği sonuçların öğrencilerin programlama başarı puanları ile yüksek korelasyona sahip olduğunu belirtmiştir. Jenkins (2002) ise çalışmasında yetenek testleri ile başarı puanları arasında ilişki bulunmayan çalışmalara vurgu yaparak, programlama dersini zorlaştıran bilişsel nedenlere odaklanmak gerektiğini vurgulamaktadır.

## **5.2 Mantıksal Düşünme Yeteneği ile Programlama Başarısı Arasında İlişki Var mıdır? Problemine İlişkin Sonuçlar**

Bulgular incelendiğinde öğrencilerin %68'i soyut operasyon döneminin altında olduğu gözlemlenmektedir. Piaget'in kuramına göre soyut operasyon dönemi 11-18 yaş arasında başlamaktadır. (Yüzüak ve Dökme, 2015, s. 445). Bulgulara baktığımızda katılımcıların birçoğunun bu dönemin gerisinde kaldığı görülmektedir. Bu durum özellikle lise seviyesinde alınan eğitimin yetersiz olduğunun bir göstergesidir.

Mantıksal düşünme yeteneği ile programlama başarısı arasındaki ilişkiyi incelemek için yapılan korelasyon analizi sonuçlarına bakıldığında, korelasyon katsayısının düşük seviyede ve istatistiksel olarak anlamsız olduğu görülmektedir ( $r_s=0,240, p>0,05$ ). Literatür incelendiğinde Mantıksal düşünme yeteneği ve programlama başarısı ilişkisine odaklanmış çalışma sayısının oldukça az olduğu görülmektedir. Sebetci ve Aksu (2014), meslek yüksekokulunda Bilgisayar programcılığı bölümünde öğrenim gören öğrenciler ile yaptığı çalışmada analitik düşünme ve mantıksal düşünme yeteneklerinin programlama başarısına etkisini incelemiş ve mantıksal düşünme yeteneği ile programlama başarısı arasında orta düzeyde pozitif bir ilişki bulmuştur.

## **5.3 Öğrencilerin Programlama Öz Yeterlilik Algıları ile Programlama Başarısı Arasında Bir İlişki Var mıdır? Problemine İlişkin Sonuçlar**

Öz yeterlilik bireyin bir işte başarılı olabileceğine dair kendine olan güvenidir. Eğer birey kendine güveniyorsa karşılaştığı zorluklarla başa çıkmada daha dirençli olacaktır. Bu direnç bireyi başarıya ulaştıracaktır. Çalışmada toplanan veriler arasında programlama başarısı ile en yüksek korelasyon ilişkisi programlama öz



yeterliliği ile olduğu görülmüştür. Programlama öz yeterliliği ile programlama başarısı arasındaki ilişkiyi incelemek için yapılan korelasyon analizinde öz yeterlilik ile programlama başarısı arasında orta düzeyde ve istatistiksel olarak anlamlı bir sonuç elde edilmiştir ( $r=0,436$ ,  $p<0,01$ ).

Literatür de de benzer sonuçlar olduğu görülmektedir. Ramalingam ve diğerleri (2004) yaptıkları çalışmada acemi programcılardan programlama öz yeterliliği yüksek olanların programlamada daha başarılı olduğu sonucuna ulaşmışlardır. Bu sonuçlar programlama başarısında öz yeterliliğin önemli bir faktör olduğunu göstermektedir.

#### **5.4 Öğrenme Stillere Göre Öğrencilerin Programlama Başarısı Farklılık Göstermekte Midir? Problemine İlişkin Sonuçlar**

Bir bilgiyi öğrenmenin farklı yolları vardır. Öğrenciler kendi yapılarına göre bu yolları farklı sıklıklarda kullanarak yeni bilgileri öğrenirler (Gencel, 2006, s. 43). Her öğrencinin bireysel farklılıkları vardır ve bu farklılıklardan kaynaklı olarak farklı öğrenme stillerine sahiptirler. Kolb'un deneyimsel öğrenme kuramına göre farklı öğrenme stillerine sahip öğrencilerin hepsine hitap eden bir eğitim sunabilmek için farklı öğrenme yollarına hitap eden öğretim yöntemleri bir döngü içerisinde sırayla kullanılacak şekilde eğitim planlanmalıdır.

Çalışmada öğrenme stilleri envanterinin uygulandığı 50 katılımcıdan 16'sı Özümseme, 15 'i Ayırıştırma, 11 'i Değiştirme ve 8 'i Yerleştirme stiline sahiptir. Öğrenme stilleri üzerinde yapılan çalışmalar incelendiğinde programlama eğitimine en için en uygun öğrenme stili Ayırıştırma, öğretmenlik için en uygun stilin Özümseme stili olduğu görülmektedir. (Aşkar ve Akkoyunlu, 1993; Byrne ve Lyons, 2001). Katılımcıların öğrenme stilleri dağılımı da bu sonuçla paralellik göstermektedir.

Çalışmada katılımcılar öğrenme stiline göre gruplandırıldığında en başarılı grubun Ayırıştırma stiline sahip olan öğrenciler olduğu, en başarısız grubunda Özümseme stiline sahip öğrenciler olduğu görülmüştür. Literatürde de benzer sonuçlara ulaşılmıştır (Byrne ve Lyons, 2001). Grupların ortalamaları arasındaki farkın istatistiksel olarak anlamlı olup olmadığını belirlemek için yapılan ANOVA sonucunda Ayırıştırma ve Özümseme grupları arasındaki farkın istatistiksel olarak

anlamli ve geniř bir etkiye sahip olduđu sonucuna ulařılmıřtır. Bu sonu programlama eđitiminde zmsemeye stiline hitap eden đretim yntemlerine daha ok ađırlık verilmesi gerektiđini gstermektedir.

### **5.5 Bilgisayar Deneyimi ve Programlama Deneyiminin Programlama Bařarisına Bir Etkisi Var Mıdır? Problemine İliřkin Sonular**

Bir iř iin harcanan emek ne kadar ok ise o iř ve benzerlerinde bařarıya ulařmak o derece kolaylařacaktır. Deneyim kazanıldıđı konu ile ilgili daha geniř bir bakıř aısı ve daha detaylı bir dřnce yapısı oluřacaktır. Bu alıřmada katılımcıların bilgisayar ve programlama deneyimleri, ilgili olunan yıl temel alınarak alıřmaya dâhil edilmiřtir. Katılımcıları bilgisayar ve programlama deneyimlerinin programlama bařarılarına olumlu bir etki gsterip gstermediđini belirlemek iin korelasyon analizi yapılmıřtır. Yapılan analiz sonucunda bilgisayar deneyiminin anlamli bir katkısı bulunmazken programlama deneyiminin dřk dzeyde bir katkı sađladıđı sonucuna ulařılmıřtır. đrencilerin programlama ile ilgilendikleri yıl sayısı arttıđı programlama bařarısı da artmaktadır. Fakat uzun yıllar boyunca bilgisayar kullanıyor olmanın programlama bařarisına bir katkısı olmamaktadır. Literatr incelendiđinde benzer sonulara ulařmak mmkn. Holden ve Weeden (2003) yaptıkları alıřmada nceden bilgisayar deneyimi olan đrencilerin ilk programlama dersinde deneyimi olmayan đrencilere gre daha bařarılı olduđu sonucuna ulařmıřtır. Mazman (2013) deneyime sahip olan đrencilerin olmayan đrencilere gre programlamadan daha bařarılı olduđu sonucuna ulařmıřtır fakat yapılan regresyon analizinde n deneyimin programlama bařarisına anlamli bir katkı sađlamadıđını vurgulamıřtır. Bunlara karřın Wilson ve Shrock (2001) niversite seviyesindeki programlamaya giriř dersindeki bařarıyı etkileyen faktrleri inceledikleri alıřmada alınan programlama dersi ve programlama ile ilgili olunan yıl temel alınarak hesaplanan programlama deneyiminin programlama bařarısı ile anlamli bir iliřkisi bulamamıřtır.

### **5.6 Programlama Başarısı Üzerinde Etkisi Olduğu Düşünülen, Programlama Yetenek Puanı, Mantıksal Düşünme Yeteneği, Öğrenme Stilleri ve Programlamaya İlişkin Öz Yeterlilik Düzeyi Programlama Başarısını Hangi Öncelikte ve Derecede Açıklamaktadır?**

Kimin programlamada daha başarılı olacağını belirlemek uzun sürelerdir üzerinde çalışılan fakat tam bir sonuca ulaşamayan bir konudur. Yapılan araştırmalar sonucunda birçok değişkenin programlama başarısını etkilediği sonucuna ulaşılmıştır. Bu çalışmada literatürde bulunan çalışmalara da katkı sağlamak amacıyla, programlama yeteneği, mantıksal düşünme yeteneği, öğrenme stilleri ve programlama öz yeterlilik algısının programlama başarısını hangi öncelik ve hangi oranda açıkladığı incelenmiştir.

Elde edilen bulgulara göre bu değişkenler programlama başarısındaki değişimin %41'ini açıklayabilmektedir. Programlama başarısına en büyük katkıyı Ayırıştırma öğrenme stiline sahip olmak, sonrasında güçlü bir öz yeterliliğe sahip olmak ve programlama yeteneğine sahip olmak yapmaktadır.

Bu sonuçlar doğrultusunda programlama eğitimine uygun öğrenme stiline sahip ve programlama da başarılı olabileceğine inanmış öğrencilerin diğer öğrencilere göre daha başarılı olabileceği düşünülebilir.

### **5.7 Öneriler**

Literatürden ve elde edilen bulgulardan yola çıkarak asıl önemsenilmesi gereken konunun öğrencilerin nitelikleri değil öğrencilerin nasıl öğrendiği ve öğrencilere nasıl öğretildiğidir. Çalışmada elde edilen sonuçlarda öğrencilerin yetenek puanı ve mantıksal düşünme yetenekleri ile programlama başarısı arasında bir ilişki bulunamamıştır. Buna karşın öğrenme stilleri ile programlama başarısı arasında bir ilişki bulunmaktadır. Ayırıştırma öğrenme stillerine sahip olan öğrenciler programlama dersinde daha başarılı olmaktadır.

