

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

MESLEK YÜKSEK OKULU ÖĞRENCİLERİNİN PROGRAMLAMA
BAŞARISINI ETKİLEYEN FAKTÖRLERİN İNCELENMESİ

YÜKSEK LİSANS TEZİ

Hazırlayan
Fahri YILMAZ

Ankara
Nisan – 2013

Fahri YILMAZ

Nisan, 2013

MESLEK YÜKSEK OKULU ÖĞRENCİLERİNİN PROGRAMLAMA
BAŞARISINI ETKİLEYEN FAKTÖRLERİN İNCELENMESİ

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

MESLEK YÜKSEK OKULU ÖĞRENCİLERİNİN PROGRAMLAMA
BAŞARISINI ETKİLEYEN FAKTÖRLERİN İNCELENMESİ

YÜKSEK LİSANS TEZİ

Fahri YILMAZ

Danışman: Yrd. Doç. Dr. Hasan ÇAKIR

Ankara
Nisan – 2013

JÜRİ ONAY SAYFASI

Fahri YILMAZ'ın **Meslek Yüksek Okulu Öğrencilerinin Programlama Başarısını Etkileyen Faktörlerin İncelenmesi** başlıklı tezi 21/05/2013 tarihinde, jürimiz tarafından Bilgisayar ve Öğretim Teknolojileri Eğitimi Ana Bilim Dalında Yüksek Lisans Tezi olarak kabul edilmiştir.

	<u>Adı Soyadı</u>	<u>İmzası</u>
Üye (Tez Danışmanı)	: Yrd. Doç. Dr. Hasan ÇAKIR	
Üye:	: Prof. Dr. Ahmet MAHİROĞLU	
Üye:	: Doç. Dr. Sami ŞAHİN	

ÖNSÖZ

Meslek yüksekokulu öğrencilerinin programlama başarılarını etkileyen faktörlerin incelenmesi amacıyla yapılan bu araştırma Gazi Üniversitesi Eğitim Bilimleri Enstitüsü Bilgisayar ve Öğretim Teknolojileri Eğitimi Ana Bilim Dalı'nda yüksek lisans tez çalışması olarak gerçekleştirilmiştir.

Eğitim öğretim hayatım boyunca, bana okumayı öğreten ilk öğretmenim Refiye KOÇ'tan, araştırmam boyunca bilgi ve deneyimiyle bana sabırla yol gösteren danışmanım Yrd. Doç. Dr. Hasan ÇAKIR'a kadar üzerimde emeği geçen bütün öğretmenlerime sonsuz teşekkür ederim.

Ayrıca, manevi desteğiyle her zaman yanımda olan eşim Ayşegül'e teşekkür ederim.

Fahri YILMAZ

ANKARA - 2013

ÖZET

MESLEK YÜKSEK OKULU ÖĞRENCİLERİNİN PROGRAMLAMA BAŞARISINI ETKİLEYEN FAKTÖRLERİN İNCELENMESİ

YILMAZ, Fahri

Yüksek Lisans, Bilgisayar ve Öğretim Teknolojileri Eğitimi Ana Bilim Dalı

Tez Danışmanı: Yrd. Doç. Dr. Hasan ÇAKIR

Nisan - 2013

Bu tez çalışmasının amacı, meslek yüksek okulu öğrencilerinin programlama başarılarını etkileyen faktörleri ortaya çıkarmak ve öğrencilerin programlama başarıları ile demografik bilgileri (cinsiyet, mezun oldukları okul türü, programlamayla ilgili geçmiş deneyimleri, genel akademik not ortalamaları), problem çözme ve analitik düşünme becerisi, kişisel öz yeterlilik ve motivasyonları, öz düzenleyici öğrenme stratejileri (hedef belirleme, içsel hedef, dışsal hedef, öz yeterlilik, öz yansıma, değer verme, üst biliş) arasında anlamlı bir ilişki bulunup bulunmadığını incelemektir.

Araştırmada ilişkisel tarama yöntemi kullanılmıştır. Araştırmanın bağımlı değişkeni öğrencilerin programlama dersindeki akademik başarıları; bağımsız değişkenleri ise demografik bilgiler, motivasyon, öz yeterlilik, öz düzenleyici öğrenme stratejileridir. Araştırmada veri toplama aracı olarak demografik bilgi anketi, problem çözme envanteri, öz düzenleyici öğrenme stratejileri ölçeği, genel öz yeterlilik ölçeği, güdülenme ve öğrenme stratejileri ölçeği, programlama başarı testi kullanılmıştır.

Araştırma, Bozok Üniversitesi Akdağmadeni MYO, Adnan Menderes Üniversitesi Aydın MYO, Ahi Evran Üniversitesi Kaman MYO bilgisayar programcılığı programı 2. sınıf öğrencileri üzerinde uygulanmıştır.

Araştırma sonunda, öğrencilerin programlama başarıları ile cinsiyet, okul türü, genel akademik not ortalamaları, motivasyon, öz düzenleyici öğrenme stratejilerinden öz yeterlilik arasında anlamlı ilişki bulunmuştur. Programlama başarıları toplam

varyansının %23'ü demografik özellikler, problem çözme ve analitik düşünme becerisi, kişisel öz yeterlilik, motivasyon değişkenleriyle, %18'i öz düzenleyici öğrenme stratejileri ile açıklanabilmektedir.

Anahtar Kelimeler: programlama başarısı, meslek yüksek okulu, programlama, programlama eğitimi

ABSTRACT

INVESTIGATION OF FACTORS AFFECTING TWO-YEAR VOCATIONAL COLLEGE STUDENTS' COMPUTER PROGRAMMING ACHIEVEMENT

YILMAZ, Fahri

Master of Science, Computer Education and Instructional Technology

Advisor: Assist. Prof. Dr. Hasan ÇAKIR

April - 2013

The purpose of this study is to explore the factors that influence two-year vocational college students' achievement in computer programming and to examine whether a relationship exists between students' achievement in computer programming with demographic information (gender , the type of school that they have graduated from , past experience with programming , overall academic grade point averages) , problem-solving and analytical thinking skills , personal self-efficacy and motivation , self-regulated learning strategies , (goal setting , internal target , external target , self-efficacy , self-reflection , valuing , metacognition).

Correlational method was utilized for this study . The dependent variable of the research is the academic achievement of students in a computer programming course. The independent variables are demographic information , motivation , self-efficacy , self-regulated learning strategies. Demographic information questionnaire , problem solving inventory , the scale of self-regulated learning strategies , general self-efficacy scale and programming achievement test were used as data collection instruments in the research .

The research was conducted in two- year vocational college students of computer programming departments in Bozok University-Akdağmadeni two-year vocational college, Adnan Menderes University-Aydın two-year vocational college, Ahi Evran University- Kaman two-year vocational college.

Results showed that a significant relationship exists between the student achievement in computer programming and gender, type of school , overall academic

grade point averages , motivation, self-efficacy from self-regulated learning strategies. 23% of the total variance of the student achievement in computer programming can be explained by demographic characteristics , problem-solving and analytical thinking skills , personal self-efficacy , motivational variables and 18% can be explained by self-regulated learning strategies.

Keywords: the student achievement in computer programming, two-year vocational college, , computer programming training.

İÇİNDEKİLER

ÖNSÖZ	i
ÖZET	ii
ABSTRACT.....	iv
İÇİNDEKİLER	vi
TABLolar LİSTESİ.....	viii
1. GİRİŞ	1
1.1 Problem Durumu.....	1
1.2 Araştırmanın Amacı.....	4
1.3 Araştırmanın Önemi	4
1.4 Araştırmanın Varsayımları	8
1.5 Araştırmanın Sınırlılıkları.....	8
1.6 Tanımlar.....	9
2. KAVRAMSAL ÇERÇEVE.....	10
2.1 Programlama	10
2.1.1 Programlamada Temel Kavramlar	12
2.1.2 Programlama Eğitimi.....	14
2.2 Programlama Başarısına Etki Eden Demografik Bilgiler	16
2.2.1 Cinsiyet	16
2.2.2 Programlama Deneyimi ve Ön Bilgi	17
2.2.3 Öğrencilerin Mezun Oldukları Okul Türü ve Mezuniyet Puanları.....	19
2.3 Motivasyon ve Öz Yeterlilik Algısı.....	21
2.4 Problem Çözme ve Analitik Düşünme Becerisi	22
2.5 Öz Düzenleyici Öğrenme Stratejileri.....	23
3. YÖNTEM	26
3.1 Araştırma Modeli	26
3.2 Evren ve Örneklem	26
3.3 Veri Toplama Araçları	27
3.3.1 Demografik bilgiler anketi.....	28
3.3.2 Heppner'in problem çözme envanteri.....	28
3.3.3 Öz düzenleyici öğrenme stratejileri ölçeği	29
3.3.4 Öz yeterlilik ölçeği.....	30
3.3.5 Güdülenme ve öğrenme stratejileri ölçeği (MSLQ)	30
3.3.6 Başarı sınavı.....	31
3.4 Verilerin Toplanması	34
3.5 Verilerin Analizi	35
4. BULGULAR VE YORUM	36
4.1 Çalışmaya Katılan Öğrencilerin Demografik Verilerine İlişkin Bulgular.....	36
4.2 MYO Öğrencilerinin Programlama Başarılarının Çalışmada Ele Alınan Bağımsız Değişkenlere Göre İncelenmesi	38
4.3 MYO Öğrencilerinin Programlama Başarıları İle Demografik Bilgileri (Cinsiyet, Mezun Oldukları Okul Türü, Programlamayla İlgili Geçmiş Deneyimleri, Genel Akademik Not Ortalamaları), Problem Çözme Ve Analitik Düşünme Becerisi, Kişisel Öz Yeterlilik, Motivasyonları Arasındaki İlişkiye Ait Bulgular	41

4.4. MYO Öğrencilerinin Programlama Başarısı İle Öz Düzenleyici Öğrenme Stratejileri (Hedef Belirleme, İçsel Hedef, Dışsal Hedef, Öz Yeterlilik, Öz Yansıma, Değer Verme, Üst Biliş) Arasındaki İlişkiye Ait Bulgular	44
5. SONUÇ VE ÖNERİLER.....	46
5.1 Öğrencilerin Demografik Bilgilerine Göre Elde Edilen Sonuçlar.....	46
5.2 MYO Öğrencilerinin Programlama Başarıları İle Demografik Bilgileri (Cinsiyet, Mezun Oldukları Okul Türü, Programlamayla İlgili Geçmiş Deneyimleri, Genel Akademik Not Ortalamaları), Problem Çözme Ve Analitik Düşünme Becerisi, Kişisel Öz Yeterlilik, Motivasyonları Arasındaki İlişkiye Ait Sonuçlar	48
5.3 MYO Öğrencilerinin Programlama Başarısı İle Öz Düzenleyici Öğrenme Stratejileri (Hedef Belirleme, İçsel Hedef, Dışsal Hedef, Öz Yeterlilik, Öz Yansıma, Değer Verme, Üst Biliş) Arasındaki İlişkiye Ait Sonuçlar	49
5.4 Öneriler	49
5.4.1 Uygulamaya yönelik öneriler.....	49
5.4.2 Yeni yapılacak araştırmalara yönelik öneriler	50
KAYNAKÇA.....	52
EKLER.....	59

TABLÖLAR LİSTESİ

Tablo 1. MYO Bilgisayar Programcılığına Ait Sayısal Bilgiler.....	6
Tablo 2. Bilgisayar Mühendisliğine Ait Sayısal Bilgiler.....	6
Tablo 3. Yazılım Mühendisliğine Ait Sayısal Bilgiler	7
Tablo 4. Meslek Lisesi Bilişim Teknolojileri Bölümü Haftalık Alan Dersleri Çizelgesi (MEB, 2011).....	20
Tablo 5. Araştırmaya Katılan Öğrencilerin Okul ve Cinsiyete Göre Dağılımları	27
Tablo 6. Başarı Testine İlişkin Belirtke Tablosu	31
Tablo 7. Uygulanan Başarı Sınavları İçerisinden Rastgele Seçilen 30 Testin Sonuçları İçin Puanlayıcıların Verdikleri Puanların Güvenilirliği.....	33
Tablo 8. Çalışmaya Katılan Öğrencilerin Cinsiyete Göre Dağılımları.....	36
Tablo 9. Çalışmaya Katılan Öğrencilerin Deneyimlerine Göre Dağılımları	36
Tablo 10. Çalışmaya Katılan Öğrencilerin Mezun Oldukları Lise Türüne Göre Dağılımları	37
Tablo 11. Çalışmaya Katılan Öğrencilerin GANO'ya Göre Dağılımları	37
Tablo 12. Çalışmaya Katılan Öğrencilere Ait Bilgiler	38
Tablo 13. Programlama Başarısının Cinsiyete Göre Bağımsız Grup T-Testi İle Karşılaştırılması	38
Tablo 14. Programlama Başarısının Deneyime Göre Bağımsız Grup T-Testi İle Karşılaştırılması	39
Tablo 15. Programlama Başarısının GANO'ya Göre Bağımsız Grup T-Testi İle Karşılaştırılması	39
Tablo 16. Programlama Başarısının Mezun Olunan Lise Türüne Göre Ortalamaları ve Standart Sapmaları	40
Tablo 17. Programlama Başarısının Öğrencilerin Mezun Oldukları Lise Türüne Göre Tek Yönlü Varyans Analizi İle Karşılaştırılması	40
Tablo 18 MYO Öğrencilerinin Programlama Başarıları İle Demografik Bilgileri, Problem Çözme ve Analitik Düşünme Becerisi, Kişisel Öz Yeterlilik , Motivasyonları Arasındaki İlişkiye Ait Regresyon Analizi Sonuçları.....	42

Tablo 19. MYO Öğrencilerinin Programlama Başarısı İle Öz Düzenleyici Öğrenme Stratejileri Arasındaki İlişkiye Ait Regresyon Analizi Sonuçları.....	44
--	----

1. GİRİŞ

Yazılım bilgisayar donanımlarının yönetimini sağlamak, insanların hesap yapmak, belge düzenlemek, veri saklamak ve getirmek ile ilgili işlerini yapmak için kullanılan programlardır (Çamoğlu, 2009). İlk defa makine dili kullanılarak geliştirilen yazılımlar günümüzde üst seviye programlama dilleri kullanılarak yapılmaktadır. İlk bilgisayar programı 1940'lı yıllarda Mark 1 ve ENIAC (Electronic Numerical Integrator and Computer) bilgisayarları için geliştirilmiştir (Marciniak, 2002). Bu dönemde yazılım basit kod satırlarıyla temsil ediliyordu. Delikli kartlardan okutulan birkaç veri yine delikli kartlardan okutulan birkaç komutla işletiliyor ve elde edilen veri yine delikli kartlara basılıyordu (Yaşar, 2011). Yazılımın bir disiplin haline gelmesi, yazılım projelerinde yaşanan problemlere bir dur demek adına 1950'li yıllarda yapılan çalışmalar sonucunda olmuştur. Yazılım (Software) kavramı ilk defa 1958 yılında John Tukey tarafından kullanılmıştır (Çamoğlu, 2009). Yazılım mühendisliği kavramı ise ilk defa 1968 yılında NATO tarafından düzenlenen konferansta ortaya çıkmıştır (Naur ve Randall, 1968). Daha sonrasında çeşitli örgütler kurularak bunların öncülüğünde yazılım mühendisliği yeni bir disiplin olarak gelişmeye başlamıştır (Çamoğlu, 2009).