Literatürdeki çalışmalar ve bu çalışma sonuçları incelendiğinde programlama dersi için şu önerilerde bulunulabilir.

Üniversite seviyesindeki programlama eğitimi, öğrencilerin yeni bir ortama ve yaşam şekline alışmaya çalıştığı üniversitenin ilk yılında okutulmamalıdır (Jenkins, 2002).

İlk defa programlama öğrenecek öğrenciler için seçilecek olan programlama dili, basit, kolay okunabilir, söz dizimi basit, ulaşılabilir kaynağı çok olan ve maliyet açısından uygun olan bir dil seçilmelidir. Ortaöğretim seviyesinde bu kriterlere uygun olarak Python programlama dili tavsiye edilmektedir (Güvenir ve Yazıcı, 2015). Ülkemizde orta öğretim seviyesinde programlama dersi sadece meslek liselerinde okutulmaktadır. Bu nedenle ortaöğretim için önerilen Python üniversite seviyesinde okutulan ilk programlama dili olarak tercih edilmesi doğru olacaktır.

Öğrencilerin anlamakta zorluk çektikleri soyut programlama kavramları ile uzun süre uğraşmak öz yeterliliklerini ve motivasyonlarını düşürmektedir. Bu nedenle soyut programlama kavramları somutlaştırarak ve eğlenceli bir şekilde anlatılmalıdır (Başer, 2013; Ersoy ve diğerleri, 2011; Özyurt ve Özyurt, 2016). Bunun için blok ve oyun temelli uygulamalar kullanılabilir.

Yapılan araştırmalarda öğrencilerin en çok şikâyet ettikleri konular, söz dizimi hataları, değişken tiplerini belirleme, anlaşılma hata mesajları, fonksiyon ve parametrelerin hatırlanması şeklinde belirtilmiştir (Bosse ve Gerosa, 2017; Özmen ve Altun, 2014). Bu sorunlara çözüm olması açısından katı veri tipi kuralları olmayan bir programlama dili ve bu dil için geliştirilmiş bir Tümüleşik Geliştirme Ortamı (TGO) kullanılmalıdır. İyi programlanmış bir TGO söz dizimi hatalarını anında göstererek çalışma anında karşılaşılabilecek söz dizimi hatalarının önüne geçmekte, otomatik tamamlama özelliği ile ezberlemesi zor olan fonksiyon isimlerini tamamlamakta ve fonksiyonlarda kullanılan parametreler hakkında bilgi sunmaktadır. Şekil 9’ da PHP programlama dili için geliştirilmiş bir TGO ‘nın söz dizimi hatası uyarısı (kırmızı çizgi ile belirtilmiş) ve fonksiyon parametrelerinin belirtilmesi gösterilmiştir. Ayrıca TGO yazılan kodları renklendirerek kod okunabilirliğini arttırmaktadır.

```
1 <?php
2
3 /**
4  * @param $taban float Üçgenin taban uzunluğu
5  * @param $yukseklk float Üçgenin yüksekliği
6  * @return float|int
7  */
8 function ucgenAlan($taban,$yukseklk){
9     return $taban*$yukseklk/2;
10 }
11
12 ucgenAlan()
```

taban : float, yukseklik : float

Şekil 9 PHPStorm tümleşik geliştirme ortamı

Özyurt ve Özyurt (2016) çalışmasında öğrencilerin programlama dersinden başarısız olmalarının bir nedeni olarak yeteri kadar pratik yapılmamasını göstermiştir. Bu duruma yönelik olarak derste işlenen konuların bol örnek yaptırılarak işlenmesi ve geçmiş konuları da içerecek ufak proje ödevleriyle öğrencileri pratik yapmaya yönlendirmek doğru olacaktır.

Her öğrencinin öğrenme hızı farklıdır. Biri hemen öğrenirken diğeri daha yavaş öğrenebilir. Programlama dersinde anlatılan konuyu ders sırasında anlayamayan bir öğrenci eğer konuyu tekrar edebileceği bir kaynak yok ise dersten geri kalacak ve giderek dersten uzaklaşır. Bu durumun önüne geçmek için derste anlatılan konuları öğrencilerin bireysel olarak ulaşabileceği bir platformda paylaşarak bireysel eğitime imkân sağlanmalıdır. Derste anlatılan konu ekran kaydı yapılarak bir video hazırlanabilir ve bu video İnternet ortamında, dersi alan öğrencilere ulaştırılarak konuyu tekrar etmesi sağlanabilir (Durak, 2013; Jenkins, 2002; Özyurt ve Özyurt, 2015).

Çalışmanın en önemli bulgusu öğretmenlerin ve öğretmen adaylarının daha çok tercih ettiği Özümseme öğrenme stilinin programlama dersinde en başarısız öğrencilerin sahip olduğu stil olmasıdır. Bilgisayar ve öğretim teknolojileri Eğitimi bölümü, geleceğin bilgisayar öğretmenlerini yetiştiren bir bölümdür. Öğretmenlerin öğrenme stilleri üzerinde yapılan diğer araştırmalar da bu bölümde Özümseme ve Ayırıştırma stillerine sahip olan öğrencilerin çoğunlukta olması sonucuyla paralellik göstermektedir(Can, 2011).

Çalışmada elde edilen sonuçlara göre en düşük başarı Özümseyen öğrenme stiline sahip olan öğrencilerdedir. Özümseyen öğrenme stiline sahip olan öğrencilerin başarılarını arttırmak için ders işlenişinde özümseyen stilinin tercih ettiği öğrenme yollarına uygun etkinliklerin sayısı artırılmalıdır. Özümseme öğrenme stili yansıtıcı gözlem ve soyut kavramsallaştırma öğrenme yollarının kesişimindeki öğrenme stilidir. Yansıtıcı gözlem öğrenme yolu için tavsiye edilen öğretim yöntemleri, tartışma, beyin fırtınası, problem çözme ve soru cevap yöntemidir (Gencel, 2006, s. 44). Soyut kavramsallaştırma öğrenme yolu için ise öğrencilere bireysel çalışma imkanı ve okuyarak öğrenmesi için ortam yaratılması önerilmektedir. Ayrıca laboratuvar çalışmaları ve projeler soyut kavramsallaştırma öğrenme yolunu tercih edenler için öğrenmeyi kolaylaştırıcı etkinliklerdir (Gencel, 2006, s. 44).

Özet olarak, programlama dersinin başarısını arttırmak için,

1. Üniversite ikinci sınıfta katı kuralları olmayan ve kolay öğrenilen bir programlama dili ve o dil için geliştirilmiş TGO kullanarak dersi işlemek
2. Öğrencilere ders dışında konuları tekrar edebilecekleri kaynaklar ve küçük proje ödevleri vermek
3. Öğrencilerin öğrenme stillerine dikkat ederek ders sürecini planlamak

gerekmektedir.

## KAYNAKÇA

- Açıkgöz, K. Ü. (2003). *Etkili öğrenme ve öğretme* (4. bs.). İzmir: Eğitim Dünyası Yayınları.
- Akdağ, M. (2015). Bilimsel araştırma yöntemleri. 1 Mayıs 2017 tarihinde <https://cms.inonu.edu.tr/uploads/contentfile/953/files/modeller.pptx> adresinden erişildi.
- Akpınar, Y. ve Altun, A. (2014). Bilgi toplumu okullarında programlama eğitimi gereksinimi. *İlköğretim Online*, 13(1), 1–4.
- Alsbaugh, C. A. (1972). Identification of some components of computer programming aptitude. *Journal for Research in Mathematics Education*, 3(2), 89. doi:10.2307/748667
- Altun, A. (2003). E-Okuryazarlık. *Milli Eğitim Dergisi*, 158(158), 1–13.
- Altun, A. ve Mazman, S. G. (2012). Programlamaya ilişkin öz yeterlilik algısı ölçeğinin Türkçe formunun geçerlilik ve güvenilirlik çalışması. *Eğitimde ve Psikolojide Ölçme ve Değerlendirme Dergisi*, 3(2), 297–308.
- Altun, H. (2016). *İlköğretim matematik öğretmenliği lisans öğrencilerinin türev konusundaki akademik başarıları ile öğrenme stilleri arasındaki ilişkilerin incelenmesi*. (Yayımlanmamış doktora). Dokuz Eylül Üniversitesi, İlköğretim Anabilim Dalı, Matematik Öğretmenliği Programı, İzmir.
- Arabacıoğlu, T., Bülbül, H. İ. ve Filiz, A. (2007). Bilgisayar programlama öğretiminde yeni bir yaklaşım. IX. Akademik Bilişim Konferansı Bildirileri içinde (ss. 193–197). Akademik Bilişim, sunulmuş bildiri, Kütahya.
- Arı, E. (2011). Temel kavramlar. S. B. Filiz (Ed.), *Öğrenme öğretme kuram ve yaklaşımları* içinde (1. bs., s. 411). Ankara: Pegem Akademi.
- Aşkar, P. ve Akkoyunlu, B. (1993). Kolb öğrenme stili envanteri. *Eğitim ve Bilim*, 17(87), 37–47.
- Aşkar, P. ve Davenport, D. (2009). An investigation of factors related to self-efficacy for java programming among engineering students. *The Turkish Online Journal of Educational Technology*, 8(1), 26–32.
- Aşkar, P. ve Dönmez, O. (2004). Eğitim yazılımı geliştirme öz-yeterlilik algısı ölçeği. *Eğitim Bilimleri ve Uygulama*, 3(6), 259–274.
- Aşkar, P. ve Umay, A. (2001). İlköğretim matematik öğretmenliği öğrencilerinin bilgisayarla ilgili öz-yeterlilik algısı. *Hacettepe Üniversitesi Eğitim Fakültesi Dergisi*, 21(21), 1–8.