Yazılım geliştirmekte kullanılan programlama dillerinin gelişimine baktığımızda ilk defa makine dilinde yazılmaya başlanan programlar günümüzde üst seviye programlama dilleri yardımıyla hazır nesneler ve sihirbazlar kullanılarak geliştirilmektedir. Bu gelişmeler program yazmayı kolaylaştırıyor olsa da program hazırlamakta hala algoritma geliştirme, hangi nesneyi ne zaman kullanacağını bilememe gibi bazı sorunlar bulunmaktadır.

1.1 Problem Durumu

Yazılım, kullanılan programlama diline özgü komutların bir araya getirilerek, bunları bir derleyici yardımıyla makine diline çevrilmesiyle oluşturulan yapılardır. Yazılım geliştirebilmek için bir bilgisayar ve derleyici yeterli olmaktadır. Ancak bunları

kullanarak ortaya kullanılabilir yazılımlar çıkarabilmek için yetişmiş insan gücü gereklidir.

Erdoğan'a (2005) göre, programlama alanında uzmanlaşmış insanlar arasında, iyi bir programcı olabilmek için bilişsel becerilerin yüksek olmasının yeterli olacağı yönünde genel bir kanı vardır. Ancak yapılan araştırmalar programlama becerisini etkileyecek birçok faktör olduğunu ortaya koymaktadır. Bu faktörler çeşitli çalışmalarda;

1. Problem çözme ve analitik düşünme becerisi (Goold ve Rimmer, 2000; Grant, 2003; Ismail, Ngah ve Umar, 2010; Pillay ve Jugoo, 2005),
2. Kullanılan materyal (Arabacıoğlu, Bülbül ve Filiz, 2007; Chalk, 2000),
3. İçsel motivasyon (Jiau, Chen ve Ssu, 2009; Matthíasdóttir, 2006),
4. Farklı öğretim stratejilerinin kullanılması (Bellardinelli, 1971; Miliszewska, Venables ve Tan, 2008),
5. Cinsiyet (Byrne ve Lyons, 2001; Goold ve Rimmer, 2000; Pillay ve Jugoo, 2005),
6. Öğrenme stilleri (Byrne ve Lyons, 2001; Goold ve Rimmer, 2000; Pillay ve Jugoo, 2005; Thomas, Ratcliffe, Woodbury ve Jarman, 2002),
7. Ön bilgi ve deneyim (Byrne ve Lyons, 2001; Goold ve Rimmer, 2000; Holden ve Weeden, 2003; Konvalina, Stephens ve Wileman, 1983; Pillay ve Jugoo, 2005),
8. Genel akademik başarı (Goold ve Rimmer, 2000; Keskinsoy, 2010),
9. Genel yetenek (Y. Erdoğan, Emin ve Kabaca, 2008; Keskinsoy, 2010),
10. Matematik başarısı (Y. Erdoğan ve diğerleri, 2008; Keskinsoy, 2010),

olarak belirlenmiştir. Görüldüğü gibi programlama başarısı tek bir nedene bağlı değildir. Programlama alanında başarılı olan öğrencilerin birçoğunun yukarıdaki niteliklere olumlu anlamda sahip oldukları görülmektedir. Yine yapılan çalışmalar incelendiğinde başka alanlarda çok başarılı olan öğrencilerin programlama alanında başarısız olabildikleri görülmüştür. Nitekim Goold ve Rimmer (2000) yaptıkları çalışma da öğrencilerin akademik başarılarının programlama becerisi üzerinde sabit bir etkisinin olmadığını bulmuşlardır. Akademik başarısı düşük olan öğrenciler programlamada

başarı gösterebildikleri gibi, akademik başarısı yüksek olan bazı öğrenciler başarı gösterememiştir. Programlama başarısı incelenirken ve programlama eğitimi verilirken sırf öğrencilerin bilişsel beceri düzeylerine bakmayıp programlamaya özgü özellikler dikkate alınmalıdır. Başarılı bir programlama eğitimi verebilmek, öğrencilerin programlama başarılarını yükseltebilmek için, programlama başarısını etkileyen faktörlerin incelenmesi gerekir.

Bilgisayar programcılığı alanında eğitim alan öğrenciler, diğer bölümlere göre daha heterojen gruplardan oluşmaktadır (Matthíasdóttir, 2006). Ülkemizde de programlama eğitimi veren bilgisayar programcılığı, bilgisayar mühendisliği, yazılım mühendisliği gibi bölümlere öğrenci alırken, meslek lisesi çıkışlı öğrencilere ek puan verilmesinin yanında, bu bölümlere girebilmek için özel bir alan şartı belirtilmemektedir (OSYM, 2011a, 2011b). Bu bölümlere yerleşen öğrencilerden kimi meslek liselerinden gelerek belli bir programlama bilgisine sahipken, kimi öğrenciler de genel liselerden gelerek daha önce herhangi bir programcılık eğitimi almamış olabiliyorlar. Öğrencilerin farklı ön bilgilere sahip olmaları, programcılık eğitiminde farklılıklar oluşturduğu düşüncesine neden olabilir. Ancak bununla ilgili çalışma yapan Pillay ve Jugoo (2005) öğrencilerin ön bilgilerinin programlama başarısı üzerinde herhangi bir etkisi olmadığını öne sürmüşlerdir. Holden ve Weeden (2003) ise öğrencilerin programlamayla ilgili ön bilgilerinin eğitim sürecinin ilk dönemlerinde etkili olduğunu, diğer dönemlerinde ise herhangi bir etkisinin olmadığını bulmuşlardır. Ancak Byrne ve Lyons(2001) bu durumun tam tersini söyleyerek ön bilgi ve deneyimin programlama başarısını etkilediğini söylemektedirler. Bu üç çalışmaya baktığımızda her ne kadar aynı özellik üzerinde çalışmış olsalar da, kullandıkları örneklem sonuçların farklı çıkmasına neden olmuştur. Dolayısıyla programlama eğitimi verilirken programlamaya özgü özelliklerin yanı sıra hitap edilen öğrenci grubu da dikkate alınmalıdır.

Tüm bu söylenenleri dikkate alırsak yazılım sektörüne yetişmiş insan gücü kazandırmak amacıyla programlama eğitimi verirken, hem öğrencilerin özellikleri, hem de öğrencilerin programlama başarılarını etkileyen faktörler dikkate alınmalıdır.

Bu çalışmada MYO öğrencilerinin programlama başarısı ile öğrencilerin cinsiyetleri, mezun oldukları okul türü, programlama derslerine karşı olan

motivasyonları, öz yeterlilik algıları, problem çözme ve analitik düşünme becerileri, öz düzenleyici öğrenme faktörleri arasındaki ilişkinin incelenmesi amaçlanmaktadır.

1.2 Araştırmanın Amacı

Bu çalışmanın amacı MYO öğrencilerinin programlama başarılarını etkileyen faktörleri ortaya çıkarmaktır.

Bu araştırmada, yukarıda belirtilen genel amaç doğrultusunda alt amaçlar olarak aşağıdaki sorulara cevap aranacaktır:

1. MYO öğrencilerinin programlama başarıları ile demografik bilgileri (cinsiyet, mezun oldukları okul türü, programlamayla ilgili üniversiteye başlamadan önce eğitim alıp almadıkları (deneyim), genel akademik not ortalamaları (GANO)), problem çözme ve analitik düşünme becerisi, kişisel öz yeterlilik ve motivasyonları arasında bir ilişki var mıdır?

2. MYO öğrencilerinin programlama başarısı ile öz düzenleyici öğrenme stratejileri (hedef belirleme, içsel hedef, dışsal hedef, öz yeterlilik, öz yansıma, değer verme, üst biliş) arasında bir ilişki var mıdır?

1.3 Araştırmanın Önemi

Yazılım sektörü ülke ekonomisine büyük katkı sağlayan bir sektördür. Özellikle doğal kaynakların yetersiz olduğu ve insanlar için yeterli istihdam sağlanamadığı durumlarda ülkeler yazılım sektörüne yönelerek, ülkeleri için katkı değer sağlamışlardır.

Yazılım, doğru bir şekilde yönlendirilirse ülkelerin gelişmesi için yararlı bir sektör olacaktır. Bunun yanı sıra yazılım diğer üretim sektörlerin değerlerine katkı sağlayabilir, ürünlerin fiyatlarını ve kalitelerini, karar mekanizmalarını, verimliliği, üretkenliği, rekabetçiliği artırabilir. Yazılım insanlar için istihdam yaratır. Enerji giderleri ve ithal girdisi çok azdır. Vergi ve ihracat giderlerini azaltır. Yazılım sektörünün güçlenmesi ve yaygınlaşması kayıt dışı ekonominin önlenmesine katkıda bulunur (Alican, 2008).

Günümüzde yazılımda öncü olan ülkeler 3I ülkeleri olarak adlandırılan Hindistan, İrlanda ve İsrail'dir. Yazılım alanında öncü olan 3I ülkelerine bakıldığında devletin sektörü geliştirmeye yönelik çabaları dikkat çekmektedir. Bu ülkelerde devlet yazılım sektörünün gelişmesi için aktif rol oynamıştır ve yazılım sektörünün gelişmesi bu ülkelerde ekonominin iyileşmesinde önemli katkı sağlamıştır (Yılmaz, 2007). Ülkemizde de son dönemlerde bilişim, özellikle de yazılım sektörüne devletin önem verdiğini, bu doğrultuda çeşitli politikalar geliştirdiğini görmekteyiz. Özellikle üniversite bünyelerinde açılan teknokentler ve devletin teknokentlerde faaliyet gösteren firmalar için hazırlamış olduğu teşvik programları yazılım sektörünün gelişmesinde, bu sektörde çalışan insanlar için istihdam yaratılmasında büyük pay sahibi olmuştur (Konuk ve Öztürk, 2010). Ancak yalnızca uygun ortamın hazırlanması yazılım alanında gelişim göstermek için yeterli değildir. Bu uygun ortamı değerlendirecek yetişmiş insan gücüne de ihtiyaç vardır (Taşkın, 2002).

Yazılım üretiminde en önemli etken kaliteli iş gücüdür. Bu alanda başarılı olan ülkelere baktığımızda başarılarını büyük ölçüde yetişmiş kaliteli insan gücüne ve iş gücü piyasalarının genişliğine borçlu olduğunu görürüz. Bu ülkelerde devlet desteği alınarak kaliteli iş gücü yetiştiren birçok eğitim kurumunun olduğunu görülmektedir (Taşkın, 2002). Ülkemizde de yazılım sektörüne eleman yetiştiren birçok kurum bulunmaktadır. Yazılım alanına eleman yetiştiren başlıca eğitim kurumları olarak orta öğretim seviyesinde meslek liselerinin bilişim bölümlerini, ön lisans seviyesinde meslek yüksek okullarının bilgisayar programcılığı bölümleri ve lisans seviyesinde üniversitelerin bilgisayar mühendisliği, yazılım mühendisliği bölümleri bulunmaktadır. Ülkemizde yazılım sektörünün yetişmiş insan gücü talebini öncelikli olarak bu bölümler sağlamaktadır. Yine bu bölümlerin dışında da birçok ön lisans ve lisans bölümlerinde programcılık eğitimi verilmektedir.

Ön lisans ve lisans düzeyinde eğitim veren bilgisayar programcılığı, bilgisayar mühendisliği ve yazılım mühendisliğine ait yüksek öğretim programları ve kontenjanları Tablo 1, Tablo 2 ve Tablo 3'te 2011 ÖSYS Yüksek Öğretim Programları ve Kontenjanlı kılavuzu kullanılarak belirtilmiştir (OSYM, 2011a, 2011b).

Tablo 1. MYO Bilgisayar Programcılığına Ait Sayısal Bilgiler

Eğitim Veren Üniversite Sayısı	87
Eğitim Veren MYO Sayısı	243
Birinci öğretim program sayısı	236
İkinci öğretim program sayısı	164
Uzaktan öğretim program sayısı	17
Birinci Öğretim Kontenjanları	11940
İkinci Öğretim Kontenjanları	8965
Uzaktan eğitim Kontenjanları	1585
Toplam Öğrenci Kontenjanı	22490

Tablo 2. Bilgisayar Mühendisliğine Ait Sayısal Bilgiler

Eğitim Veren Üniversite Sayısı	76
Birinci öğretim program sayısı	90
İkinci öğretim program sayısı	15
Uzaktan öğretim program sayısı	3
Birinci Öğretim Kontenjanları	4049
İkinci Öğretim Kontenjanları	1062
Uzaktan eğitim Kontenjanları	280
Toplam Öğrenci Kontenjanı	5391

Tablo 3. Yazılım Mühendisliğine Ait Sayısal Bilgiler

Eğitim Veren Üniversite Sayısı	13
Birinci öğretim program sayısı	19
İkinci öğretim program sayısı	2
Uzaktan öğretim program sayısı	0
Birinci Öğretim Kontenjanları	938
İkinci Öğretim Kontenjanları	83
Uzaktan eğitim Kontenjanları	0
Toplam Öğrenci Kontenjanı	1021

Yazılım sektörüne doğrudan eleman yetiştiren ön lisans ve lisans düzeyindeki bölümlerin toplam kontenjanı 28.802'dir. Bunlardan ön lisans düzeyinde eğitim veren bölümlerin toplam kontenjanı 22.490'dır. Ön lisans düzeyinde eğitim veren bölümlerin kontenjanlarının toplam kontenjanlar içerisindeki yeri %78'dir. Bilgisayar mühendisliği bölümlerinin toplam kontenjanı 5.391 ve toplam kontenjan içerisindeki yeri %18'dir. Yazılım Mühendisliğinin toplam kontenjanı 1021 ve toplam kontenjan içerisindeki yeri %4'tür (OSYM, 2011a, 2011b). Görüldüğü gibi yazılım sektörünün yetişmiş insan gücünün büyük çoğunluğunu MYO'lar karşılamaktadır.

Bu bölümlerde öğrencilere bilgisayar programcılığına ait temel vasıflar kazandırılmaya çalışılmaktadır. Yapılacak olan bu çalışma öğrencilerin programcılık başarısını etkileyen faktörleri ortaya çıkararak, bunlara yönelik tedbir ve düzeltme imkanı sağlayacaktır.

Çalışmanın MYO öğrencileri üzerinde yapılması, yazılım sektörüne eleman yetiştiren bölümler arasında MYO öğrencilerinin, tüm öğrencilerin %78'ini oluşturması açısından önemlidir. Programlama başarısını etkileyen faktörlerle ilgili yapılan çalışmaların çoğunluğu küçük örneklem üzerinde yürütülmüş ve daha önce MYO öğrencileri üzerinde herhangi bir çalışma yapılmamıştır. Daha önceki çalışmalarda küçük örneklem gruplarının seçilmesi elde edilen sonuçların örneklemin özelliğine göre farklılıklar göstermesine neden olmuştur. Örneğin bir çalışmada öğrencilerin

cinsiyetlerinin programlama başarısına etki ettiği görülürken başka bir çalışmada tam tersi sonuç çıkmıştır. Yapılacak olan bu çalışmada farklı okullardan seçilecek büyük bir örneklem oluşturulmuştur. Böylelikle elde edilen sonuçların evreni temsil etme gücü yükselmiştir.

Yine bu alanda yapılan çalışmalar tek bir ders kapsamında yürütülmüştür. Programlamaya genel bir bakış açısı ile yaklaşılmamıştır. Oysaki programlama birbirinden bağımsız derslerden oluşan bir disiplin değildir. Programlama eğitimi verilen bölümlerin müfredatlarına baktığımızda birbirinin ön koşulu ya da hazırlayıcısı olan birçok dersten oluştuğunu görürüz. Bir dersin amacı olan kazanımlar, başka bir dersin kazanımları için temel oluşturmaktadır. Bu alanda verilen tüm derslerin birbirleriyle doğrudan ilişkileri vardır. Bu çalışmada programcılık başarısı belirlenirken tek bir derse bağlı kalmaksızın, programın genelini kapsayan bir değerlendirme yapılmıştır. Böylelikle elde edilecek sonuçlar tek bir dersin başarı kriterlerini değil, bütün halinde bir programın başarılı olabilmesi için gerekli kriterleri ortaya çıkaracaktır.

1.4 Araştırmanın Varsayımları

1. Tüm MYO bilgisayar programcılığı programlarında, derslerin İkmep kapsamında oluşturulan müfredata göre, uygun laboratuvar ortamlarında yapıldığı varsayılmaktadır.

1.5 Araştırmanın Sınırlılıkları

1. Bu araştırma çalışmaya katılan MYO öğrencileri ile sınırlıdır.
2. Programlama başarısına etki eden faktörler incelenen faktörler ile sınırlıdır.
3. Uygulamada kullanılan ölçeklerin uzun olması öğrencilerin sıkılarak içtenlikle cevap vermesini engellemiş olabilir.

1.6 Tanımlar

Genel Akademik Not Ortalaması (GANO): GANO, öğrencilerin aldıkları derslerin puanları dersin kredisine çarpılarak toplanır ve toplam alınan ders kredisine bölünerek hesaplanır.

2. KAVRAMSAL ÇERÇEVE

2.1 Programlama

Programlama, bir problemin çözümünün programlama dili kullanılarak oluşturulan kod satırlarına verilen isimdir. (Arabacıoğlu ve diğerleri, 2007). Programlamanın bir başka tanımı ise iyi bir şekilde analiz ve tasarımı yapılmış problemin çözümüne dair adımlar ile çözümün oluşturulup bir programlama dili ile bilgisayar ortamına aktarım işidir (Eryılmaz, 2003). Programlamada elimizdeki problemin çözümüne yönelik işlem adımlarını sırasıyla işlemciye yaptıran kod blokları kullanılır (Yaşar, 2011).

Programlama, aşamaları olan ve her bir aşamada ciddi çalışma gerektiren bir süreçtir. Programlamaya yeni başlayan birinden hemen bir program yazması beklenemez. Programlamaya yeni başlayan birinin öncelikle problem çözme becerisine sahip olması ve bir problemin çözüm aşamalarını tasarlayıp, akış diyagramlarını oluşturabilmesi gerekir (Eryılmaz, 2003).

Programlama bu amaç için hazırlanmış özel komut ve sembollerden oluşan programlama dilleri ile yapılır. Farklı bilgisayar ve platformlar için birçok programlama dili oluşturulmuştur (Ersoy, Madran ve Gülbahar, 2011).

Eryılmaz'a (2003) göre geliştirilen ilk bilgisayarla birlikte, bilgisayarla kullanıcı arasındaki iletişimi sağlayacak bir mekanizmaya ihtiyaç duyulmuştur. Önceleri bu iletişimi sağlamak amacıyla makine dili kullanılmıştır. Makine dili kullanımının zor olması programlama dillerinin ortaya çıkmasını sağlamıştır. 1957 yılında Fortran, 1958 yılında Algol geliştirilmiştir. Bu dilleri Cobol, Basic, Pascal, C gibi programlama dilleri takip etmiştir. Günümüzde ise, bu programlama dillerinin daha gelişmiş versiyonları ve nesneye yönelik programlamayı destekleyen C++, Java, C# gibi üst düzey programlama dilleri geliştirilmiş ve programların oluşturulmasında bu dillerin kullanımı ağırlık kazanmıştır.

Gelişimlerine göre programlama dilleri beş seviyede incelenir (Çamoğlu, 2009).

Birinci Nesil Programlama Dilleri: İlk bilgisayarların geliştirilmesiyle ortaya çıkan birinci nesil programlama dilleri, tamamen geliştirildikleri makineye özgü düzenlenmiş makine seviyesi dillerdir. Komutlar direk bilgisayar üzerine yerleştirilmiş bir panel aracılığıyla her hangi bir derleyiciye ya da çeviriciye ihtiyaç duyulmadan yazılıyordu.

Birinci nesil programlama dilleri işlemciye doğrudan verilen komutlarla çalışması nedeniyle yetenek ve hız açısından önemli bir avantaj sağlıyordu. Ancak öğrenilmesi ve kullanılması zordu. Hatta bazı durumlarda dillerle yazılan programları yönetmek sıkıntılı bir iş haline geliyordu. Ayrıca belli bir makine için yazılan program diğer makinelerde çalışmıyordu. Kullanılacak olan programın her makine için ayrı ayrı yazılması gerekiyordu. ABM, Intel 80x86, IBM 360 bu programlama dillerinden bazılarıdır.

İkinci Nesil Programlama Dilleri: Birinci nesil programlama dillerinin öğrenilmesinin ve kullanımının zorluğu insanları daha anlaşılır olan assembly-çevirmen diller olarak adlandırılan programlama dillerini geliştirmeye itti. İkinci nesilde işlemciye doğrudan gönderilen kodların yerini, komutlar ve* kısaltmalar almaya başladı. Fakat bu programlama dillerinde de programlar her makine için özel olarak hazırlanması gerekiyordu. Ayrıca bu dillerde yazılan programların makine dillerine çevrilmesi gerekmektedir. Bu noktada assembler yani çevirmen programlar devreye girerek kodların makine diline çevrilmesini sağlıyordu. Bu dillere örnek olarak TASM (Turbo Assembler), EMU8086, BAL (Basic Assembler) verilebilir.

Üçüncü Nesil Programlama Dilleri: Günümüzde yaygın olarak kullanılan ve bilinen programlama dillerinin geliştirildiği nesil üçüncü nesildir. Üçüncü nesilde programlama dilleri ağırlıklı olarak İngilizce kullanılarak insanların kullandıkları dillere yakın, yüksek seviyede yapısal diller olarak geliştirildiler. Üçüncü nesil programlama dilleri ile yazılan programlar makinelerden bağımsız olarak yazılıyordu. Yazılan programlar farklı bilgisayarlarda da çalışabiliyordu. Bunun içinse programın çalıştırılacağı bilgisayara bir derleyici kurmak gerekir. Bu dillere örnek olarak Fortran, Cobol, C, C++ örnek olarak verilebilir.

Dördüncü Nesil Programlama Dilleri: Üçüncü nesil programlama dilleri programlamayı büyük ölçüde kolaylaştırırsa da ticaret ve iş yaşamında özel durumlara

yönelik hızlı çözümler üretebilmek için dördüncü nesil programlama dilleri geliştirilmiştir. Kullanımı çok daha kolay, daha az kod yazarak yönergeler, hazır şablonlar ve sihirbazlar sayesinde belirli ihtiyaçlarda uzmanlaşmış, pratik çözümler üretmeye yönelik bu diller rapor üretici, form üretici, vaka tasarımı, veri yönetimi, istatistiksel analiz, vb alanlarda uygulamalar geliştirmeye yönelik olarak geliştirilmişlerdir. SQL, ABAP, MATLAB, C# bu dillerden bazılarıdır.

Beşinci Nesil Programlama Dilleri: Beşinci nesil programlama dilleri, programcının algoritma geliştirerek çözüm geliştirmesinin ötesinde, koşulları ve kısıtlamaları bilgisayara vererek, bilgisayarın çözümü kendisinin bulmasına yönelik olarak tasarlanmaktadır. Örnek olarak Prolog, Lisp, OPS5 ve Mercury programlama dillerini verebiliriz.

Birinci nesil programlama dillerinden beşinci nesil programlama dillerine kadar yaklaşık 500 adet programlama dili geliştirilmiştir ve hala geliştirilmeye devam edilmektedir (Sureau, 2001). İnsanlar arasında yaygın olarak yazılım geliştirebilmek için her hangi bir programlama dilini öğrenmenin yeteceği fikri olsa da, program geliştirme kullanılan programlama diline ait söz dizimini kurallarını öğrenmekten çok daha fazlasıdır (Kak, 2008).

Bütün programlama dillerinin kendine has bir söz dizimi olsa da, hepsinde kullanılan ortak temel kavramlar mevcuttur. Programlama dilleri zamanla değişim göstererek, işlemleri kolaylaştıracak yeni komut ve işlevler eklenebilir (Sureau, 2001). Ancak değişmeyecek bir şey varsa o da genel programlama bilgisidir (Eryılmaz, 2003). Programcılık eğitimi verilirken yalnızca kullanılan programlama diline ait söz diziminin öğretilmesi, öğrencilerin o dili öğrenmesini sağlar. Ancak başarılı bir programcı olabilmeleri için temel kavramları çok iyi öğrenmeleri gerekir (Kak, 2008).

2.1.1 Programlamada Temel Kavramlar

Programlama, her programlama diline özgü kurallardan oluşan bir kavram olsa da, her programcının bilmesi gereken bazı temel kavramlar vardır. Bu temel kavramları Eryılmaz (2003) aşağıdaki şekilde sıralamıştır;

Genel programlama bilgisi: Genel programlama bilgisi, problem çözme ve bu problemin aşamalarını belirleyebilme, program tanımını bilme ve algoritma hazırlayabilme kavramlarını içerir.

Problem çözümü: Problem çözümü, bilgisayarda bir problemin çözerken takip edilmesi gereken adımları içerir. Bilgisayarda bir problemi çözerken takip edilmesi gereken adımlar; problemin tanımlanması, çözüm yollarını belirleme ve en uygun çözüm yolunun seçilmesi, seçilen çözüm yolunun algoritmasının belirlenmesi, akış çizelgesinin hazırlanması, programın kodlanması ve uygulamaya alma adımlarından oluşur.

Algoritma: Algoritma, çözümü yapılacak bir problemin tüm işlem adımlarını belirleme işidir. Algoritma hazırlanırken sırası ile yapılacak tüm işlemler belirtilir ve program bu işlem sırasına göre kodlanır.

Değişkenler ve sabitler: Değişken ve sabitler, program hazırlarken kullanılan değerlerin belleğe saklandığı konumları gösteren sembolik işaretlerdir. İki tür gösterim biçimi vardır. Sabit değer atamasında, değer bir kez atanır ve program içerisinde değiştirilmeden kullanılır. Değişken de ise, program içerisinde istenildiği kadar değer ataması yapılabilir.

Atama işlemi: Atama işlemi, bir değişkene değer aktarmak için kullanılır.

Karar verme ve kontrol işlemleri: Karar verme ve kontrol işlemleri, programın işleyişi içerisinde değişkenlerin değerlerini kontrol etmek, programın bu değerlere göre nasıl devam edeceğini kararlaştırmak için kullanılır.

Döngü yapıları: Programın belli bir kısmının birden fazla tekrarlanmasına döngü denir. Belirli durumlarda aynı komutların tekrarlanması gerekiyorsa, komutları tekrar tekrar yazmak yerine, bunların tekrar etmesini sağlayacak yapılar kullanılır. Bunlara döngü yapıları denir. Döngü yapıları 2 çeşittir. Birinci yapıda işlemlerin kaç kere tekrar edeceği önceden belirlidir ve döngü bu tekrar sayısını verecek şekilde düzenlenir. İkincisinde döngünün kaç kez tekrarlanacağı önceden belli değildir. Tekrar sayısı belli bir şarta bağlı olarak belirlenir. Belirlenen şart gerçekleştiği sürece işlemler de tekrarlanır.

Diziler: Diziler, aynı türde değişkenlerin tek bir değişken ismiyle tutulmasını sağlayan yapılardır.

Her programlama dilinde yukarıda belirtilen kavramların söz dizimi farklı olsa da kullanım amacı aynıdır.

2.1.2 Programlama Eğitimi

Programlama, son yıllarda öğrenciler arasında popülaritesi hızla yükselen ve iyi bir kariyer vaat eden bir alan olarak görülmektedir. Ancak öğrenilmesi de bir o kadar zor bir alandır. Programlama dersleri genellikle zor olarak kabul edilir ve öğrencilerin başarı oranları da çok yüksek değildir (Robins, Rountree ve Rountree, 2003). Genellikle programlamaya yeni başlayan birinin uzman haline gelebilmesi için 10 yıllık bir deneyime sahip olması gerektiği söylenir (Winslow, 1996). Peki, iyi bir yazılımcı hangi temel konuları bilmelidir ve iyi bir yazılımcı yetiştirebilmek için nasıl bir eğitim verilmelidir. Son 25 yıl içerisinde teknoloji ve bilgisayar alanında yaşanan gelişmeler, bu soruya cevap bulmayı daha zor hale getirmiştir (Kak, 2008).