- Austin, H. S. (1987). Predictors of Pascal programming achievement for community college students. *ACM SIGCSE Bulletin*, 19(1), 161–164. doi:10.1145/31726.31752
- Aydın, F. ve Kara, F. N. (2013). Öğretmen adaylarının teknolojiye yönelik tutumları: Ölçek geliştirme çalışması. *Türk Fen Eğitimi Dergisi*, 10(4), 103–118.
- Aziz, A. (2010). *Sosyal bilimlerde araştırma yöntemleri ve teknikleri* (5. bs.). Ankara: Nobel Yayınları.
- Bandura, A. (1994). Self-Efficacy. V. S. Ramachandran (Ed.), *Encyclopedia of human behavior* içinde (C. 4, ss. 71–81). New York: Academic Press.
- Bandura, A. (2012). On the Functional properties of perceived self-efficacy revisited. *Journal of Management*, 38(1), 9–44. doi:10.1177/0149206311410606
- Başer, M. (2013). Bilgisayar programlamaya karşı tutum ölçeği geliştirme çalışması. *The Journal of Academic Social Science Studies*, 6(Volume 6 Issue 6), 199–215. doi:10.9761/JASSS1702
- Bayman, P. ve Mayer, R. E. (1988). Using conceptual models to teach BASIC computer programming. *Journal of Educational Psychology*, 80(3), 291–298. doi:10.1037/0022-0663.80.3.291
- Bennedsen, J. ve Caspersen, M. E. (2007). Failure rates in introductory programming. *ACM SIGCSE Bulletin*, 39(2), 32. doi:10.1145/1272848.1272879
- Bilgin, İ. ve Ateş, S. (2004). İlköğretim bölümü öğrencilerinin mantıksal düşünme yeteneklerinin alan ve cinsiyet açısından karşılaştırması. *Abant İzzet Baysal Üniversitesi Sosyal Bilimler Dergisi*, 2004–1(8), 17–27.
- Blackwell, A. (2002). What is programming. PPIG 2002 - 14th Annual Workshop içinde (ss. 204–218). 14th Workshop of the Psychology of Programming Interest Group, sunulmuş bildiri, Brunel University.
- Bornat, R., Dehnadi, S. ve Simon. (2008). Mental Models, Consistency and Programming Aptitude. Proceedings of the Tenth Conference on Australasian Computing Education - Volume 78 içinde , ACE '08 (ss. 53 – 61). Tenth Australasian Computing Education Conference, sunulmuş bildiri, Darlinghurst, Australia: Australian Computer Society, Inc.
- Bosse, Y. ve Gerosa, M. A. (2017). Why is programming so difficult to learn? : Patterns of Difficulties Related to Programming Learning Mid-Stage. *ACM SIGSOFT Software Engineering Notes*, 41(6), 1–6. doi:10.1145/3011286.3011301
- Bozdoğan, A. (2007). *Fen bilgisi öğretiminde çalışma yaprakları ile öğretimin öğrencilerin fen bilgisi tutumuna ve mantıksal düşünme becerilerine etkisi.*



(Yayımlanmamış yüksek lisans). Çukurova Üniversitesi, Sosyal Bilimler Enstitüsü, İlköğretim Anabilim Dalı.

Brusilovsky, P., Calabrese, E., Hvorecky, J., Kouchnirenko, A. ve Miller, P. (1997). Mini-languages: a way to learn programming principles. *Education and Information Technologies*, 2(1), 65–83. doi:10.1023/A:1018636507883

Büyüköztürk, Ş. (2011). *Sosyal bilimler için veri analizi el kitabı: İstatistik, araştırma deseni, SPSS uygulamaları ve yorum* (15. bs., C. 1-15). Ankara: Pegem Yayınları.

Büyüköztürk, Ş., Çokluk, Ö. ve Köklü, N. (2015). *Sosyal bilimler için istatistik* (17. bs.). Ankara: Pegem Akademi.

Byrne, P. ve Lyons, G. (2001). The effect of student attributes on success in programming (ss. 49–52). ACM Press. doi:10.1145/377435.377467

Can, A. (2017). *SPSS ile bilimsel araştırma sürecinde nicel veri analizi* (5. bs.). Ankara: Pegem Akademi.

Can, Ş. (2011). Sınıf öğretmeni adaylarının öğrenme stilleri ile bazı değişkenler arasındaki ilişkinin araştırılması. *Hacettepe Üniversitesi Eğitim Fakültesi Dergisi*, (41), 70–82.

Casey, P. J. (1997). Computer programming: A medium for teaching problem solving. *Computers in the Schools*, 13(1–2), 41–51. doi:10.1300/J025v13n01\_05

Clements, D. H. ve Gullo, D. F. (1984). Effects of computer programming on young children's cognition. *Journal of Educational Psychology*, 76(6), 1051–1058. doi:10.1037/0022-0663.76.6.1051

Computer Programming Aptitude Test. (t.y.). 3 Ekim 2016 tarihinde <https://www.kent.ac.uk/careers/tests/computer-test.htm> adresinden erişildi.

Davidson, K., Larzon, L. ve Ljunggren, K. (2010). Self-efficacy in programming among STS students. *Retrieved August, 12*. <http://www.it.uu.se/edu/course/homepage/datadidaktik/ht10/reports/Self-Efficacy.pdf> adresinden erişildi.

Dehnadi, S. (2006). Testing programming aptitude (ss. 22–37). Proceedings of the 18th Annual Workshop of the Psychology of Programming Interest Group, sunulmuş bildiri.

Dehnadi, S. ve Bornat, R. (2006). The camel has two humps. *Middlesex University, UK*, 1–21.

Demir, F. (2015, Aralık). *Programlama öğretiminde eğitsel programlama dilinin farklı kullanımlarının programlama başarısı ve kaygısına etkisi*.

(Yayımlanmamış doktora). Atatürk Üniversitesi, Eğitim Bilimleri Enstitüsü, Bilgisayar ve Öğretim Teknolojileri Eğitimi Ana Bilim Dalı.

Demirer, V. ve Sak, N. (2016). Dünyada ve Türkiye’de programlama eğitimi ve yeni yaklaşımlar. *Eğitimde Kuram ve Uygulama*, 12(3), 521–546.

Deniz, S., Görgen, İ. ve Şeker, H. (2006). Tezsiz yüksek lisans öğretmen adaylarının teknolojiye yönelik tutumları. *Eurasian Journal of Educational Research (EJER)*, (23).

Devlet Planlama Teşkilatı. (2001). *Bilişim teknolojileri ve politikaları Özel İhtisas Komisyonu Raporu* ( No: DPT. 2560-ÖİK. 576). Sekizinci Beş Yıllık Kalkınma Planı. Ankara: Devlet Planlama Teşkilatı.

Driscoll, M. (2000). Introduction to theories of learning and instruction. *Psychology of learning for instruction*, 3–28.

Dunn, R., Beudury, J. S., Klavas, A. ve Babadoğan, C. (1991). Öğrenme stilleriyle ilgili araştırmaların taraması. *Ankara Üniversitesi Eğitim Bilimleri Fakültesi Dergisi*, 2(24), 603–619.

Durak, G. (2013, Aralık). *Programlama dillerinin çevrimiçi öğretimi: Öğrenenlerin tutumlarının, memnuyetlerinin ve akademik başarılarının incelenmesi*. (Yayımlanmamış doktora). Anadolu Üniversitesi, Sosyal Bilimler Enstitüsü, Uzaktan Eğitim Anabilim Dalı.

Erdoğan, B. (2005). *Programlama başarısı ile akademik başarı, genel yetenek, bilgisayara karşı tutum, cinsiyet ve lise türü arasındaki ilişkilerin incelenmesi*. (Yayımlanmamış yüksek lisans). Marmara Üniversitesi, Eğitim Bilimleri Enstitüsü, Bilgisayar ve Öğretim Teknolojileri Eğitimi Ana Bilim Dalı.

Ersoy, H., Madran, R. O. ve Gülbahar, Y. (2011). Programlama dilleri öğretimine bir model önerisi: robot programlama. XIII. Akademik Bilişim Konferansı Bildirileri içinde (ss. 731–736). Akademik Bilişim’11, sunulmuş bildiri, Malatya: İnönü Üniversitesi.

Eymen, U. E. (2007). *SPSS 15.0 ile veri analiz yöntemleri* (1. bs., C. 1-1, C. 1). İstatistik Merkezi.  
[http://yunus.hacettepe.edu.tr/~tonta/courses/spring2009/bby606/SPSS\\_15.0\\_ile\\_Veri\\_Analizi.pdf](http://yunus.hacettepe.edu.tr/~tonta/courses/spring2009/bby606/SPSS_15.0_ile_Veri_Analizi.pdf) adresinden erişildi.

Fowler, L., Campbell, V., McGill, D. ve Roy, G. (2003). An Innovative Approach to Teaching First Year Programming Supported by Learning Style Investigation. Engineering Education for a Sustainable Future: Proceedings of the 14th Annual Conference for Australasian Association for Engineering Education and 9th Australasian Women in Engineering Forum içinde (ss. 337–348). 14th Annual AAEE Conference, sunulmuş bildiri, Melbourne, Australia: Australasian Association for Engineering Education.