Programlama eğitimi vermeye başlamadan önce sormamız gereken bazı sorular bulunmaktadır. Programlamayı öğrenmek neden bu kadar zor? Başarılı olabilmek için sahip olunması gereken bilişsel özellikler nelerdir? Öğrenciler için başarılı olan ve başarısız olan stratejiler nelerdir ve özellikle cevap aramamız gereken soru, iyi bir programcının sahip olması gereken özellikler nelerdir?

İyi bir programcının sahip olması gereken özellikleriyle ilgili bir çok çalışma yapılmıştır. Bu çalışmalar genellikle uzman yazılımcılar üzerinde gerçekleştirilmiştir. Bu çalışmalar, uzman yazılımcıların problem çözme stratejileri üzerinde yoğunlaştıklarını, gelişmiş bilgi birikimlerinin olduğunu ve matematik, satranç gibi alanlarda çok iyi olduklarını göstermektedir (Robins ve diğerleri, 2003). Ancak programlamaya yeni başlayan insanların uzman yazılımcılardaki özelliklerin birçoğuna sahip olmadığı görülmektedir (Winslow, 1996).

Yeni başlayanlar için programlamayı öğrenmek zor bir iştir. Boulay (1989) programlamaya yeni başlayanların bilmesi gereken konuları, programlama nedir ve

programlamayla ne yapabiliriz, programların olmadığı bir bilgisayar modeli hayal etme, belirli bir programlama diline ait söz dizimi ve semantiği, programlama içerisinde geçen kavramlar, planlama becerileri, geliştirme test etme, hata ayıklama olarak açıklamıştır. Programlama eğitimi verilirken bu becerilerin kazandırılması gerekir.

Bazı öğrenciler bu kavramları kendi kendilerine çeşitli kaynaklardan öğrenebilirler. Ancak eğitimciler öğrencilerin çoğunun, programlamayı kendi kendilerine öğrenemeyeceklerini ve öğrencilerin başarılı olabilmeleri için çeşitli stratejiler geliştirmek gerektiğini düşünürler (Norvell ve Bruce-Lockhart, 2004). En azından başlangıç için bir eğitim almak gerekir.

İmal ve Eser'e (2005) göre, programlama eğitimine yeni başlayan öğrencilerin öncelikle öğrenmesi gereken konuların başında algoritma geliştirmek vardır. Algoritma eğitimiyle birlikte öğrencilere anlaşılması kolay, kuralları basit, yüksek seviyeli bir programlama dili de öğretilir. Algoritma geliştirmeyi öğrendikten sonra öğrencilerin, programlamayla ilgili temel kavramları ve program yazma stratejilerinin öğretilmesi gerekir (Ala-Mutka, 2004).

Programlama eğitimi çoğunlukla üst düzey beceri gerektirdiği için öğrenilmesi zor bir süreçtir (Gültekin, 2006). Genellikle öğretmenler bu eğitimi verirken, gerekli bilgileri verdikten sonra öğrencilerin karşılaştıkları problemleri çözmelerini isterler. Ancak öğrenciler şikayetlerini dile getirirken sıklıkla, öğretmen anlatırken anladıklarını, fakat kendi problemlerine çözüm üretemediklerini söylemişlerdir. (Garner, 2003).

Bu durum araştırmacıları farklı yöntemler deneyerek programlama başarısını artırmaya yöneltmiştir. Bu çalışmalardan bir tanesinde Arabacıoğlu ve diğerleri (2007) programlama mantığını öğretebilmek amacıyla bir uygulama dili tasarlamışlardır. Teorik yöntemin sıkıcı ve etkili olmadığını söyleyen araştırmacılar geliştirdikleri materyal ile bu kısıtlayıcıların önüne geçmeyi amaçlamışlardır. Bu çalışmayla araştırmacılar soyut bir kavram olan programlamayı daha etkili hale getirmeye çalışmışlardır. Bunun gibi birçok çalışmada da çeşitli araçlar, yöntemler geliştirilerek programlama eğitimin daha başarılı olması amaçlanmıştır. Ancak Jiau (2009) ve diğerleri bir materyal kullanılarak programlama eğitimi somutlaştırılsa da yine de öğrencilerin bazen başarılı olamadıklarını söylemektedirler. Öyleyse programlama başarısının sadece öğretim yönteminden kaynaklanmadığını söyleyebiliriz. Bu başarıyı

etkileyen diğer faktörler incelenerek bunları geliştirmeye yönelik tedbirler alınması gerekir.

2.2 Programlama Başarısına Etki Eden Demografik Bilgiler

2.2.1 Cinsiyet

Araştırmamız içerisinde ele alacağımız değişkenlerden biri olan cinsiyet faktörü bir çok çalışmada ele alınarak, teknoloji kullanımının cinsiyete göre farklılıklar gösterip göstermediğine ilişkin çeşitli çalışmalar yapılmıştır. Bundan 15 yıl önce teknolojiye erişim günümüzdeki kadar yaygın değildi (NCREL, 2005). O dönemlerde yapılan çalışmalar, kadınların teknolojiye olan ilgisinin erkeklere göre daha düşük olduğunu göstermektedir (Mims, 2012). Bu dönemdeki çalışmalarda erkeklerin kızlara göre teknolojiye daha meraklı ve teknolojiyi kullanmada daha yetenekli oldukları görülmektedir (Hawkins, 1985). Bu çalışmalarda teknolojinin erkeklere daha uygun bir alan olduğu söylenmektedir (Wilder, Mackie ve Cooper, 1985). Bu dönemde erkeklerin ve kızların teknolojiye erişim imkanlarının da farklılık gösterdiği ve erkeklerin teknolojiyi kullanma sürelerinin kızlara göre daha fazla olduğu görülmektedir (Schubert, 1986).

Günümüze geldiğimizde ise teknoloji kullanma sürelerinde hala erkeklerin daha fazla kullandığı, fakat teknoloji kullanımına yönelik tutumlar arasındaki farkın ortadan kalktığı görülmektedir (Ikolo ve Okiy, 2012). Deniz, Görgen ve Şeker (2006) yaptıkları çalışmalarında cinsiyet ile teknolojiye yönelik tutum arasında herhangi bir fark olmadığını belirtmişlerdir. Köse, Gencer ve Gezer (2007) MYO öğrencilerinin internet ve teknoloji kullanımına yönelik tutumlarını inceledikleri çalışmalarında yine cinsiyet açısından bir fark olmadığını göstermişlerdir.

Cinsiyet faktörü, teknolojinin bir alt dalı olan programlama ile arasındaki ilişkilerin incelendiği çalışmalara da konu olmuştur. Ancak bu çalışmalar, teknolojiyle paralellik göstermeyerek, cinsiyet faktörüne göre çeşitli çalışmalarda farklı sonuçlar elde edilmiştir.

Goold ve Rimmer (2000) 2 dönem süren çalışmalarında üniversite öğrencilerinin birinci sınıftaki bilgisayar programlama derslerindeki başarılarını etkileyen faktörleri incelemişler ve cinsiyet faktörünün öğrencilerin birinci dönem başarılarında etkili olurken ikinci dönem başarılarında herhangi bir etkisinin olmadığını göstermişlerdir. Byrne ve Lyons (2001) programlama başarısına etki eden öğrenci faktörlerini inceledikleri çalışmalarında cinsiyet bakımından herhangi bir fark bulamamışlardır. McDowell, Werner, Bullock ve Fernald (2003) cinsiyetin programlama başarısı üzerinde etkisi olmadığını söylemektedirler. Yine Pillay ve Jugoo (2005) yaptıkları araştırmada cinsiyet faktörünün programlama başarısı üzerinde etkisi olmadığını göstermiştir.

Ancak tam aksini gösteren çalışmalar da mevcuttur. Bunlara örnek olarak Lau ve Yuen (2009), Pioro (2004) ve Houston'ın (1993) araştırmalarını örnek gösterebiliriz. Bu araştırmalarda tahmin edilenin aksine kızların erkeklere göre daha başarılı oldukları görülmektedir.

Görülüyor ki, araştırma yapılan ülkeye, öğrenci grubuna göre farklı çalışmalarda farklı sonuçlar alınabilmektedir. Yapılan literatür taramasında benzer bir çalışmanın Türkiye ve MYO örneği bulunamamıştır. Bu yüzden bu çalışmada incelenen faktörler arasına cinsiyet eklenmiştir.

2.2.2 Programlama Deneyimi ve Ön Bilgi

Hazır bulunuşluk düzeyi, öğrencinin yeni bir davranış değişikliği meydana getirebilmesini sağlayan önemli faktörlerden bir tanesidir (Harman ve Çelikler, 2012). Bu yüzden öğrenci yeni öğreneceği bilgiler için gerekli olan ön koşul niteliğindeki bilişsel, duyuşsal ve devinimsel davranışlara sahip olmalıdır (Başar, 2001).

Thorndike hazır bulunuşluk kavramını;

- Birey etkinliği yapmaya hazırsa etkinliği yapması mutluluk verir.
- Birey etkinliği yapmaya hazır; fakat etkinliği yapmasına izin verilmezse bu durum kızgınlığa neden olur.

- Birey etkinliđi yapmaya hazır deđil ve etkinliđi yapmaya zorlanırsa bu durum bireyde kızgınlıđa neden olur (Akt:Senemođlu, 2009) řeklinde aıklamıřtır.

Hazır bulunuřluk dzeyi, đrencinin zel ve genel yeteneklerini, hedef davranıřlarla ilgili bilgi ve becerilerini, ilgi, tutum ve gdlenmiřliđini ierir ve đrencinin zgemiři olarak da ifade edilebilir (Bloom, 1995). Eđitime nerden bařlanacađının tespiti iin, đrencilerin hazır bulunuřluđunun bilinmesi gerekir (Fidan, 1986). Yani eđitime bařlamadan nce đrencilerin n bilgilerini lerek eđitimi đrencilerin n bilgilerine ve hazır bulunuřluklarına gre dzenlemeliyiz.

MYO’larda eđitime bařlayan đrenciler faklı liselerden geldikleri iin bir kısmı programlamayla ilgili bilgi sahibiymen bir kısmı ilk defa karřılařmaktadırlar. đrencilerin hazır bulunuřlukları birbirinden farklıdır. Bu yzden eđitim đrenciler ilk defa đreniyormuř gibi verilir. Bu durumda akla daha nceden programcılık eđitimi almıř ya da deneyim sahibi olan đrencilerin, yani hazır bulunuřluđu yksek olan đrencilerin, diđerlerine gre daha bařarılı olup olmayacakları sorusu gelmektedir. Bu sorudan yola ıkararak deneyim ve n bilginin programlama bařarısı zerindeki etkisi sıklıkla incelenmiřtir.

Literatrde yapılan incelemeler sonucunda deneyim ve n bilginin programlama bařarısı zerinde etkisi incelenen alıřmalarda farklı sonular alınmıřtır. Bazı alıřmalarda đrencilerin programlamayla ilgili n bilgileri ve deneyimlerinin bařarıları zerinde etkisi olduđu sylenirken, bazı alıřmalarda da herhangi bir etkisine rastlanmamıřtır.

Konvalina, Stephens ve Wileman (1983) alıřmalarında gemiř bilgisayar deneyimlerinin đrencilerin bařarılarında etkili olduđu grlmřtr. Ancak Pillay ve Jugoo (2005) yaptıđı alıřmada nceki bilgisayar bilgilerinin etkisi olmadıđını bulmuřtur. Holden ve Weeden (2003) bir ders kapsamında 3 dnem sren alıřmalarında nceki bilgisayar deneyimlerinin dersin ilk dneminde etkili olduđu, diđer dnemlerde herhangi bir etkisinin bulunmadıđı sonucuna ulařmıřlardır. Byrne ve Lyons (2001) alıřmalarında daha nceden programlama deneyimi olan đrencilerin daha bařarılı olduklarını bulmuřtur.

Öğrenciler daha önceden programlama eğitimi almış olsunlar veya olmasınlar, aldıkları eğitim sonrasında hepsinin aynı düzeye çıkması, başlangıçtaki farkın ortadan kalkması beklenir. Bu amaca ulaşp ulaşamadığımızı görmek için programlama deneyimi ve ön bilgi bu çalışmanın değişkenleri arasına eklenmiştir.

2.2.3 Öğrencilerin Mezun Oldukları Okul Türü ve Mezuniyet Puanları

MYO’larda eğitim alan öğrenciler değişik liselerde eğitim alarak gelmektedirler. Temel olarak bu liseler meslek liseleri ve genel liseler olarak ayrılmaktadır.

Meslek liseleri Türk eğitim sisteminin genel amaçlarına uygun, öğrencileri meslek hayatına, yükseköğretim kurumuna hazırlayan dört yıllık eğitim kurumlarıdır. Meslek liselerinde ilk yıl, diğer liselerle aynı öğretim programı uygulanmaktadır. İlk yılın sonunda öğrenciler meslek alanlarını seçerler. İkinci yılda seçtikleri alanla ilgili temel bilgileri öğrenirler. Üçüncü yılda öğrenciler kendi alanlarında yetenekli oldukları dallara yönlendirilir. Son yıl ise öğrenciler alan ve dallarına göre işletmelerde staj yaparlar. Dört yılını tamamlayarak mezun olan öğrencilere lise diplomalarıyla birlikte işyeri açma belgesi verilir. Böylece öğrenciler mesleklerini yapmaya hazır hale gelirler. Bunun yanı sıra öğrencilerin üniversite sınavlarına girerek yüksek öğretime devam etme hakları da bulunur. Ayrıca meslek lisesi Bilişim Teknolojileri bölümü öğrencileri sınava girmeden MYO bilgisayar programcılığı programlarında eğitimlerine devam edebilirler.

Genel liseler Türk Millî Eğitiminin amaçları doğrultusunda, öğrencilere ortaöğretim seviyesinde asgari genel kültür veren ve yüksek öğretime öğrenci hazırlayan öğretim kurumlarıdır. Bu okulların öğretim süresi ilköğretim üzerine dört yıl olup, sınavsız öğrenci alınmaktadır. Genel liseleri bitiren öğrencilere diploma verilir. Öğrenci, isterse yükseköğretim programlarına ilgili sınavlara girerek devam edebilir.

Her iki lise türü mezunu öğrenciler aynı bölümde okuma imkanlarına sahiptirler. Ancak meslek lisesi mezunu öğrenciler alanlarıyla ilgili bilgileri önceden öğrenerek gelmektedirler. Meslek lisesi bilişim teknolojileri bölümü mezunu öğrencileri programlamayla ilgili bir çok ders almaktadırlar. Bu dersler Tablo 4’te gösterilmiştir. Genel lise mezunları ise yalnızca 9. sınıfta seçmeli olarak Bilgi ve İletişim Teknolojileri dersi almaktadırlar. Bu ders kapsamında da programlama eğitimi verilmemektedir.

Tablo 4. Meslek Lisesi Bilişim Teknolojileri Bölümü Haftalık Alan Dersleri Çizelgesi (MEB, 2011)

DERS KATEGORİLERİ	DERSLER	HAFTALIK DERS SAATİ			
		9. SINIF	10. SINIF	11. SINIF	12. SINIF
ALAN ORTAK DERSLERİ	Mesleki Gelişim		2		
	Bilişim Teknolojilerinin Temelleri		4		
	Programlama Temelleri		4		
	Paket Programlar		2		
	Temel Elektronik ve Ölçme		2		
	Bilişim Teknik Resmi		2		
DAL DERSLERİ	İşletmelerde Beceri Eğitimi				
	Açık Kaynak İşletim Sistemi				
	Mesleki Yabancı Dil				
	Nesne Tabanlı Programlama Veritabanı				
	Gelişmiş İnternet Uygulamaları				
	Sistem Bakım Ve Onarım				
	Elektronik Uygulamaları				
	Bilgisayarlı Devre Tasarımı				
	Sistem Kontrol Uygulamaları			18	26
	Sunucu İşletim Sistemi Ve Ağ Güvenliği				
	Web Tasarımı Ve Programlama				
	Grafik Ve Animasyon				
	İnternet Programcılığı				
	Veritabanı Organizasyonu				
	Ağ Sistemleri Ve Yönlendirme				
	Sunucu İşletim Sistemi				
	E-Posta Sunucu				

Meslek lisesinden gelen öğrenciler daha önce programlama eğitimi alırken, genel liselerden gelen öğrenciler herhangi bir programlama eğitimi almadan eğitimlerine başlamaktadırlar. Bu farklılıktan dolayı öğrencilerin programlama derslerinde başarılarının farklılık göstereceği söylenebilir. Gold ve Rimmer (2000) yaptıkları çalışmada, mezun olunan okul türü değişkeni açısından programlama başarısı incelendiğinde bu değişkenin etkisinin sabit olmadığını söylenmiştir.

2.3 Motivasyon ve Öz Yeterlilik Algısı

Motivasyon, kişinin istek ve iradesiyle yaptığı davranışların altında yatan sebepleri açıklayan bir olgudur ve davranışın altında yatan sebepleri açıklar (Lai, 2011). Motivasyon, organizmayı amaçlarına ulaşması, gerekli davranışları yapması için harekete geçiren, duygusal bir yükselmeye neden olan, enerji veren, davranışları yönlendiren itici bir güçtür (Özevin, 2006). İnsanların günlük yaşamlarında gösterdikleri davranışların hızını, sürekliliğini ve şiddetini belirleyen bir çok faktör vardır. Bu faktörler içten (bireyin kişisel özellikleri) ya da dıştan (çevre) gelecek çeşitli etkenlere göre değişmektedir (Akbaba, 2006). Bu etkenler içsel motivasyonu oluşturan, kişinin ilgi, zevk ve hoşuna giden davranışlarla, dışsal motivasyonu oluşturan çevresel faktörlerdir (Lai, 2011). Günlük yaşamın aksine eğitimde öğrencilere kasıtlı davranışlar kazandırmak için bu etkenler kontrol altına alınmalıdır (Akbaba, 2006).

Eğitim sırasında bazı öğrencilerin derse, konuya ya da karşılaşılan probleme çözüm üretmekte istekli oldukları gözlenirken, bazı öğrencilerin ise derslerde isteksiz oldukları, karşılaştıkları problemlerle mücadele etmek yerine daha çok kaçmayı tercih ettikleri gözlemlenmektedir ve öğrenciler arasındaki bu farkın oluşmasında etki eden başlıca etken motivasyon olarak görülmektedir (Akbaba, 2006). Motivasyon, kişinin istekli hale gelerek, harekete geçmesini sağlar (Deci, Koestner ve Ryan, 1999).

Yine motivasyonla ilişkili olan öz yeterlilik algısı, kişinin bir işi başarabilmek için gerekli becerilere sahip olduğu konusundaki inancıdır. Öz yeterlilik insan davranışlarını etkileyen bir faktördür (Kurbanoğlu, 2004). Bandura'ya (1982) göre, başarı için gerekli becerilere sahip olmak tek başına yeterli değildir. Başarılı olabilmek için bu becerileri etkin bir biçimde kullanmak gerekir. Kişinin her hangi bir işi yapabilecek becerisi olduğu halde, kendisine güveni yoksa bu işte başarısız olabilir (Gawith, 1995). Bandura'nın (1982) çalışmaları, kişinin öz yeterlilik algısının sadece davranışlarını değil aynı zamanda motivasyonunu ve başarısını da etkilediğini ortaya çıkarmıştır. İnsanın herhangi bir konuda başarılı olabilmesi için, bu işi başarabileceğine dair inancının olması gerekir. Yapılan araştırmalar sonucunda kişiler, istedikleri sonucu alabileceklerine inandıklarında harekete geçtiklerini, aksi halde davranış sergilemek için isteksiz olduklarını göstermiştir (Kurbanoğlu, 2004).

Programlama eğitimi zor bir süreç olduğu için öğrencilerin motivasyonu her zaman üst seviyede olmamaktadır ve yine karşısına çıkan zor problemler öğrencilerin öz yeterlilik inançlarını azaltarak kendilerini yetersiz hissetmelerine neden olmaktadır. Bu da öğrencilerin başarılarını etkilemektedir.

Programlama başarısıyla motivasyonun ilişkisini araştıran Matthíasdóttir (2006) sınıf ortamının daha heyecan verici, teşvik edici, motivasyonu artırıcı olduğunda öğrencilerin daha başarılı olduğunu belirtmiştir. Jiau, Chen, ve Ssu (2009) çalışmalarında öğrencilerin motivasyonunu artırmak için programlama dersini eğitsel bir oyun çerçevesinde düzenlemişler ve öğrencilerin motivasyonları arttığında daha başarılı olduğunu tespit etmişlerdir.

2.4 Problem Çözme ve Analitik Düşünme Becerisi

Problem kelimesi karmaşık, olumsuz ve çatışma durumlarını ifade eden bir kavram olarak hayat boyu karşılaşılan durumları ifade etmektedir (Yalçın, Tetik ve Açıkgoz, 2010). Problem çözme ise, bir amaca erişmekte karşılaşılan güçlükleri yenme süreci (Yalçın ve diğerleri, 2010) ve bir problemle karşılaşan bireylerin bu olumsuzluk ve çatışma durumlarıyla yaptıkları mücadeledir (Bayraktar, Güngörmüş, Gülbahçe, Şahin ve Bastık, 2011). Heppner ve Peterson (1982) problem çözmeyi, problemlerle başa çıkma kavramıyla açıklamaktadır. Problem çözme doğuştan itibaren öğrenilen ve okul yıllarında geliştirilen bir kavramdır (Miller ve Nunn, 2001).

Yapılan çalışmalarda problem çözme ve analitik düşünme becerisinin programlama başarısını doğrudan etkilediği görülmüştür. Ancak bir çok öğrenci bu beceriden yoksun olarak üniversite eğitimine başlamaktadır (İsmail ve diğerleri, 2010). Henderson (1986) problem çözme ve analitik düşünme becerisinin bilgisayar bilimlerinde başarılı olabilmek için önemli bir faktör olduğunu belirterek öğrencilerin en çok bu noktada sıkıntı yaşadıklarını belirtmiştir. Programlamada başarılı olabilmek için öğrencilerin sistemli bir şekilde çalışarak, öğrendikleri bilgileri analitik düşünme ve problem çözme becerileri üzerine inşa etmeleri gerekir (İsmail ve diğerleri, 2010).

Programlama baştan sona bir problem çözme sürecidir. Bu yüzden problem çözme başarısını etkileyen en önemli ve en çok etkisi incelenen faktörlerden bir tanesidir.

Bu konuyla ilgili yapılan çalışmalarda, Ismail ve diğerleri (2010) Malezya’da yaptığı çalışmasında programlama derslerinde başarısız olan öğrencilerin problem çözme ve analitik düşünme becerilerinin yeterli olmadığı sonucuna ulaşmıştır. Pillay ve Jugoo (2005) çalışmasında programlama başarısını etkileyen faktörlerden bir tanesinin de problem çözme ve analitik düşünme becerisi olduğunu aktarmıştır. Grant (2003) ise çalışmasında duruma farklı bir açıdan bakarak programlama derslerinin öğrencilerin problem çözme yeteneklerini geliştirip geliştirmediğini araştırmış ve programlama derslerinin öğrencilerin problem çözme becerilerini geliştirmediği saptamıştır.

2.5 Öz Düzenleyici Öğrenme Stratejileri

Öğrencilerin ders başarılarını etkileyebilecek bir çok faktör vardır. Bu faktörler aile, okul, arkadaş çevresi ve öğrencinin kendi özellikleri ile ilgili olabilir. Son yıllarda yapılan akademik başarı ile ilgili çalışmalar öğrencilerin kendi öğrenme süreci üzerinde etkin rol oynadığı öz-düzenleme kavramı oluşturmuştur (Üredi ve Üredi, 2005). Akademik başarı ile doğrudan ilgili olduğu düşünülen öz düzenleme pek çok kişi tarafından tanımlanmıştır. Pintrich (2000) tarafından "öğrencilerin, kendi öğrenme hedeflerini belirledikleri, bilişlerini, motivasyonlarını ve davranışlarını düzenlemeye çalıştıkları, hedefleri ve çevrelerindeki bağlamsal özellikler tarafından yönlendirilip, sınırlandırıldıkları, aktif ve yapıcı bir süreç” olarak tanımlanan öz-düzenleme, Risemberg ve Zimmerman (1992) tarafından, “amaçlar belirleme, bu amaçları gerçekleştirmek için stratejiler geliştirme ve bu stratejilerin kazandırdıklarını denetleme” olarak tanımlanmaktadır. Kauffman (2004)’a göre ise öz-düzenleme, “öğrenenin karmaşık öğrenme etkinliklerini kontrol etmeye ve yönetmeye yönelik çabası”dır. Bu tanımlar ışığında öz-düzenleyici öğrenme, kişinin kendisini tanıma işi ve kendi kendine öğrenebilmesi yolunda kullanabildiği her türlü işlem, teknik, taktik ve strateji olarak tanımlanabilir (Çiltaş, 2011).

Öz-düzenleyici öğrenmeye göre akademik olarak başarılı öğrenciler öğrenmede daha aktif bir rol oynamaktadır; yeni öğrendiği bilgiyi daha etkili bir şekilde işlemekte, eski ve yeni bilgiyi ilişkilendirebilmekte, sunulan materyali örgütlemekte ve dönüştürebilmekte, kendileri için amaç belirleyebilmekte, stratejilerini planlayabilmekte ve ihtiyaç duyduklarında yardım isteyebilmektedir. Yani başarılı öğrenciler kendi öğrenme etkinliklerini kendileri düzenleyebilmekte (Bland, 2005), öğrenme ürünlerine dayalı kendi çabalarını düzenli olarak uyarlayabilmektedir (Zimmerman, Bonner ve Kovach, 1996). Akademik başarı yüksek öğrencilerle, düşük öğrenciler arasında güdülenme düzeyinde de fark olduğu belirtilmektedir (Ruban ve Reis, 2006). Ayrıca beklenenin altında başarı gösteren ile düşük başarılı öğrenciler arasında da fark görülmektedir. Krouse ve Krouse (1981)'e göre öğrencilerin beklenenin altında başarı göstermesinin temel nedeni öz-kontrol stratejilerini etkili kullanmamaları ve öz-düzenleme becerilerinin olmamasıdır.

Akademik başarı üzerinde önemli bir rolü olan öz düzenleme bir çok çalışmada ele alınmıştır. Pintrich ve De Groot (1990) çalışmada, öz-düzenlemeyi öğrenme, motive olabilme ve akademik performans arasındaki ilişkileri incelenmiş; öz-yeterlik ve içsel değer bilişsel bağlantı ve performansla pozitif bir şekilde ilişkili olduğunu ortaya çıkarmışlardır. Bunun yanında öz-düzenleme, öz -yeterlik ve test kaygısının performansın en iyi belirleyicileri olarak ortaya çıktığı görülmüştür. Üredi ve Üredi (2005) ilköğretim 8.sınıf öğrencilerinin öz-düzenleme stratejileri ve motivasyonel inançlarının matematik başarısını yordama gücünü inceledikleri çalışmalarında, öz düzenleme stratejileri ve motivasyonel inançların matematik başarısına ilişkin toplam varyansın %30 unu açıkladığı ve en güçlü yordayıcı değişkenin bilişsel strateji kullanımı olduğunu belirlemişlerdir. Haşlaman ve Aşkar (2007) tarafından yapılan araştırmada; öğrencilerin değer verme, dışsal hedefe yönelme, hedef belirleme, yineleme, öz-yansıma, öz-yeterlik algısı, çaba gösterme, başkalarıyla çalışma ve zaman yönetiminden oluşan öz-düzenleyici öğrenme stratejilerinin, başarının %71 ini açıkladığı belirlenmiştir. Alcı ve Altun (2007) matematik dersine yönelik yaptıkları çalışma sonucunda, cinsiyete ve lise sınıf düzeyine göre öğrencilerin öz-düzenleme ve biliş üstü becerilerinde anlamlı farklılıklar olduğu, buna karşın alanlara göre söz konusu becerilere ilişkin bir farklılığın olmadığı ortaya konulmuştur.

Öz düzenleme bir çok farklı alanda öz düzenlemenin başarı üzerinde etkisi incelenmesine rağmen, programlama başarısı üzerine etkisi pek incelenmemiştir. Bu çalışmada öz düzenlemenin programlama başarısı üzerindeki etkisi incelenecektir.

3. YÖNTEM

3.1 Araştırma Modeli

Bu araştırmada ilişkisel tarama yöntemi kullanılarak MYO öğrencilerinin programlama başarılarını etkileyen faktörler incelenmiştir.