- Geban, Ö., Aşkar, P. ve Özkan, İ. (1992). Effects of computer simulations and problem-solving approaches on high school students. *The Journal of Educational Research*, 86(1), 5–10. doi:10.1080/00220671.1992.9941821
- Gencel, İ. E. (2006). *Öğrenme stilleri, deneysel öğrenme kuramına dayalı eğitim, tutum ve sosyal bilgiler program hedeflerine erişimi düzeyi*. (Yayımlanmamış doktora). Dokuz Eylül Üniversitesi, Dokuz Eylül Üniversitesi , Eğitim Bilimleri Enstitüsü , Eğitim Bilimleri Anabilim Dalı , Eğitim Programları ve Öğretim Bilim Dalı.
- Gencel, İ. E. (2007). Kolb'un deneysel öğrenme kuramına dayalı öğrenme stilleri envanteri III 'ü Türkçe 'ye uyarlama çalışması. *Dokuz Eylül Üniversitesi Sosyal Bilimler Enstitüsü Dergisi*, 9(2), 120–139.
- Gorman, H. ve Bourne, L. E. (1983). Learning to think by learning LOGO: Rule learning in third-grade computer programmers. *Bulletin of the Psychonomic Society*, 21(3), 165–167. doi:10.3758/BF03334676
- Grant, N. S. (2003). A study on critical thinking, cognitive learning style, and gender in various information science programming classes (s. 96). ACM Press. doi:10.1145/947121.947142
- Güler, Z. (2010, Eylül). *İlköğretim öğrencilerinin SBS puanları ile ders başarıları, bilimsel süreç becerileri ve mantıksal düşünme yetenekleri arasındaki ilişki*. (Yayımlanmamış yüksek lisans). Abant İzzet Baysal Üniversitesi, Abant İzzet Baysal Üniversitesi Sosyal Bilimler Enstitüsü İlköğretim Fen Bilgisi Öğretmenliği Bilim Dalı.
- Gündüz, Ş. ve Odabaşı, F. (2004). Bilgi çağında öğretmen adaylarının eğitiminde öğretim teknolojileri ve materyal geliştirme dersinin önemi. *The Turkish Online Journal of Educational Technology*, 3(1), 7.
- Güvenir, H. altay ve Yazıcı, C. (2015). Üniversite öncesi kodlama eğitiminde programlama aracı ve dili seçimi (ss. 94–103). Eğitimde Fatih Projesi Eğitim Teknolojileri Zirvesi, sunulmuş bildiri, Ankara.
- Harris, J. (2014). Testing programming aptitude in introductory programming courses. *J. Comput. Sci. Coll.*,30(2), 149–156.
- Holden, E. ve Weeden, E. (2003). The impact of prior experience in an information technology programming course sequence (ss. 41–46). CITC4 '03 Proceedings of the 4th conference on Information technology curriculum, sunulmuş bildiri, ACM Press. doi:10.1145/947121.947131
- Illeris, K. (Ed.). (2009). *Contemporary theories of learning: learning theorists -- in their own words* (1. bs.). London; New York: Routledge. <http://public.ebib.com/choice/publicfullrecord.aspx?p=425525> adresinden erişildi.

- Informatics Defined. (t.y.). 7 Kasım 2016 tarihinde <https://soic.iupui.edu/about/what-is-informatics/> adresinden erişildi.
- Jegede, P. O. (2009). Predictors of Java programming self efficacy among engineering students in a Nigerian university. *International Journal of Computer Science and Information Security*, 4(1 & 2), 196–202.
- Jenkins, T. (2002). On the difficulty of learning to program. Proceedings of the 3rd Annual Conference of the LTSN Centre for Information and Computer Sciences içinde (C. 4, ss. 53–58). Citeseer.
- Jiau, H. C., Chen, J. C. ve Ssu, K.-F. (2009). Enhancing self-motivation in learning programming using game-based simulation and metrics. *IEEE Transactions on Education*, 52(4), 555–562. doi:10.1109/TE.2008.2010983
- Karabak, D. ve Güneş, A. (2013). Ortaokul birinci sınıf öğrencileri için yazılım geliştirme alanında müfredat önerisi. *Eğitim ve Öğretim Araştırmaları Dergisi*, 2(3), 175–181.
- Karasar, N. (2017). *Bilimsel araştırma yöntemi: kavramlar, ilkeler, teknikler* (30. bs.). Ankara: Nobel.
- Kellner, D. (2000). New technologies/new literacies: Reconstructing education for the new millennium. *International Journal of Technology and Design Education*, 11(1), 67–81.
- Keskinsoy, A. (2010). *Mesleki liselerde görsel programlama başarısını etkileyen faktörler*. (Yayımlanmamış yüksek lisans). Marmara Üniversitesi, Eğitim Bilimleri Enstitüsü, Bilgisayar ve Öğretim Teknolojileri Eğitimi Ana Bilim Dalı.
- Kılıç, D. ve Sağlam, N. (2009). Öğrencilerin mantıksal düşünme yeteneklerinin bazı değişkenler açısından incelenmesi. *Ege Eğitim Dergisi*, 10(2). doi:10.12984/eed.73239
- Kılıç, E. (2002, Ocak). *Web temelli öğrenmede baskın öğrenme stiline öğrenme etkinlikleri tercihi ve akademik başarıya etkisi*. (Yayımlanmamış yüksek lisans). Ankara Üniversitesi, Ankara Üniversitesi Eğitim Bilimleri Enstitüsü Eğitim Programları ve Öğretim Ana Bilim Dalı.
- Kılınç, A. ve Salman, S. (2006). Fen ve matematik alanları öğretmen adaylarında bilgisayar okuryazarlığı. *Mersin Üniversitesi Eğitim Fakültesi Dergisi*, 2(2), 150–166.
- Kolb, A. Y. ve Kolb, D. A. (2005). Learning styles and learning spaces: Enhancing experiential learning in higher education. *Academy of management learning & education*, 4(2), 193–212.

- Kolb, D. A. (1984). *Experiential learning: experience as the source of learning and development*. Englewood Cliffs, NJ: Prentice Hall. <https://academic.regis.edu/ed205/kolb.pdf> adresinden erişildi.
- Kurbanoglu, S. ve Akkoyunlu, B. (2001). Öğrencilere bilgi okuryazarlığı becerilerinin kazandırılması üzerine bir çalışma. *Hacettepe Üniversitesi Eğitim Fakültesi Dergisi*, (21), 81–88.
- Lacher, L. L. ve Lewis, M. C. (2015). The effectiveness of video quizzes in a flipped class (ss. 224–228). ACM Press. doi:10.1145/2676723.2677302
- Lau, W. W. F. ve Yuen, A. H. K. (2011). Modelling programming performance: Beyond the influence of learner characteristics. *Computers & Education*, 57(1), 1202–1213. doi:10.1016/j.compedu.2011.01.002
- Lawson, A. E. (1978). The development and validation of a classroom test of formal reasoning. *Journal of Research in Science Teaching*, 15(1), 11–24. doi:10.1002/tea.3660150103
- Lukow, J. E. (2002). *Learning styles as predictors of student attitudes toward the use of technology in recreation courses*. (Doktora Tezi). <https://www.learntechlib.org/p/117999/> adresinden erişildi.
- Mannila, L., Peltomäki, M. ve Salakoski, T. (2006). What about a simple language? Analyzing the difficulties in learning to program. *Computer Science Education*, 16(3), 211–227. doi:10.1080/08993400600912384
- Mazman, S. G. (2013). *Programlama performansını etkileyen faktörlerin bilişsel tabanlı bireysel farklılıklar temelinde modellenmesi*. (Yayımlanmamış doktora). Hacettepe Üniversitesi, Eğitim Bilimleri Enstitüsü, Bilgisayar ve Öğretim Teknolojileri Eğitimi Anabilim Dalı, Ankara.
- Meštrović, A. ve Martinčić-Ipšić, S. (2012). *Initial study on students' success in a first programming course*. 12th Workshop SE Education and Reverse Engineering. <http://www2.informatik.hu-berlin.de/swt/intkoop/daad/opat2012/talks/5.Day%20-%202.Pres%20-%20Mestrovic%20Martincic.pdf> adresinden erişildi.
- Milli Eğitim Bakanlığı. (2013). *İlköğretim Teknoloji ve Tasarım Dersi (6, 7 ve 8. Sınıflar) Öğretim Programı*. Devlet Kitapları.
- Özden, Y. ve Şimşek, H. (1998). Davranışçılıktan oluşturmaçılığa: “öğrenme” paradigmasının dönüşümü ve Türk eğitimi. *Bilgi ve Toplum Dergisi*, (1), 71–82.
- Özdiñç, F. ve Altun, A. (2014). Bilişim teknolojileri öğretmeni adaylarının programlama sürecini etkileyen faktörler. *İlköğretim Online*, 13(4). doi:10.17051/io.2014.54872

- Özmen, B. ve Altun, A. (2014). Undergraduate students' experiences in programming: difficulties and obstacles. *Turkish Online Journal of Qualitative Inquiry*, 5(3). doi:10.17569/tojqi.20328
- Özyurt, Ö. ve Özyurt, H. (2015). Bilgisayar programcılığı öğrencilerinin programlamaya karşı tutum ve programlama öz-yeterliklerinin belirlenmesine yönelik bir çalışms / A study for determining computer programming students' attitudes towards programming and their programming self-efficacy /. *Eğitimde Kuram ve Uygulama*, 11(1), 51–67. doi:10.17244/eku.53204
- Özyurt, Ö. ve Özyurt, H. (2016). Programlama öğrenmek gerçekten zor mu? Yazılım mühendisliği birinci sınıf öğrencileri ile bir durum çalışması. *10 th International Computer & Instructional Technologies Symposium* içinde (ss. 763–768). 10 th International Computer & Instructional Technologies Symposium, sunulmuş bildiri, Rize.
- Pajares, F. (1996). Self-efficacy beliefs in academic settings. *Review of Educational Research*, 66(4), 543–578. doi:10.3102/00346543066004543
- Rajala, T., Laakso, M.-J., Kaila, E. ve Salakoski, T. (2008). Effectiveness of program visualization: A case study with the ViLLE tool. *Journal of Information Technology Education*, 7, 15–32.
- Ramalingam, V., Labelle, D. ve Wiedenbeck, S. (2004). Self-efficacy and mental models in learning to program. *Proceedings of the 9th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education* içinde, ITiCSE '04 (ss. 171–175). New York, NY, USA: ACM. doi:10.1145/1007996.1008042
- Ramalingam, V. ve Wiedenbeck, S. (1998). Development and validation of scores on a computer programming self-efficacy scale and group analyses of novice programmer self-efficacy. *Journal of Educational Computing Research*, 19(4), 367–381. doi:10.2190/C670-Y3C8-LTJ1-CT3P
- Resnick, M. (2013, 8 Mayıs). Learn to code, code to learn. *Learn to Code, Code to Learn*. 27 Mart 2017 tarihinde <https://www.edsurge.com/news/2013-05-08-learn-to-code-code-to-learn> adresinden erişildi.
- Resnick, M. ve Ocko, S. (1990). *LEGO/logo–learning through and about design*. Epistemology and Learning Group, MIT Media Laboratory Cambridge, MA. <http://icsd.ccp14.ac.uk/ccp/ccp14/ftp-mirror/alife/zooland/pub/research/ci/Alife/mitchel-resnik/memo8.PS.gz> adresinden erişildi.
- Saeli, M., Perrenet, J., Jochems, W. M. ve Zwaneveld, B. (2011). Teaching programming in secondary school: a pedagogical content knowledge perspective. *Informatics in Education-An International Journal*, (Vol 10\_1), 73–88.