Regresyon analizi olarak da isimlendirilen ilişkisel tarama yöntemi, iki ya da daha fazla değişkenden birinin bağımlı diğerinin bağımsız değişken olarak ayrılarak aralarındaki ilişkiyi matematiksel eşitliklerle açıklanması sürecidir. İlişkisel tarama modelleri geçmişteki ya da mevcut durumu var olduğu şekilde açıklamayı amaçlar (Karasar, 1999). İlişkisel tarama yöntemi kullanılan bağımlı ve bağımsız değişkene göre üçe ayrılır.

a) Basit Regresyon Analizi: İncelenen bağımlı bağımsız değişkenin birer tane olduğu araştırma yöntemidir.

b) Çoklu Regresyon Analizi: İncelenen bağımlı değişkenin bir tane, bağımsız değişkenin iki ve daha fazla olduğu araştırma yöntemidir.

c) Çok Değişkenli Regresyon Analizi: Bağımlı değişkenin iki ya da daha fazla olduğu araştırma yöntemidir.

Bu çalışmada incelenen bağımlı değişken (öğrencilerin programlama dersindeki akademik başarıları) bir tane, bağımsız değişkenler(demografik bilgiler, motivasyon, öz yeterlilik, öz düzenleyici öğrenme stratejileri) birden fazla olduğu için çoklu regresyon analizi kullanılmıştır.

3.2 Evren ve Örneklem

Araştırmada çalışma grubu olarak, MYO Bilgisayar Programcılığı Programı öğrencileri hedef alınmıştır. Bu çalışmaya Bozok Üniversitesi Akdağmadeni MYO, Adnan Menderes Üniversitesi Aydın MYO, Ahi Evran Üniversitesi Kaman MYO bilgisayar programcılığı programı 2. sınıf öğrencilerinden toplam 195 öğrenci katılmıştır. Öğrencilerin okullara ve cinsiyetlere göre dağılımları Tablo 5'de verilmiştir.

Tablo 5. Araştırmaya Katılan Öğrencilerin Okul ve Cinsiyete Göre Dağılımları

Okul	Bölüm	Sınıf	Erkek	Kız	Toplam
Akdağmadeni MYO	Bilgisayar Programcılığı	2	26	10	36
Aydın MYO	Bilgisayar Programcılığı	2	18	9	27
Kaman MYO	Bilgisayar Programcılığı	2	88	44	132

3.3 Veri Toplama Araçları

Çalışmada verileri toplamak amacıyla;

1. Yaş, cinsiyet, mezun oldukları okul ve mezuniyet puanları, programlamayla ilgili geçmiş deneyimleri bilgilerini öğrenmek için demografik bilgi anketi,
2. Öğrencilerin problem çözme becerilerini ölçmek için Taylan (1990) tarafından Türkçe'ye uyarlanan Heppner'in (1982) problem çözme envanteri,
3. Öğrencilerin öz düzenleyici öğrenme stratejilerini ölçmek için Haşlaman tarafından hazırlanan öz düzenleyici öğrenme stratejileri ölçeği,
4. Öğrencilerin programlama alanında öz yeterliliklerini ölçmek için Aypay (2010) tarafından Türkçe'ye uyarlanan Schwarzer ve Jerusalem'in (1995) genel öz yeterlilik ölçeği,
5. Öğrencilerin programlama alanındaki motivasyonlarını ölçmek için Büyüköztürk, Akgün, Özkahveci ve Demirel (2004) tarafından Türkçe'ye uyarlanan Pintrich ve arkadaşları tarafından geliştirilen (1991) güdülenme ve öğrenme stratejileri ölçeği (MSLQ),
6. Programlama başarısını ölçmek için programlama başarı sınavı uygulanmıştır.

3.3.1 Demografik bilgiler anketi

Araştırmada öğrencilerin yaşlarını, cinsiyetlerini, mezun oldukları okul türünü ve mezuniyet puanlarını, programlamayla ilgili geçmiş deneyimlerini, GANO bilgilerini öğrenmek amacıyla demografik bilgi anketi hazırlanmıştır. GANO bilgisinin doğrudan öğrencilere sorulmasının, öğrencilerin doğru cevap vereceği konusunda tereddütler olmasına karşın, meta analiz çalışmaları öğrencilerin kendi bildirdikleri not ortalamalarının araştırmalar için kabul edilebilir olduğunu göstermektedir (Cassady, 2001; Kuncel, Crede ve Thomas, 2005, Akt:Delialioğlu, Öncü, Çakır, 2008).

3.3.2 Heppner'in problem çözme envanteri

Heppner ve Pearson (1982; Akt: Akay, 2006, s 113) tarafından geliştirilen problem çözme envanteri 35 maddeden oluşan insanların karşılaştıkları problemlere nasıl tepkide bulunduklarını ve nasıl davrandıklarını tanımlayan likert tipi bir ölçektir. Ölçekte her bir madde için 1) Hiç katılmıyorum, 2) Kısmen katılıyorum, 3) Çok az katılmıyorum, 4) Çok az katılıyorum, 5) Kısmen katılıyorum, 6) Tamamen katılıyorum seçenekleri sunulmuştur. Ölçek uygulanırken katılımcıların tüm soruları kendilerine en yakın gelen seçeneğe göre işaretlemeleri istenmiştir.

Ölçeğin Türkçeye uyarlanma çalışmaları Taylan(1990) tarafından yapılmıştır. Ölçek analiz edilirken 9, 22, 29 numaralı maddeler değerlendirmeye alınmaz. Puanlama diğer 32 madde üzerinden yapılır. 1, 2, 3, 4, 11,13, 14, 15, 17, 21, 25, 26, 30 ve 34 numaralı maddeler olumsuz ifade edildiğinden ters olarak puanlanır. Bu maddeler puanlanırken 1=6, 5=2, 4=3, 2=5, ve 6=1 şeklinde tersine çevrilerek kullanılır. Ölçekteki maddelerin yeterli problem çözme becerilerini temsil ettiği varsayılır. Ölçekten 32 - 192 arasında puan alınabilir. Değerlendirme yapılırken düşük puanlar problem çözmede başarılı olduğunu, yüksek puanlar ise problem karşısında etkili çözümler bulamamayı gösterir. Ölçeğin güvenilirliğinin hesaplanmasında alfa iç tutarlılık katsayısı kullanılmış ve ölçeğin alfa iç tutarlılık katsayısı 0.84 olarak hesaplanmıştır. Ölçeğin Türkçeye uyarlanması sonrasında geçerlik çalışmasında; ölçüt bağıntılı geçerlik yöntemi kullanılmış ve ölçeğin toplam

puanı ile Beck Depresyon Envanteri toplam puanları arasındaki korelasyon -0,33 ve STAI-T toplam puanları arasındaki korelasyon katsayısı ise -0,45 olarak hesaplanmıştır.

3.3.3 Öz düzenleyici öğrenme stratejileri ölçeği

Öz düzenleyici öğrenme stratejileri ölçeği Haşlaman(2005) tarafından Pintrich ve arkadaşları tarafından geliştirilen MSLQ (Motivated Strategies for Learning Questionnaire) ölçeğinden yararlanılarak geliştirilmiştir. Ölçek 15 bölüm ve 77 sorudan oluşmaktadır. Ölçeğin bölümleri, hedef belirleme (2 madde) , stratejik planlama (5 madde), içsel hedefe yönelme (4 madde), dışsal hedefe yönelme (4 madde), öğrenme ve performansla ilgili öz yeterlik (9 madde), görev değeri (5 madde), yineleme (4 madde), açıklama (4 madde), düzenleme (4 madde), emek yönetimi, çaba harcama (4 madde), akran işbirliği (3 madde), yardım arama (4 madde), üst bilişsel (9 madde), zaman ve çalışma ortamı (7 madde) ve öz yansıma (9 madde) dan oluşmuştur.

Ankette bulunan 77 maddeden araştırmamızla ilgili olan hedef belirleme (2 madde), içsel hedefe yönelme (4 madde), dışsal hedefe yönelme (4 madde), öğrenme ve performansla ilgili öz yeterlik (9 madde), görev değeri (5 madde), öz yansıma (9 madde) ve üst biliş (9 madde) ile ilgili toplam 43 madde kullanılmıştır.

Katılımcılar ölçekte yer alan maddeler katılma düzeylerini, beni hiç yansıtmıyor (1), beni tamamen yansıtıyor (10) arasında değişen likert tipi onlu derecelendirme ölçeği üzerinde işaretleyerek belirtmişlerdir. Ölçeğin uygulanması yirmi ile otuz dakika arasında değişmektedir. Toplam 77 maddeden oluşan ölçeğin güvenilirlik ve geçerlilik çalışmaları Hacettepe Üniversitesi Meslek Yüksek Okulu Bilgisayar Programlama Bölümü birinci ve ikinci sınıf öğrencilerinden elde edilen veriler üzerinde yapılmıştır (n=100). Testin güvenilirliğinin ölçümünde kullanılan alfa iç tutarlılık katsayısı 0.95 olarak hesaplanmıştır. Ölçeğin geçerliliği faktöryel (yapı) geçerliği, doğrulayıcı faktör analizi ile incelenmiştir. Analiz sonucu ölçeğin faktör yapısına ilişkin önerilen modelin geçerli olduğunu göstermiştir.

3.3.4 Öz yeterlilik ölçeği

Çalışmada öğrencilerin öz yeterlilikleri Aypay (2010) tarafından Türkçe'ye çevrilen Schwarzer ve Jerusalem'in (1995) genel öz yeterlilik ölçeği kullanılarak belirlenmiştir. Aypay ölçeğin Türkçe versiyonunu geliştirirken 693 üniversite öğrencisi üzerinde çalışmıştır. Çeviri-tekrar çeviri tekniği ile Türkçe'ye çevirisi yapılan ölçeğin yapı geçerliği için Temel Bileşenler Analizi ve Varimax döndürme teknikleri kullanılmıştır. Bileşen analizi sonucu, öz değerleri 1'den büyük iki bileşenli bir yapı ortaya çıkmıştır. Ölçek bileşenleri için alfa iç tutarlılık katsayıları 0.79 ve 0.63'tür. Toplamda hesaplanan Alfa katsayısı .83'tür. Ölçeğin test-tekrar test güvenirlik katsayısı ($r=.80$, $p<.001$)'dir. Sonuçlar ölçeğin Türkçe versiyonunun geçerli ve güvenilir olduğunu göstermektedir. Ölçeğin uygulamadan elde edilen alfa iç tutarlılık katsayısı 0.87 olarak hesaplanmıştır.

Ölçek 10 maddeden oluşan likert tipi bir ölçektir. Ölçekte her madde için 1) doğru değil, 2) biraz doğru, 3) kararsızım, 4) daha doğru, 5) tümüyle doğru seçenekleri sunulmuştur. Ölçek uygulanırken katılımcıların tüm sorulara kendilerine en yakın gelen seçeneği işaretlemeleri istenmiştir.

3.3.5 Güdülenme ve öğrenme stratejileri ölçeği (MSLQ)

Çalışmada öğrencilerin programlama derslerine karşı olan motivasyonları Büyüköztürk ve arkadaşları (2004) tarafından Türkçe'ye uyarlanan Pintrich ve arkadaşları tarafından geliştirilen (1991) güdülenme ve öğrenme stratejileri ölçeği (MSLQ) ile ölçülmüştür. Ölçek değer, içsel motivasyon, öz yeterlilik ve derin strateji düzeyleri olmak üzere 4 bölümden oluşmaktadır. Çalışmada ölçeğin içsel motivasyonla alakalı 7 maddesi kullanılmıştır. Ölçekte her madde için 1) doğru değil, 2) biraz doğru, 3) kararsızım, 4) daha doğru, 5) tümüyle doğru seçenekleri sunulmuştur. Ölçek uyarlanırken katılımcıların tüm sorulara kendilerine en yakın gelen seçeneği işaretlemeleri istenmiştir. Ölçeğin uygulamadan elde edilen alfa iç tutarlılık katsayısı 0.85 olarak hesaplanmıştır. Ölçeğin yapı geçerliklerini incelemek için Açıklayıcı ve

Doğrulayıcı Faktör Analizi yöntemleri kullanılmış ve ölçeğin geçerli olduğu tespit edilmiştir.

3.3.6 Başarı sınavı

Öğrencilerin programlama başarısını ölçmek için bir başarı testi kullanılmıştır. Başarı testi hazırlanırken Tablo 6' daki belirtke tablosundan faydalanılmıştır.

Tablo 6. Başarı Testine İlişkin Belirtke Tablosu

Ölçülmek istenen davranış	Sorular
Programlamayla ilgili temel kavramların tanımı yapar	1
Hazır bir programı farklı yöntemler kullanarak tekrar yazar	2.a , 2.b
Değişken tanımlaması yapar	2a, 3, 4
Değişkenlere değer aktarır	2a, 3, 4
Değişkende saklanan değerleri okur	3,4,5
Karar yapılarını kullanarak program yazar	2.b, 3,4
Döngü yapılarını kullanarak program yazar	2.a, 3,4
Dizi türü değişkenleri kullanır	3
Verilen bir probleme uygun programı yazar	3,4
Hazırlanmış bir programın ekran çıktısını yazar	5

Öğrencilerin programlama başarısını ölçmek için hazırlanan ölçek 100 tam puan üzerinden değerlendirilmiştir. Ölçeğin hazırlanmasında Tukiainen ve Mönkkönen (2002) tarafından oluşturulan programlama başarı sınavından yararlanılmıştır. Sınav soruları beş bölümden oluşmaktadır.

Birinci bölümde temel programlama kavramları ile ilgili 5 tanım sorularak öğrencilerin cevaplaması istenmiştir. Bu tanımlar tüm programlama dillerinde ortak olarak kullanılan, programlama, değişken, döngü, karar yapısı, dizi kavramlarını içermektedir.

İkinci bölümde do – while döngü yapısı kullanılarak hazırlanmış olan program parçasının for döngü yapısı kullanılarak ve switch – case yapısı kullanılarak hazırlanmış kod parçasının if karar yapısı kullanılarak tekrar yazılması istenmiştir. Bu soruda öğrencilerin hazır bir uygulamayı alternatif yollarla tekrar çözme becerisi ölçülmeye çalışılmıştır.

Üçüncü bölümde integer türündeki 15 elemanlı bir diziye rastgele sayılar atayarak bu sayılardan en büyüğü ile en küçüğü arasındaki farkı kullanıcıya bildirecek bir programın yazılması istenmiştir. Bu bölümde öğrencilerin program yazabilme becerileri ölçülmeye çalışılmıştır.

Dördüncü bölümde yine öğrencilerin program yazma becerilerini ölçmek amacıyla, öğrencilerden bir program yazması istenmiştir. Bu programda kullanıcıdan 20 tane sayı alınarak bunlardan kaç tanesinin 50’den büyük, kaç tanesinin 50’ye eşit ve kaç tanesinin 50’den küçük olduğunu ve girilen sayıların ortalamalarının kaç olduğu sorulmuştur.

Beşinci bölümde hazır bir program verilerek, program çalıştırıldığında kullanılan değişkenlerin başlangıçtaki değerlerine göre program çalıştırıldığında alacakları son değerleri bulmaları istenmiştir.

Her bir bölüm 20 puan üzerinden değerlendirilerek öğrencilerin başarı puanı hesaplanmıştır. Sınavın tamamlanması için öğrencilere 45 dakikalık süre verilmiştir. Sınavda sorulacak sorularda ve alınacak cevaplarda kullanılacak programlama dili her okulda farklı programlama dilleri öğretilebileceğinden belli bir programlama dili ile sınırlandırılmamıştır. Öğrenciler soruları istedikleri her hangi bir programlama dili ile cevaplandırmışlardır. Bu yüzden sınavın 2 ve 5 numaralı bölümlerinde öğrencilere verilmiş olan kod parçaları çalışmanın uygulanacağı okuldaki öğrencilerin bildikleri programlama dilleri öğrenilerek alternatif programlama dilleri kullanılarak hazırlanmıştır.

Başarı sınavı hazırlanırken sorulacak soruları belirleme aşamasında uzman görüşleri alınarak hazırlanan belirtke tablosu kullanılmıştır. Belirtke tablosuna göre hazırlanan başarı ölçeği daha sonra 3 uzmanın görüşünü sunulmuştur. Görüşü alınan uzmanlar programlama konusunda deneyimli olup, hali hazırda üniversitede

programlama dersleri vermektedirler. Uzmanlardan alınan dönütlere göre başarı testi tekrar düzenlemiş ve son hali verilmiştir.

Başarı testi açık uçlu sorulardan oluştuğu için güvenilirlik puanlayıcılar arası tutarlılık olarak hesaplanmıştır. Puanlayıcılar arası tutarlılığı hesaplayabilmek için, toplanan ölçekler arasından t-testi uygulanabilmesi için yeterli olan rastgele 30 tanesi seçilerek, iki uzman tarafından Ek2 de bulunan başarı test değerlendirme ölçeğine göre ayrı ayrı değerlendirilmiş, uzmanların vermiş oldukları puanların arasındaki korelasyon ilişkili örneklemeler için t-testi ile incelenmiştir. Yapılan inceleme sonucunda ortaya çıkan veriler şu şekildedir.

Başarı Testi Sonuçları İçin Puanlayıcıların Verdiği Puanların Güvenilirliği:

Uygulanan başarı testleri içerisinde rastgele seçilen 30 teste puanlayıcılar tarafından verilen puanlara ait istatistikler tablo 7’de verilmiştir.

Tablo 7. Uygulanan Başarı Sınavları İçerisinden Rastgele Seçilen 30 Testin Sonuçları İçin Puanlayıcıların Verdikleri Puanların Güvenilirliği

Puanlayıcı	N	Ortalama	SS	Sd	T	p
Puanlayıcı 1	30	21.26	15.27	58	0.143	0.886
Puanlayıcı 2	30	20.70	15.33			

* $p < 0.05$

Tablo 7’de başarı sınavları içerisinde rastgele seçilen 30 testin sonuçları için puanlayıcıların verdikleri puanların güvenilirliğine ait t-testi sonuçları yer almaktadır.

Buna göre; birinci puanlayıcının verdiği puanların ortalaması ($X=21.26$, $ss = 15.27$), ikinci puanlayıcının verdiği puanların ortalamasından ($X=20.70$, $ss = 15.33$) yüksektir. Ancak ortalamalar arasındaki bu fark anlamlı değildir, $t(58)=0.143$, $p>0.05$. Yani her iki puanlayıcın verdiği puanlar arasında bir fark yoktur.

3.4 Verilerin Toplanması

Verilerin toplanması sürecinde uygulanmış olan anketler ve başarı sınavı bir kitapçık haline getirilerek öğrencilere sunulmuş ve öğrencilerden sırası ile, demografik bilgiler anketini, Taylan (1990) tarafından Türkçe'ye uyarlanan Heppner'in (1982) problem çözme envanterini, Haşlaman (2005) tarafından hazırlanan öz düzenleyici öğrenme stratejileri ölçeğini, Aypay (2010) tarafından Türkçe'ye uyarlanan Schwarzer ve Jerusalem'in (1995) genel öz yeterlilik ölçeğini doldurmuş ve son olarak programlama başarısı ölçme testini yapmışlardır.

Araştırmada kullanılan ölçeklerin tamamı tek seferde uygulanan okuldaki öğretim görevlileri tarafından uygulanmıştır. Araştırmaya katılan öğrenciler Bilgisayar Programcılığı 2. sınıf 3. dönem öğrencilerinden oluşmuştur. Öğrenciler 3 dönemlik öğrenimleri sürecinde programlama eğitimiyle alakalı, algoritma ve programlamaya giriş, programlama temelleri 1, programlama temelleri 2, görsel programlama 1 derslerini almıştır. Öğrenciler başarı testini bu derslerde öğrendikleri bilgilere göre cevaplamışlardır. Öğrencilerin genel akademik not ortalaması (GANO) uygulama gerçekleştirildiğinde 3. dönem tamamlanmamış olduğu için geçmiş 2 dönemin ortalamasına göre tespit edilmiştir.

Araştırma sonuçlarının sağlıklı olarak elde edilebilmesi için ölçekler doldurulmadan önce çalışmanın amacı anlatılarak öğrencilerin samimi bir şekilde doldurmaları istenmiştir. Başarı sınavı uygulanırken öğrencilere ders başarısı ölçme araçlarından biri olduğu söylenerek uygulanmıştır. Böylelikle öğrencilerin sorulara içtenlikle cevap vermeleri sağlanmıştır.

Araştırmada kullanılan anket ve ölçekler 4 ayrı meslek yüksek okuluna gönderilmiştir. Toplamda 250 adet gönderilen anketlerden 229 tane geri dönüş olmuştur. Geri dönen anketlerin bir kısmında eksik veri olduğu için kullanılamamıştır. Toplamda kullanılabilen anket sayısı 195 olmuştur. Gönderilen anketlerden toplamda %78 geri dönüş alınmıştır.

3.5 Verilerin Analizi

Araştırmanın bağımlı değişkeni programlama başarısı, bağımsız değişkenleri ise cinsiyet, mezun oldukları okul türü, deneyim, GANO, motivasyon, öz yeterlilik algısı, öz düzenleyici faktörler, problem çözme ve analitik düşünme becerisi olarak belirlenmiştir.

Araştırmaya katılan öğrencilerin programlama başarısı ortalama, standart sapma gibi betimsel istatistik yöntemleri kullanılarak hesaplanmıştır. Demografik bilgilerden elde edilen bağımsız değişkenlere göre (cinsiyet, mezun olunan okul türü, mezuniyet notu, deneyim, GANO) öğrenciler arasında anlamlı bir farkın olup olmadığı Çok Faktörlü Faktör Analizi uygulanarak tespit edilmiştir.

Programlama başarısı ve diğer bağımsız değişkenler arasındaki ilişkinin tespit edilmesi için çoklu regresyon analizi tekniği kullanılmıştır.

Uygulanacak olan tüm istatistiksel yöntemlerde istatistiksel analiz paket programı kullanılmıştır. Bu çalışmada kullanılan tüm istatistiksel yöntemlerde anlamlılık düzeyi .05 olarak alınmıştır.

4. BULGULAR VE YORUM

Çalışmanın bu bölümünde uygulanan ölçme araçlarından elde edilen verilere ilişkin bulgu ve yorumlara yer verilmiştir.

4.1 Çalışmaya Katılan Öğrencilerin Demografik Verilerine İlişkin Bulgular

Çalışmada öğrencilerin programlama başarıları incelenirken, hangi değişkenlerin bu başarıyı etkilediğini görebilmek için demografik bilgiler anketi uygulanmıştır. Öğrencilerin vermiş olduğu cevaplara ilişkin sonuçlar frekans ve yüzde değerleri olarak aşağıdaki tablolarda verilmiştir. Frekans değeri her cevaba karşılık gelen öğrenci sayısını, yüzde kısmı ise bu öğrencilerin toplam öğrenci sayısına oranını göstermektedir.

Tablo 8. Çalışmaya Katılan Öğrencilerin Cinsiyete Göre Dağılımları

Cinsiyet	Frekans(f)	Yüzde (%)
Kız	63	32.3
Erkek	132	67.7
Toplam	195	100

Tablo 8 incelendiğinde çalışmaya katılan kız öğrencilerin sayısının 63 (%32.3), erkek öğrencilerin sayısının 132 (%67.7) olduğu görülmektedir. Çalışmaya katılan erkek öğrenci sayısı kız öğrenci sayısından daha fazladır.

Tablo 9. Çalışmaya Katılan Öğrencilerin Deneyimlerine Göre Dağılımları

Deneyim	Frekans(f)	Yüzde (%)
Deneyimi Olan	44	22.6
Deneyimi Olmayan	151	77.4
Toplam	195	100

Tablo 9 incelendiğinde araştırmaya katılan öğrencilerden daha önce herhangi bir programlama deneyimi olan öğrencilerin sayısının 44 (%22.6), programlama deneyimi olmayan öğrencilerin sayısının ise 151 (%77.4) olduğu görülmektedir.

Tablo 10. Çalışmaya Katılan Öğrencilerin Mezun Oldukları Lise Türüne Göre Dağılımları

Lise Türü	Frekans(f)	Yüzde (%)
Genel Lise	23	11.8
Meslek Lisesi	113	57.9
Teknik Lise	59	30.3
Toplam	195	100

Tablo 10 incelendiğinde araştırmaya katılan öğrencilerden genel lise mezunlarının sayısının 23 (%11.8), meslek lisesi mezunlarının sayısının 113 (%57.9) ve teknik lise mezunlarının sayısının 59 (%30.3) olduğu görülmektedir.

Tablo 11. Çalışmaya Katılan Öğrencilerin GANO'ya Göre Dağılımları

Ortalama	Frekans(f)	Yüzde (%)
1 – 1.99 Arası	8	4.1
2 – 2.99 Arası	112	57.4
3 – 4 arası	75	38.5
Toplam	195	100

Tablo 11 incelendiğinde araştırmaya katılan öğrencilerden GANO'su 1 – 1.99 arasında olan öğrencilerin sayısının 8 (%4.1), 2 – 2.99 arasında olan öğrencilerin sayısının 112 (%57.4), 3 – 4 arasında olanların sayılarının 75 (%38.5) olduğu görülmektedir.

4.2 MYO Öğrencilerinin Programlama Başarılarının Çalışmada Ele Alınan Bağımsız Değişkenlere Göre İncelenmesi

Çalışmaya katılan tüm öğrencilerin istatistiki bilgileri Tablo 12'de verilmiştir. Tabloda da görülebileceği gibi araştırmaya katılan toplam öğrenci sayısı 195'dir. Öğrencilerin yapılan başarı sınavından elde ettikleri puanların ortalaması 15.83, standart sapması ise 12.636 olarak bulunmuştur.

Tablo 12. Çalışmaya Katılan Öğrencilere Ait Bilgiler

	N	Ortalama	SS
Toplam	195	15.83	12.636

Öğrencilerin programlama başarıları, cinsiyetlerine göre analiz edilerek sonuçları Tablo 13'de verilmiştir.

Tablo 13. Programlama Başarısının Cinsiyete Göre Bağımsız Grup T-Testi İle Karşılaştırılması

Cinsiyet	N	Ortalama	SS	sd	T	p
Erkek	132	14.55	12.654	193	2.073	0.039*
Kız	63	18.52	12.263			

* $p < 0.05$

Tablo 13'te öğrencilerin programlama başarılarının cinsiyetlerine göre bağımsız gruplar T testi ile karşılaştırılmasına ait sonuçlar yer almaktadır.

Buna göre; kız öğrencilerin programlama başarılarının ortalaması ($X=18.52$, $ss=12.263$), erkek öğrencilerin programlama başarılarının ortalamasından ($X=14.55$, $ss=12.654$) yüksektir. Ortalamalar arasındaki bu fark anlamlıdır, $t(193)=2.073$, $p < 0.05$. Yani cinsiyete göre programlama başarısına baktığımızda kız öğrencilerin daha başarılı olduğu görülmektedir.

Öğrencilerin programlama başarıları, deneyimlerine göre analiz edilerek sonuçları Tablo 14'te verilmiştir.

Tablo 14. Programlama Başarısının Deneyime Göre Bağımsız Grup T-Testi İle Karşılaştırılması

Deneyim	N	Ortalama	SS	sd	T	p
Var	44	18.23	13.411	193	1.434	0.153
Yok	151	15.13	12.360			

* $p < 0.05$

Tablo 14’te öğrencilerin programlama başarılarının deneyimlerine göre bağımsız gruplar T testi ile karşılaştırılmasına ait sonuçlar yer almaktadır.

Buna göre; daha önce programlama deneyimi olan öğrencilerin programlama başarılarının ortalaması ($X=18.23$, $ss=13.411$), programlama deneyimi olmayan öğrencilerin programlama başarılarının ortalamasından ($X=15.13$, $ss=12.360$) yüksektir. Ancak ortalamalar arasındaki bu fark anlamlı değildir, $t(193)=1.434$, $p>0.05$. Daha önce programlama deneyimi olan öğrencilerle, deneyimi olmayan öğrencilerin başarıları arasında herhangi bir fark yoktur.

Öğrencilerin programlama başarıları, genel akademik not ortalamasına (GANO) göre analiz edilerek sonuçları Tablo 15’te verilmiştir.

Tablo 15. Programlama Başarısının GANO’ya Göre Bağımsız Grup T-Testi İle Karşılaştırılması

GANO	N	Ortalama	SS	sd	T	p
1-2,99 Arası	120	13.45	9.459	193	3.418	0.001
3-4 Arası	75	19.64	15.837			

* $p < 0.05$

Tablo 15’te öğrencilerin programlama başarılarının GANO’ya göre bağımsız gruplar T-testi ile karşılaştırılmasına ait sonuçlar yer almaktadır.