- Sebetci, Ö. ve Aksu, G. (2014). Öğrencilerin mantıksal ve analitik düşünme becerilerinin programlama dilleri başarısına etkisi. *Eğitim Bilimleri ve Uygulama*, 13(25), 65–83.
- Senemoğlu, N. (2012). *Gelişim öğrenme ve öğretim: Kuramdan uygulamaya* (22. bs.). Ankara: Pegem Akademi.
- Solmaz, E. (2014, Aralık). *Programlama dili öğretiminde Alice yazılımını ders başarısı, eleştirel düşünme ve problemler çözme becerileri ile üstbilişsel farkındalık düzeyine etkisi*. (Yayımlanmamış doktora). Gazi Üniversitesi, Eğitim Bilimleri Enstitüsü.
- Şahin, İ. (2009). Eğitsel internet kullanım özyeterliliği inançları ölçeğinin geçerliği ve güvenilirliği. *Selçuk Üniversitesi Sosyal Bilimler Enstitüsü Dergisi*, 0(21), 461–471.
- Thomas, L., Ratcliffe, M., Woodbury, J. ve Jarman, E. (2002). Learning styles and performance in the introductory programming sequence. *Proceedings of the 33rd SIGCSE Technical Symposium on Computer Science Education* içinde, SIGCSE '02 (ss. 33–37). New York, NY, USA: ACM. doi:10.1145/563340.563352
- Tobin, K. G. ve Capie, W. (1981). The development and validation of a group test of logical thinking. *Educational and Psychological Measurement*, 41(2), 413–423.
- Tortop, H. S. (2016). ÜYÜKEP bilgisayar programlama dersinin üstün yetenekli öğrencilerin mantıksal düşünme becerilerine etkisinin incelenmesi. *Üstün Zekâlılar Eğitimi ve Yaratıcılık Dergisi*, 3(1), 10–16.
- Tukiainen, M. ve Mönkkönen, E. (2002). Programming aptitude testing as a prediction of learning to program (ss. 45–57). 14th Workshop of the Psychology of Programming Interest Group, sunulmuş bildiri, Brunel University.
- TÜSİAD. (2016). *Türkiye 'nin küresel rekabetçiliği için bir gereklilik olarak Sanayi 4.0* (No: TÜSİAD-T/2016-03/576).
- Wilson, B. C. ve Shrock, S. (2001). Contributing to success in an introductory computer science course: a study of twelve factors. *ACM SIGCSE Bulletin*, 33(1), 184–188. doi:10.1145/366413.364581
- Winrow, B. (1999, Eylül). The Walden programmer analyst aptitude test. *Dr. Dobb's*. 14 Nisan 2017 tarihinde <http://www.drdoobs.com/the-walden-programmer-analyst-aptitude-t/184411169> adresinden erişildi.
- Wolfe, J. M. (1971). Perspectives on testing for programming aptitude (ss. 268–277). ACM Press. doi:10.1145/800184.810494

- Yıldırım, U. ve Öner, Ş. (2004). Bilgi toplumu sürecinde yerel yönetimlerde eğitim-bilişim teknolojilerinden yararlanma: Türkiye’de E-belediye uygulamaları. *The Turkish Online Journal of Educational Technology–TOJET*, January, 3(1), 49–60.
- Yılmaz, F. (2013). *Meslek yüksek okulu öğrencilerinin programlama başarısını etkileyen faktörlerin incelenmesi*. (Yayımlanmamış yüksek lisans). Gazi Üniversitesi, Eğitim Bilimleri Enstitüsü, Bilgisayar Ve Öğretim Teknolojileri Eğitimi Anabilim Dalı, Ankara.
- Yiğit, M. F. (2016). *Görsel programlama ortamı ile öğretimin öğrencilerin bilgisayar programlamayı öğrenmesine ve programlamaya karşı tutumlarına etkisinin incelenmesi*. (Yayımlanmamış yüksek lisans). Ondokuz Mayıs Üniversitesi, Samsun.
- Yüzüak, A. V. ve Dökme, İ. (2015). Lawson mantıksal düşünme testinin (çoktan seçmeli versiyonu) uyarlanması adaptation of Lawson’s classroom of science reasoning (multiple-choice version). *Gazi Üniversitesi Eğitim Fakültesi Dergisi*, 35(3), 443–456.



## EKLER

### Ek 1 Programlama Yetenek Testi

**Ad – Soyad: Numara:**

Ölçekte yer alan sorulara verdiğiniz yanıtlar, kesinlikle size not vermek ya da sizi eleştirmek amacıyla kullanılmayacaktır. Ölçekte vereceğiniz cevaplarınız tamamıyla gizli tutulacak ve sadece araştırmacılar tarafından değerlendirilecektir; elde edilecek bilgiler bilimsel yayımlarda kullanılacaktır.

Bu sorular sizin programlamaya ne kadar yatkın olduğunuzu belirlemek içindir. Her sorunun bir doğru cevabı vardır.

1)	jkojaspfskasfk	jkojaspfskasfk
2)	iqtgiwegasdio	iqtgiyegasdio
3)	iqtgiwegasdio	iqtgiwegasdio
4)	wyesdsdlxvc	wyesdsdlxvc
5)	jasbhuhoqhwlk	jasbhuhoqhwlk
6)	poanksdnuqwb	poanksdnuqwb

1. Yukarıdaki 6 satırda sol ve sağ taraf aynı karakter kümelerinden oluşmuştur, fakat bir satır farklıdır. Bu farklı satır hangisidir?

A) 1 B) 2 C) 3 D) 4 E) 5 F) 6

1)	f7y349hwe[nfl';SD, M	f7y349hwe[nfl';SD, M
2)	CDBI98QEy39o;@: IP	CDBI98QEy39o;@: IP
3)	\ vnrlrvoprr~f.4'wr	\ vnrlrvoprr~f.4'wr
4)	;sapr){3I4JM\@~mm	;sapr){3I4JM\@~mm
5)	dsNvn(Pos\o;l\md;)	dsNvn(Pos\e;l\md;)
6)	pUJ304'f.:gWV,[c];IMq;I	pUJ304'f.:gWV,[c];IMq;I

2. Yukarıdaki 6 satırda sol ve sağ taraf aynı karakter kümelerinden oluşmuştur, fakat bir satır farklıdır. Bu farklı satır hangisidir?

A) 1 B) 2 C) 3 D) 4 E) 5 F) 6

1)	'z;agle5pao5[,zd,pE;"	'z;agle5pao5[,zd,pE;"
2)	\kjfo5;gm"va'a'c,\fz	\kjfo5;gm"va'a'c,\fz
3)	zk;rgkggr'lz#g#[[pr[s5o;	zk;rgkggr'lz#g#[[pr[s5o;
4)	[aepv5[.3.q3#[kca;fh'	[aepv5[.3.q3#[kca;fh'
5)	oaejo[v,5./Av?cva;VS'	oaejo[v,5./Av?cva;VS'
6)	ladnl:vdlne,842rybqi\m	ladnl:vdlne;842rybqi\m

3. Yukarıdaki 6 satırda sol ve sağ taraf aynı karakter kümelerinden oluşmuştur, fakat bir satır farklıdır. Bu farklı satır hangisidir?

A) 1 B) 2 C) 3 D) 4 E) 5 F) 6

4. Ahmet aklından bir sayı tutuyor. Bu sayının karesini alıyor, 5 çıkartıyor, 4 ile çarpıyor, 7 çıkartıyor 3'e bölüyor ve son olarak 6 ekliyor. Sonuç 9 ise, Ahmet'in aklından tuttuğu sayı kaçtır

A) 1 B) 2 C) 3 D) 4 E) 5 F) 6

5. Bir saatin akrebi öğleden sonra 2'den öğleden önce 9'a geri alınır, akrep kaç derece dönmüş olur?

A) 120 B) 135 C) 150 D) 165 E) 180 F) 205

6. Yukarıdaki şekilde taranmış alan, toplam alanın yaklaşık yüzde kaçdır?

A) 60 B) 63 C) 66 D) 69 E) 72 F) 75

**7, 8 ve 9. soruları aşağıdaki alfabeye göre çözün**

ALFABE: A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

7. Eğer  $ADD=9$ ,  $BAD=7$  ve  $CAD=8$  ise  $ADA$  kaçtır?

A) 3 B) 4 C) 5 D) 6 E) 7 F) 8

8. Eğer  $BAD=10$ ,  $DAC=11$ , ve  $CGI=22$  ise,  $OCCAM$  kaçtır?

A) 35 B) 36 C) 37 D) 39 E) 40 F) Hiçbiri

9. Eğer  $DATA=52$ ,  $CACHE=40$ , ve  $BIT=62$  ise  $BABBAGE$  kaçtır?

A) 40 B) 41 C) 42 D) 43 E) 44 F) Hiçbiri

10. Yönünüz kuzeye doğrudur. Sola 90 derece dönün. 180 derece sağa dönün. Yönünüzü ters çevirin. Sola 45 derece dönün. Yönünüzü ters çevirin. Sağa 270 derece dönün. Şimdi yönünüz nedir?

A) Kuzey B) Batı C) Güney Doğu

D) Güney Batı E) Kuzey Batı F) Hiçbiri

Bir uçak firması biletleri şu şekilde kodlamaktadır:

- İngiltere'ye uçuşlar A, Avrupa'ya uçuşlar B, Asya'ya uçuşlar C, ve Amerika'ya uçuşlar D ile kodlanmaktadır.
- Eğer uçuşlar akşam 10 ile sabah 6 arasında ise, aynı harflerin küçükleri kullanılmaktadır (a,b,c, ve d)
- Bay yolcular X, bayan yolcular Y ile kodlanmaktadır. Çocuklar için küçük harfler kullanılmaktadır (x ve y)
- Yemekler şu şekilde kodlanmaktadır: Avrupa yemekleri G, Asya yemekleri H, Vejeteryen (etyemez) K. Çocuklar için aynı harflerin küçükleri kullanılmaktadır. (g,h,k)
- Birinci sınıf yolcular P, iş adamları Q, Ekonomi sınıfı R ile kodlanmaktadır

11. Ekonomi sınıfında sabah 5 uçağı ile Paris'e giden 8 yaşındaki vejeteryen (etyemez) kız çocuk için kod nedir?

- A) bYkR B) bykr C) bykR D) BykP E) aykR F) Hiçbiri

12. Üç bilgisayar bir sıra halinde dizilmektedir. Dell (D), Exper (E)'in solunda fakat yanında olmak zorunda değildir. Mavi bilgisayar Beyaz bilgisayarın sağındadır. Siyah bilgisayar Casper (C)'in solundadır. Casper, Exper'in solundadır. Bilgisayarların soldan sağa sıralanışı nasıldır?

- A) E, C, D B) E, D, C C) C, D, E D) C, E, D E) D, E, C F) D, C, E

13. Tuncay, verilen bir kutu şekerin üçte birini yeyince, kendisini hasta hissediyor ve duruyor. Sonraki gün kalanın üçte birini yiyor. Bir sonraki gün ise kalanın üçte birini yiyor. Geriye 8 şeker kalmış ise, Tuncay'a kaç şeker verilmiştir?

- A) 18 B) 21 C) 24 D) 27 E) 30 F) 33

**14 ve 15. soruları aşağıdaki bilgilere göre çözün**

Akıllı maymunların kullandığı bir sayı sistemi şu şekildedir:

- Bir muz = 1
- Bir portakal ve iki muz = 6
- Bir portakal yarım mango değerindedir

14. İki mango, bir portakal ve bir muzun sayı değeri kaçtır?

- A) 21 B) 24 C) 27 D) 30 E) 33 F) 36

15. İki mango ve bir portakalın bir portakal ve bir muz bölümünün meyve cinsinden değeri nedir?  
A) Bir mango B) Bir muz C) Bir portakal  
D) İki muz E) Üç muz F) Bir portakal ve bir muz
- 

**16, 17 ve 18. soruları aşağıdaki alfabeyle göre çözün:**

ALFABE: A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

16. Eğer JAVA'nın kodu LCXC ise, BASIC'in kodu nedir?  
A) CBTJD B) DCUKE C) EDVLF D) FEWMG E) CDFFG F) Hiçbiri
- 

17. Eğer FORTRAN'ın kodu GMUPWUU ise PASCAL'ın kodu nedir?  
A) QYVFG B) QCVGFR C) QCPGVR D) GMPGFR  
E) QCVXFF F) Hiçbiri
- 

18. Eğer PHP'nin kodu QLY ise. SQL'in kodu nedir?  
A) TUU B) TUS C) TRM D) TUB  
E) VUS F) Hiçbiri
- 

**19 ve 20. soruları aşağıdaki şekle göre cevaplayınız.**

19. 6E kutusunda Doğu'ya doğru bakmaktasınız. 3 kutu ilerleyin. Saat yönünde 90 derece dönün, iki kutu ilerleyin, saat yönünün tersinde 180 derece dönün. 5 kutu ilerleyin, saat yönünün tersine 90 derece dönün. 4 kutu ilerleyin, saat yönünde 90 derece dönün. İki kutu geri gidin. Şu anda bulunduğunuz konumun Y koordinatı nedir?  
A) 6 B) 7 C) 8 D) 9 E) 10 F) 11
- 

20. 6E kutusunda Güney-Batı'ya doğru bakmaktasınız. Üç kutu ilerleyin. Saat yönünde 135 derece dönün. 4 kutu ilerleyin. Saat yönünde 45 derece dönün. 2 kutu ilerleyin. Saat yönünün tersine 90 derece dönün ve 4 kutu geri gidin. Bulduğunuz konumun X koordinatı nedir?  
A) C B) D C) E D) F E) G F) H
- 

21. Saat 10:30'da akrep ve yelkovan arasındaki açı kaç derecedir?  
A) 75 B) 90 C) 105 D) 120 E) 135 F) 150
- 

Bir çikolata fabrikasında, bir makine 1 kg'lık çikolata kalıplarını alıp 10gr'lık diktörtgenler halinde kesmektedir. Daha sonra bu diktörtgen çikolatalar, 6gr lık silindirler haline getiriliyor. Artan çikolatalar boşa gitmektedir. Silindir halindeki

çikolatalardan 4 tanesi bir poşete konulmakta ve bu poşetlerden 6 tanesi bir kutuya konularak paktetlenmektedir.

22. Bu kutulardan tam dolu olanları, orjinal kilogramın yüzde kaçına denk gelir?  
A) %58 B) %60 C) %62 D) %64 E) %66 F) %68
- 

*23, 24 ve 25. soruları aşağıdaki şekle göre cevaplayınız. Burada bir hücrenin koordinatları o hücrenin değerini verir.*

23. A4, D3 ile çarpılıp C2'ye bölünürse sonuç kaç olur?  
A) 24 B) 26 C) 28 D) 30 E) 33 F) Hiçbiri
- 

24. B4 artı A2 sonucunu F1'de saklayın. A4 eksi D2 sonucunu F3'de saklayın. F1 ile F3 çarpımı kaç olur?  
A) 17 B) 84 C) 96 D) 104 E) 108 F) Hiçbiri
- 

Basamak 1: C3 ile D4'ü çarpın sonucu F4'de saklayın

Basamak 2: F4'ü 3 ile çarpın ve sonucu F4'de saklayın. E3'e 1 ekleyin

Basamak 3: E3'ün değeri 3 olana kadar Basamak 2'yi tekrarlayın

25. F4'ün değeri nedir?  
A) 45 B) 345 C) 405 D) 450 E) 1215 F) Hiçbiri
- 

26. A1+B3+C4+D2 sonucunu E2'de saklayın. A3+B1+C2+D4 sonucunu E4'de saklayın. Eğer E4, E2'den büyükse, değerlerini deęiş tokuş (swap) yapın, deęilse olduęu gibi bırakın. E2 ile D1'i çarpıp sonuçtan A4'ü çıkarın ve sonucu F2'de saklayın. F2'nin değeri kaçtır.  
A) 79 B) 83 C) 96 D) 95 E) 94 F) Hiçbiri
-

## Ek 2 Mantıksal Düşünme Yeteneği Testi

**AÇIKLAMA:** Bu test, çeşitli alanlarda, özellikle Fen ve Matematik dallarında karşılaşabileceğiniz problemlerde neden - sonuç ilişkisini görüp, problem çözme stratejilerini ne derece kullanabileceğinizi göstermesi açısından çok faydalıdır. Bu test içindeki sorular mantıksal ve bilimsel olarak düşünmeyi gösterecek cevapları içermektedir.

**NOT:** Soru Kitapçığı üzerinde herhangi bir işlem yapmayınız ve cevaplarınızı yalnızca cevap kağıdına yazınız. CEVAP KAĞIDINI doldururken dikkat edilecek hususlardan birisi, 1 den 8 e kadar olan sorularda her soru için cevap kağıdında iki kutu bulunmaktadır. Soldaki ilk kutuya sizce sorunun uygun cevap şikkını yazınız, ikinci kutucuğa yani AÇIKLAMASI yazılı kutucuğa ise o soruyla ilgili soru kitapçığındaki Açıklaması kısmındaki şıkları okuyarak sizce en uygun olanını seçiniz. Örneğin 12 'nci sorunun cevabı sizce b ise ve Açıklaması kısmındaki en uygun açıklama ikinci şık ise cevap kağıdını aşağıdaki gibi doldurun:

12.  AÇIKLAMASI

9. ve 10. soruları ise soru kitapçığında bu sorularla ilgili kısımları okurken nasıl cevaplayacağınızı daha iyi anlayacaksınız.

**SORU 1:** Bir boyacı, aynı büyüklükteki altı odayı boyamak için dört kutu boyakullandığına göre sekiz kutu boya ile yine aynı büyüklükte kaç oda boyayabilir?

- a. 7 oda
- b. 8 oda
- c. 9 oda
- d. 10 oda
- e. Hiçbiri

**Açıklaması:**

1. Oda sayısının boya kutusuna oranı daima  $\frac{3}{2}$  olacaktır.
2. Daha fazla boya kutusu ile fark azalabilir.
3. Oda sayısı ile boya kutusu arasındaki fark her zaman iki olacaktır.
4. Dört kutu boya ile fark iki olduğuna göre, altı kutu boya ile fark yine iki olacaktır.
5. Ne kadar çok boyaya ihtiyaç olduğunu tahmin etmek mümkün değildir.

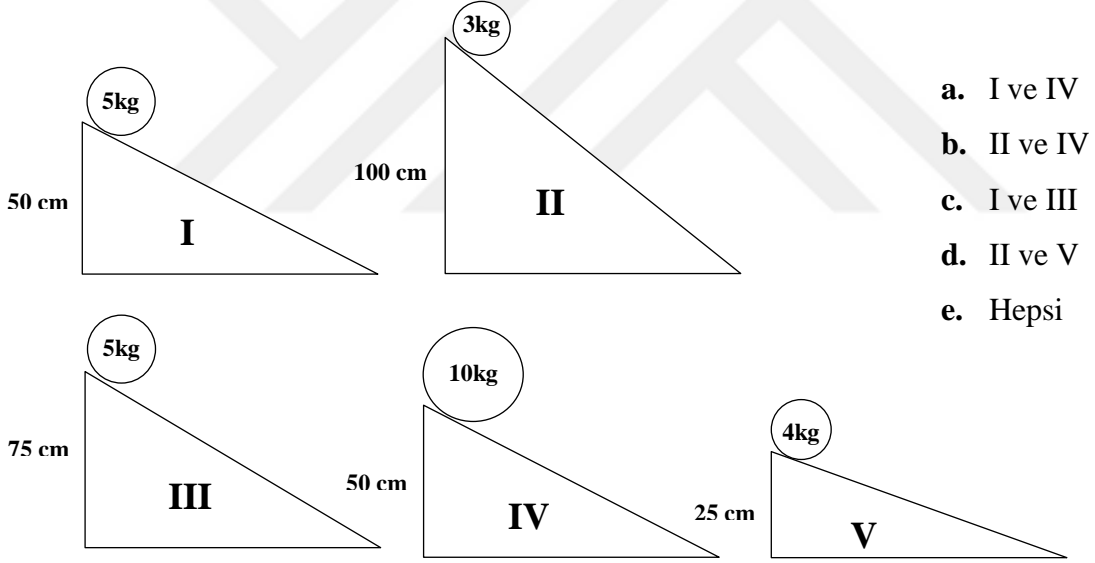
**SORU 2:** On bir odayı boyamak için kaç kutu boya gerekir? (Birinci soruya bakınız)

- a. 5 kutu
- b. 7 kutu
- c. 8 kutu
- d. 9 kutu
- e. Hiçbiri

**Açıklaması:**

1. Boya kutusu sayısının oda sayısına oranı daima  $\frac{2}{3}$  dür.
2. Eğer beş oda daha olsaydı, üç kutu boya daha gerekecekti.
3. Oda sayısı ile boya kutusu arasındaki fark her zaman ikidir.
4. Boya kutusu sayısı oda sayısının yarısı olacaktır.
5. Boya miktarını tahmin etmek mümkün değildir.

**SORU 3:** Topun eğik bir düzlemden (rampa) aşağı yuvarlandıktan sonra kat ettiği mesafe ile eğik düzlemin yüksekliği arasındaki ilişkiyi bulmak için deney yapmak



- a. I ve IV
- b. II ve IV
- c. I ve III
- d. II ve V
- e. Hepsi

isterseniz, aşağıda gösterilen hangi eğik düzlem setlerini kullanırdınız?

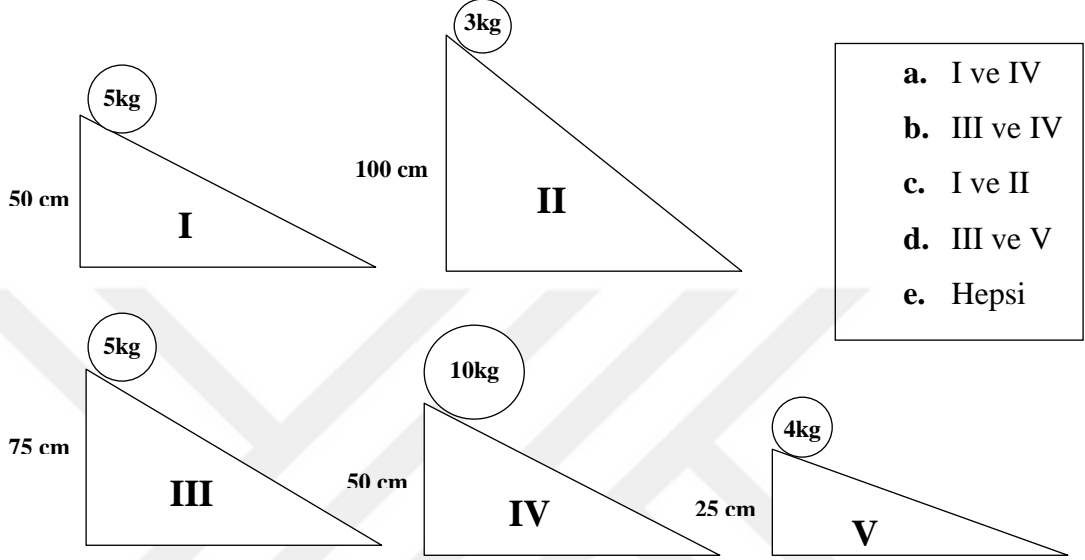
**Açıklaması:**

1. En yüksek eğik düzleme (rampa) karşı en alçak olan karşılaştırılmalı
2. Tüm eğik düzlem setleri birbiriyle karşılaştırılmalıdır.
3. Yükseklik arttıkça topun ağırlığı azalmalıdır.
4. Yükseklikler aynı fakat top ağırlıkları farklı olmalıdır.



5. Yükseklikler farklı fakat top ağırlıkları aynı olmalıdır.

**SORU 4:** Tepeden yuvarlanan topun eğik düzlemden (rampa) aşağı yuvarlandıktan sonra kat ettiği mesafenin topun ağırlığıyla olan ilişkisini bulmak için bir deney



yapmak isterseniz, aşağıda verilen hangi eğik düzlem setlerini kullanırdınız?

**Açıklaması:**

1. En ağır olan top en hafif olan ile kıyaslanmalıdır.
2. Tüm eğik düzlem setleri birbiriyle karşılaştırılmalıdır.
3. Topun ağırlığı artıkça, yükseklik azaltılmalıdır.
4. Ağırlıklar farklı fakat yükseklikler aynı olmalıdır.
5. Ağırlıklar aynı fakat yükseklikler farklı olmalıdır.

**SORU 5:** Bir Amerikalı turist Şark Ekspresi'nde altı kişinin bulunduğu bir kompartımana girer. Bu kişilerden üçü yalnızca İngilizce ve diğer üçü ise yalnızca Fransızca bilmektedir. Amerikalının kompartımana ilk girdiğinde İngilizce bilen biriyle konuşma olasılığı nedir?

- a. 2 de 1

- b. 3 de 1
- c. 4 de 1
- d. 6 da 1
- e. 6 da 4

**Açıklaması:**

1. Ard arda üç Fransızca bilen kişi çıkabildiği için dört seçim yapılması gerekir.
2. Mevcut altı kişi arasından İngilizce bilen bir kişi seçilmelidir.
3. Toplam üç İngilizce bilen kişiden sadece birinin seçilmesi yeterlidir.
4. Kompartımandakilerin yarısı İngilizce konuşur.
5. Altı kişi arasından, bir İngilizce bilen kişinin yanı sıra, üç tanede Fransızca bilen kişi seçilebilir.

**SORU 6:** Üç altın, dört gümüş ve beş bakır para bir torbaya konulduktan sonra, dört altın, iki gümüş ve üç bakır yüzük de aynı torbaya konur. İlk denemede torbadan altın bir nesne çekme olasılığı nedir?

- a. 2 de 1
- b. 3 de 1
- c. 7 de 1
- d. 21 de 1
- e. Yukarıdakilerden hiçbiri

**Açıklaması:**

1. Altın, gümüş ve bakırdan yapılan nesnelere arasında bir altın nesne seçilmelidir.
2. Paraların  $\frac{1}{4}$ ' ü ve yüzüklerin  $\frac{4}{9}$ 'u altından yapılmıştır.

3. Torbadan çekilen nesnenin para ve yüzük olması önemli olmadığı için toplam 7 altın nesneden bir tanesinin seçilmesi yeterlidir.
4. Toplam yirmi bir nesneden bir altın nesne seçilmelidir.
5. Torbadaki 21 nesnenin 7 si altından yapılmıştır.

**SORU 7:** Altı yaşındaki Ahmet'in şeker almak için 50 lirası vardır. Bakkaldaki kapalı iki şeker kutusundan birinde 30 adet kırmızı ve 50 adet sarı renkte şeker bulunmaktadır. İkinci bir kutuda ise 20 adet kırmızı ve 30 adet sarı şeker vardır. Ahmet kırmızı şekerleri sevmektedir. Ahmet'in ikinci kutudan kırmızı şeker çekme olasılığı birinci kutuya göre daha fazla mıdır?

- a. Evet
- b. Hayır

**Açıklaması:**

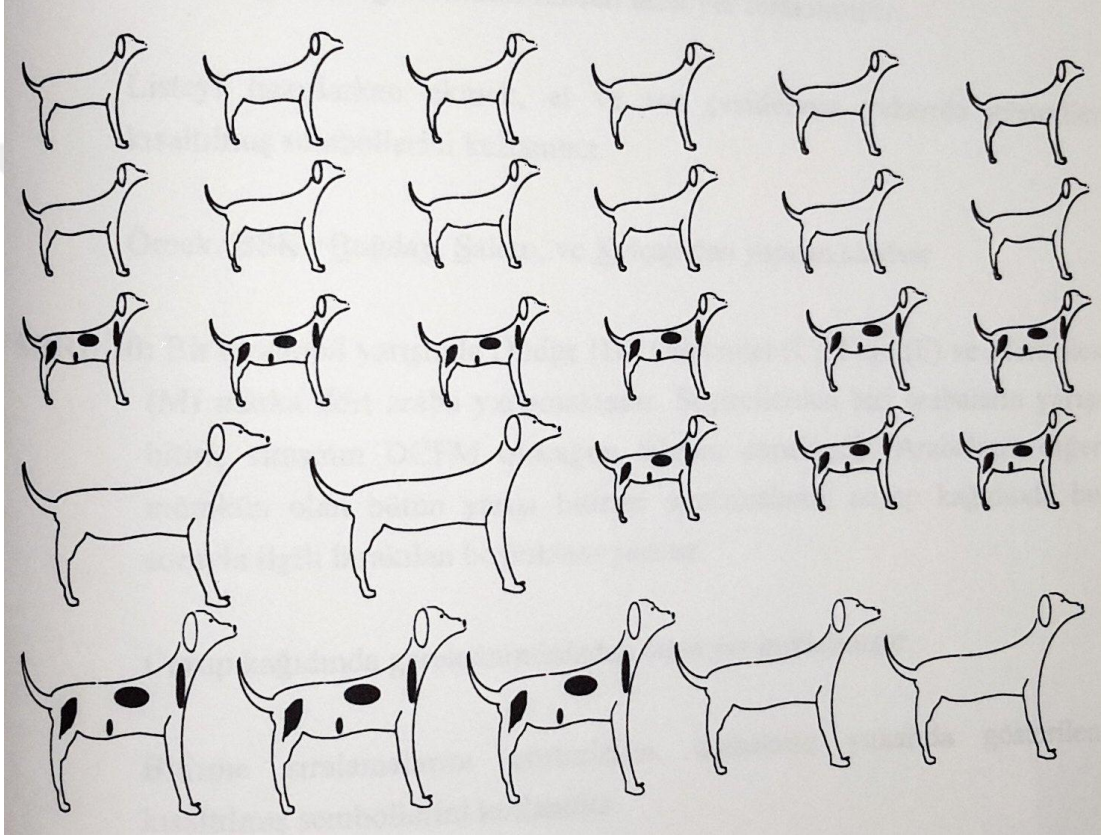
1. Birinci kutuda 30, ikincisinde ise yalnızca 20 kırmızı şeker vardır.
2. Birinci kutuda 20 tane daha fazla sarı şeker, ikincisinde ise yalnızca 10 tane daha fazla sarı şeker vardır.
3. Birinci kutuda 50, ikincisinde ise yalnızca 30 sarı şeker vardır.
4. İkinci kutudaki kırmızı şekerlerin oranı daha fazladır.
5. Birinci kutuda daha fazla sayıda şeker vardır.

**SORU 8:** 7 büyük ve 21 tane küçük köpek şekli aşağıda verilmiştir. Bazı köpekler benekli bazıları ise beneksizdir. Büyük köpeklerin benekli olma olasılıkları küçük köpeklerden daha fazla mıdır?

- a. Evet
- b. Hayır

**Açıklaması:**

1. Bazı küçük köpeklerin ve bazı büyük köpeklerin benekleri vardır.
2. Dokuz tane küçük köpeğin ve yalnızca üç tane büyük köpeğin benekleri vardır.
3. 28 köpekten 12 tanesi benekli ve geriye kalan 16 tanesi beneksizdir.
4. Büyük köpeklerin  $\frac{3}{7}$ 'si ve küçük köpeklerin  $\frac{9}{21}$ 'i beneklidir.
5. Küçük köpeklerden 12 sinin, fakat büyük köpeklerden ise sadece 4'ünün beneği yoktur.



**SORU 9:** Bir pastanede üç çeşit ekmek, üç çeşit et ve üç çeşit sos kullanılarak sandviçler yapılmaktadır.

Ekmek Çeşitleri

Et Çeşitleri

Sos Çeşitleri

Buğday (B)

Salam (S)

Ketçap (K)

Çavdar (Ç)

Piliç (P)

Mayonez (M)

Yulaf (Y)

Hindi (H)

Tereyağı (T)

Her bir sandviç ekmek, et ve sos içermektedir. Yalnızca bir ekmek çeşidi, bir et çeşidi kullanılarak kaç çeşit sandviç hazırlanabilir?

Cevap kağıdı üzerinde bu soruyla ilgili bırakılan boşluklara bütün olası sandviç çeşitlerinin listesini çıkarın.

Cevap kağıdında gereksiniminizden fazla yer bırakılmıştır.

Listeyi hazırlarken ekmek, et ve sos çeşitlerinin yukarıda gösterilen kısaltılmış sembollerini kullanınız.

Örnek: BSK= Buğday, Salam ve Ketçap dan yapılan sandviç

**SORU 10:** Bir otomobil yarışında Dodge (D), Chevrolet (C), Ford (F) ve Mercedes(M) marka dört araba yarışmaktadır. Seyircilerden biri arabaların yarışı bitiriş sırasının DCFM olacağını tahmin etmektedir. Arabaların diğer mümkün olan bütün yarışı bitirme sıralamalarını cevap kağıdında bu soruyla ilgili bırakılan boşluklara yazınız.

Cevap kağıdında gereksiniminizden fazla yer bırakılmıştır.

Bitirme sıralamalarını gösterirken, arabaların yukarıda gösterilen kısaltılmış sembollerini kullanınız.

Örnek: DCFM yarışı sırasıyla önce Dodge'nin, sonra Chevrolet'in, sonra Ford'un ve en sonra Mercedes'in bitirdiğini gösterir.

### Ek 3 Kolb Öğrenme Stilleri Envanteri III

**Ad – Soyad:**

**Numara:**

Aşağıdaki yarım bırakılan cümleler öğrenme stilinizi belirlemek için hazırlanmıştır. Lütfen her bir ifadeyi dikkatlice okuyunuz ve yarım bırakılan cümleyi tamamlayacak ifadeleri sizce en uygun olacak şekilde sıralayın. En uygun olan için 4 puan vererek en az uygun olana doğru 3, 2, 1 şeklinde puan veriniz. Aşağıda bu işlemi nasıl yapacağımızı göstermek için bir örnek verilmiştir.

Örnek,

Öğrenirken,

\_\_3\_\_ Mutlu olurum

\_\_2\_\_ Dikkatli olurum

\_\_1\_\_ Hızlı davranırım

\_\_4\_\_ Kendi fikrimi oluştururum

Verdiğiniz yanıtlar, kesinlikle size not vermek ya da sizi eleştirmek amacıyla kullanılmayacaktır. Ölçekte vereceğiniz cevaplarınız tamamıyla gizli tutulacak ve sadece araştırmacılar tarafından değerlendirilecektir; elde edilecek bilgiler bilimsel yayımlarda kullanılacaktır.

**1. Öğrenirken ...,**

\_\_\_ Duygularımı da öğrenmeye katarım.

\_\_\_ Öğrendiğim fikirler üzerinde düşünmeyi severim.

\_\_\_ Bir şeyler yapıyor olmaktan hoşlanırım.

\_\_\_ İzlemekten ve dinlemekten hoşlanırım.

**2. En iyi öğrenme yolum...,**

\_\_\_ Dikkatle dinlemek ve izlemektir.

\_\_\_ Kendi mantığımla yorumlamaktır.

\_\_\_ Duygularıma ve sezgilerime güvenmektir.

\_\_\_ Çok çalışıp bir şeyleri başarmaktır.

**3. Öğrenirken...,**

\_\_\_ Mantığıma uygun olan sonucu bulmaya çalışırım.

\_\_\_ Öğrenmede sorumlu olduğumu hissederim.

\_\_\_ Derse katılmadan sessizce izlerim.

\_\_\_ Derse yoğun bir şekilde katılırım.

**4. En iyi...,**

\_\_\_ Duygularımla öğrenirim.

\_\_\_ Yaparak öğrenirim.

\_\_\_ İzleyerek öğrenirim.

\_\_\_ Fikirler üzerinde düşünerek öğrenirim.

**5. Öğrenirken...,**

- Konuyla ilgili yeni bilgilere/fikirlerle açığım.
- Konuyu her yönüyle/ayrıntılarıyla ele alırım.
- Konuyu kendi içinde küçük bölümlere ayırırım.
- Konuyla ilgili öğrendiğim şeyleri yapmaktan/uygulamaktan hoşlanırım

**6. Öğrenirken...,**

- Gözlem yapan biriyim.
- Öğrenmeye katılan biriyim.
- Duygularıyla hareket eden biriyim.
- Mantıklı davranan biriyim.

**7. En iyi öğrenme yolum...,**

- Konuyla ilgili gözlem yapmaktır.
- İnsanlarla konuyla ilgili konuşmak, iletişim kurmaktır.
- Konunun dayandığı temel fikirleri düşünmektir.
- Konuyla ilgili deneme ve uygulama yapmaktır.



**8. Öğrenirken...,**

- \_\_\_ Çalışmamın sonuçlarını görmekten hoşlanırım.
- \_\_\_ Konuyla ilgili temel fikirleri düşünmeyi severim.
- \_\_\_ Acele etmekten hoşlanmam.
- \_\_\_ Kendimi tamamen öğrenme işinin içinde hissederim.

**9. En iyi öğrenme yolum...,**

- \_\_\_ İzlemektir.
- \_\_\_ Hissettiklerimi dikkate almaktır.
- \_\_\_ Öğrendiklerimi uygulamaktır.
- \_\_\_ Kendi düşüncelerimi dikkate almaktır.

**10. Öğrenirken...,**

- \_\_\_ Çekingen biri olurum.
- \_\_\_ Öğrendiklerimi sorgulamadan kabul ederim.
- \_\_\_ Sorumluluklarını bilen biriyim.
- \_\_\_ Öğrendiğim şeyler üzerinde düşünen biriyim.

**11. Öğrenirken...,**

\_\_\_ Derse katılırım.

\_\_\_ Derse katılmadan izlerim.

\_\_\_ Öğrendiklerimi değerlendiririm.

\_\_\_ Aktif olmaktan hoşlanırım.

**12. En iyi öğrenme yolum...,**

\_\_\_ Anlatılan fikirleri (konuları) tek tek ele almaktır.

\_\_\_ Yeni fikirleri öğrenmeye açık olmaktır.

\_\_\_ Dikkatli olmaktır.

\_\_\_ Anlatılanları uygulamaktır.

## Ek 4 Programlamaya İlişkin Öz Yeterlilik Ölçeği

Merhaba, bu anket sizin programlamaya ilişkin öz yeterlilik algınızı ölçmek için hazırlanmıştır. Kullandığınız herhangi bir programlama dilini esas alarak maddeleri cevaplayabilirsiniz.

Cinsiyet – Yaşınız: \_

Bugüne kadar programlamaya ilişkin ya da programlama içeren kaç ders aldınız?.

Hangi programlama dillerini kullanabiliyorsunuz?

.

Kaç yıldır bilgisayar kullanıyorsunuz?.

Kaç yıldır program yazıyorsunuz?.

Aşağıdaki programlama görevlerini yaparken kendinize olan güveninizi 1 - 7 arasında puanlayınız	Hiç Güvenmiyorum	Genellikle güveniyorum	Biraz Güveniyorum	Yarı yarıya güveniyorum	Oldukça Güveniyorum	Genellikle Güveniyorum	Tamamen güveniyorum
	1	2	3	4	5	6	7
1. "Merhaba Dünya" mesajının görüntülenebileceği bir program yazabilirim.							
2. Üç sayının ortalamasını hesaplayan bir program yazabilirim.							
3. Verilen herhangi bir sayı dizisinin ortalamasını hesaplayan bir program yazabilirim.							
4. İstenilenler açıkça tanımlandığında bir problemin çözümüne yönelik oldukça karmaşık ve uzun bir program yazabilirim.							

5. Yazacağım bir programı modüler bir biçimde organize edip tasarlayabilirim							
6. Yazdığım uzun ve karmaşık bir programdaki tüm hataları ayıklayabilir ve çalışabilir hale getirebilirim.							
7. Uzun, karmaşık ve birden fazla dosya gerektiren bir programı kavrayabilirim.							
8. Bir programın daha okunabilir ve açık olması için uzun ve karmaşık kısımları yeniden yazabilirim.							
9. Çevrede bir sürü dikkat dağıtıcı olsa bile programa odaklanma yollarını bulabilirim.							