Buna göre; GANO’su 3 - 4 arasında olan öğrencilerin programlama başarıları ($X=19.64$, $ss=15.837$), GANO’su 1 – 2.99 arasında olan öğrencilerin programlama başarılarının ortalamasından ($X=13.45$, $ss=9.75$), 2 – 2.99 yüksektir. Ortalamalar arasındaki bu fark anlamlıdır, $t(193)=3.418$, $p<0.05$. Yani GANO’ya göre programlama

başarısına baktığımızda GANO'su 3 - 4 arasında olan öğrencilerin daha başarılı olduğu görülmektedir.

Öğrencilerin programlama başarıları, mezun oldukları lise türüne göre analiz edilerek sonuçları Tablo 16’de verilmiştir.

Tablo 16. Programlama Başarısının Mezun Olunan Lise Türüne Göre Ortalamaları ve Standart Sapmaları

Lise Türü	N	Ortalama	SS
Genel Lise	23	12.96	9.791
Meslek Lisesi	113	14.14	10.401
Teknik Lise	59	20.19	16.143
Toplam	195	15.83	12.636

Tablo 16’ya göre; genel liseden mezun olan öğrencilerin programlama başarılarının ortalaması 12.96, meslek lisesinden mezun olan öğrencilerin programlama başarılarının ortalaması 14.14, teknik liseden mezun olan öğrencilerin programlama başarılarının ortalaması 15.83 olarak belirlenmiştir.

Öğrencilerin programlama başarıları, mezun oldukları lise türüne göre tek yönlü varyans analizi ile analiz edilerek sonuçları Tablo 17’de verilmiştir

Tablo 17. Programlama Başarısının Öğrencilerin Mezun Oldukları Lise Türüne Göre Tek Yönlü Varyans Analizi İle Karşılaştırılması

Kaynak	Kareler Toplamı	Sd	Kareler Ortalaması	F	p
Gruplar Arası	1631.775	2	815.888	5.338	0.006*
Gruplar İçi	29343.640	192	152.831		
Toplam	30975.415	194			

* p<0.05

Tablo 17'e göre öğrencilerin mezun oldukları lise türüne göre başarıları arasında anlamlı bir fark bulunmuştur ($F=5.338$, $p<0.05$). Anlamlı farkın hangi gruplar arasında olduğunu tespit edilmek üzere Tukey testi yapılmıştır. Tukey testi sonuçlarına göre yalnızca teknik lise mezunu öğrencilerle, meslek lisesi mezunu öğrenciler arasında teknik lise mezunları lehinde ($X_{\text{Teknik Lise}}=20.19$, $X_{\text{Meslek Lisesi}}=14.14$) anlamlı fark saptanmıştır.

4.3 MYO Öğrencilerinin Programlama Başarıları İle Demografik Bilgileri (Cinsiyet, Mezun Oldukları Okul Türü, Programlamayla İlgili Geçmiş Deneyimleri, Genel Akademik Not Ortalamaları), Problem Çözme Ve Analitik Düşünme Becerisi, Kişisel Öz Yeterlilik, Motivasyonları Arasındaki İlişkiye Ait Bulgular

MYO öğrencilerinin programlama başarıları ile demografik bilgileri (cinsiyet, mezun oldukları okul türü, programlamayla ilgili geçmiş deneyimleri, genel akademik not ortalamaları), problem çözme ve analitik düşünme becerisi, kişisel öz yeterlilik, motivasyonları kullanılarak yapılan regresyon analizi sonucunda aşağıdaki tablo elde edilmiştir.

Tablo 18 MYO Öğrencilerinin Programlama Başarıları İle Demografik Bilgileri, Problem Çözme ve Analitik Düşünme Becerisi, Kişisel Öz Yeterlilik , Motivasyonları Arasındaki İlişkiye Ait Regresyon Analizi Sonuçları

Değişken	B	Standart Hata	Beta (β)	t	p
Sabit	-14.110	9.080		-1.554	0.122
Cinsiyet	3.889	1.794	0.144	2.168	0.031*
Deneyim	-2.129	2.007	-0.071	-1.061	0.290
GANO	5.599	1.715	0.226	3.265	0.001*
Genel Lise	5.199	2.874	-0.133	-1.809	0.072
Meslek Lisesi	-5.199	1.908	-0.168	-2.250	0.026*
Problem Çözme	-4.293	1.328	0.068	0.973	0.332
Öz Yeterlilik	0.166	1.131	0.012	0.147	0.883
Motivasyon	2.917	1.016	0.225	2.871	0.005*

* p<0.05

R = 0.476 R² = 0.227

F(8.186) = 6.826 p = 0.000

Regresyon analizi sonuçlarına göre başarının yordanmasına ilişkin regresyon eşitliği aşağıda verilmiştir.

BAŞARI = -14.110 + 3.889 Cinsiyet – 2.129 Deneyim + 5.559 GANO +- 5.199 Lise Türü – 4.293 Problem Çözme + 0.166 Öz yeterlilik + 2.917 Motivasyon

Tablo 18’de görüldüğü gibi bağımsız başarı değişkeni ile cinsiyet (p = 0.031), GANO (p = 0.001), teknik liseyle karşılaştırıldığında meslek lisesi (p = 0.026) ve motivasyon (p = 0.005) arasında anlamlı bir ilişki bulunmaktadır. Diğer değişkenlerle başarı arasında anlamlı bir ilişki yoktur. Başka bir deyişle diğer değişkenlerin başarıya önemli bir katkısı yoktur.

Bağımlı değişken ve bağımsız değişkenler arasındaki ilişki incelenirken R ve R^2 değerleri bağımsız değişkenlerin bir arada bağımlı değişken üzerindeki toplam varyansı hakkında bilgi verir (Büyüköztürk, 2011). Regresyon analizi sonucunda $R=0.476$ ve $R^2= 0.227$ olduğuna göre başarı üzerinde etkisi araştırılan cinsiyet, deneyim, GANO, lise türü, problem çözme becerisi, öz yeterlilik ve motivasyon değişkenleri başarı üzerindeki toplam varyansın yaklaşık %23 ünü açıklamaktadır. Diğer bir deyişle cinsiyet, deneyim, GANO, lise türü, problem çözme becerisi, öz yeterlilik ve motivasyon değişkenleri başarı puanının ancak %23'ünü etkilemektedir.

Farklı ölçme birimleri ve varyanslara sahip bağımsız değişkenlerin bağımlı değişkene ilişkin göreceli önem sıralarını yorumlamak için regresyon katsayıları değil standardize edilmiş regresyon katsayıları olan β (beta) değerleri kullanılır. Bağımsız değişkenlerin sahip olduğu β değerleri işaretlerine dikkat edilmeksizin karşılaştırılır (Büyüköztürk, 2011). Araştırmada yapılan regresyon analizinde cinsiyet, deneyim, GANO, genel lise, meslek lisesi, problem çözme becerisi, öz yeterlilik ve motivasyon değişkenlerinin β değerleri sırasıyla 0.144, -0.071, 0.226, -0.133, -0.168, 0.068, 0.012, 0.225'dir. Bu sayısal değerler karşılaştırıldığında, standartlaştırılmış regresyon katsayısına göre (β) bağımsız değişkenlerin başarı üzerindeki göreceli önem sırası azalan sırada GANO, motivasyon, meslek lisesi, cinsiyet, genel lise, deneyim, problem çözme, öz yeterlilik şeklindedir. Başka bir deyişle GANO değişkeni başarıyı yordama bakımından değişkenler arasında en etkili değişkendir.

Bağımsız değişkenlerin her birinin bağımlı değişken üzerinde ne kadar anlamlı bir yordayıcı olduğunu belirlemek için t-testi sonuçları kullanılmaktadır. Tablo 19'a göre değişkenlerin t-testi sonuçları sırasıyla 2.168, -1.061, 3.265, -1.809, -2.250, 0.973, 0.147, 2.871 olarak hesaplanmıştır. Buna göre regresyon katsayılarının anlamlılığına ilişkin t-testi sonuçları incelendiğinde GANO değişkeninin diğer değişkenlere göre başarı puanı üzerinde daha anlamlı bir yordayıcı olduğu görülmektedir. İkinci sırayı Motivasyon almıştır.

4.4. MYO Öğrencilerinin Programlama Başarısı İle Öz Düzenleyici Öğrenme Stratejileri (Hedef Belirleme, İçsel Hedef, Dışsal Hedef, Öz Yeterlilik, Öz Yansırma, Deęer Verme, Üst Biliş) Arasındaki İlişkiye Ait Bulgular

MYO öğrencilerinin programlama başarısı ile öz düzenleyici öğrenme stratejileri (hedef belirleme, içsel hedef, dışsal hedef, öz yeterlilik, öz yansırma, deęer verme, üst biliş) arasındaki ilişkiye ait bulgular kullanılarak yapılan regresyon analizi sonucunda aşağıdaki tablo elde edilmiştir.

Tablo 19. MYO Öğrencilerinin Programlama Başarısı İle Öz Düzenleyici Öğrenme Stratejileri Arasındaki İlişkiye Ait Regresyon Analizi Sonuçları

Deęişken	B	Standart Hata	β	t	p
Sabit	-0.372	40.139		-0.090	0.928
Hedef Belirleme	0.511	0.540	0.097	0.946	0.346
İçsel Hedef	0.841	0.597	0.158	10.409	0.161
Dışsal Hedef	0.353	0.453	0.059	0.778	0.437
Öz Yeterlilik	10.509	0.685	0.240	20.203	0.029*
Öz Yansırma	-0.117	0.846	-0.016	-0.138	0.890
Deęer Verme	0.294	0.604	0.050	0.487	0.627
Üst Biliş	-0.730	0.712	-0.100	-10.026	0.306

* $p < 0.05$

$R = 0.426$ $R^2 = 0.182$

$F(7.187) = 5.938$ $p = 0.000$

Regresyon analizi sonuçlarına göre başarının yordanmasına ilişkin regresyon eşitliği aşağıda verilmiştir.

$BAŞARI = -0.372 + 0.511 \text{ Hedef Belirleme} + 0.841 \text{ İçsel Hedef} + 0.353 \text{ Dışsal Hedef} + 10.509 \text{ Öz Yeterlilik} - 0.117 \text{ Öz Yansırma} + 0.294 \text{ Deęer Verme} - 0.730 \text{ Üst Biliş}$

Tablo 19’da görüldüğü gibi bağımsız başarı deęişkeni ile Öz yeterlilik ($p = 0.029$) arasında anlamlı bir ilişki bulunmaktadır. Diğer deęişkenlerle başarı arasında

anlamlı bir ilişki yoktur. Başka bir deyişle diğer değişkenlerin başarıya önemli bir katkısı bulunmamaktadır.

Regresyon analizi sonucunda $R=0.426$ ve $R^2= 0.182$ olduğuna göre başarı üzerinde etkisi araştırılan değişkenler, başarı üzerindeki toplam varyansın yaklaşık %18 ini açıklamaktadır.

Araştırmada yapılan regresyon analizinde Hedef Belirleme, İçsel Hedef, Dışsal Hedef, Öz Yeterlilik, Öz Yansıma, Değer Verme, Üst Biliş değişkenlerinin β değerleri sırasıyla 0.097, 0.158, 0.059, 0.240, -0.016, 0.050, -0.100'dir. Bu sayısal değerler karşılaştırıldığında, standartlaştırılmış regresyon katsayısına göre (β) değişkenlerin başarı üzerindeki göreceli önem sırası azalan sırada Hedef Belirleme, İçsel Hedef, Dışsal Hedef, Öz Yeterlilik, Öz Yansıma, Değer Verme, Üst Biliş şeklindedir. Bu değerlere bakıldığında Hedef Belirleme değişkeni başarıyı yordama bakımından değişkenler arasında en etkili değişkendir.

Bağımsız değişkenlerin her birinin bağımlı değişken üzerinde ne kadar anlamlı bir yordayıcı olduğunu belirlemek için t-testi sonuçları kullanılmaktadır. Tablo 20'ye göre değişkenlerin t-testi sonuçları sırasıyla 0.946, 10.409, 0.778, 20.203, -0.138, 0.487, -10.026 olarak hesaplanmıştır. Buna göre regresyon katsayılarının anlamlılığına ilişkin t-testi sonuçları incelendiğinde Öz Yeterlilik değişkeninin diğer değişkenlere göre başarı puanı üzerinde daha anlamlı bir yordayıcı olduğu görülmektedir. İkinci sırayı İçsel Hedef değişkeni almıştır.

5. SONUÇ VE ÖNERİLER

Çalışmada MYO öğrencilerinin programlama başarıları incelenirken iki ayrım kullanılmıştır. Bunlardan birincisi öğrencilerin demografik bilgileri. Diğeri ise öz düzenleyici öğrenme stratejileridir.

5.1 Öğrencilerin Demografik Bilgilerine Göre Elde Edilen Sonuçlar

Öğrencilerin başarılarını demografik özelliklerinden olan cinsiyete göre karşılaştırdığımızda kız öğrencilerin erkek öğrencilere göre, çok yüksek bir fark olmasa da daha başarılı olduğu görülmektedir. Elde ettiğimiz sonucu literatürle karşılaştırdığımızda paralellik olduğunu söyleyebiliriz. Daha önce yapılan çalışmalara baktığımızda programlama başarıları ile cinsiyet arasında ya herhangi bir fark olmadığı yönünde sonuçlar bulunmuş (Byrne ve Lyons, 2001; McDowell ve diğerleri, 2003; Pillay ve Jugoo, 2005) ya da kız öğrencilerin daha başarılı olduğu tespit edilmiştir (Houston, 1993; Lau ve Yuen, 2009; Piro, 2004). Araştırmaya başlamadan önce genel beklenti, erkek öğrencilerin daha başarılı olması yönündeydi. Ancak 2000'li yıllarla beraber kız öğrencilerin teknolojiye karşı olan tutumunda meydana gelen iyileşmelerin, programlama başarılarını da etkilediğini söyleyebiliriz. 2000 öncesi yapılan çalışmalarda kız öğrencilerin teknolojiye karşı tutumu erkek öğrencilere göre daha olumsuzdu (Hawkins, 1985; Mims, 2012; NCREL, 2005; Schubert, 1986; Wilder ve diğerleri, 1985). 2000'li yıllardan sonra aradaki fark kapanmış, kız öğrencilerde erkek öğrenciler kadar teknolojiyle ilgilenir hale gelmiştir (Deniz ve diğerleri, 2006; Ikolo ve Okiy, 2012; Köse ve diğerleri, 2007).

Öğrencilerin daha önce programlamayla ilgili deneyimlerinin olması ile olmaması arasında her hangi bir fark bulunmamaktadır. Öğrencilerin hazır bulunuşluğu açısından daha önce programlama deneyimi olan öğrencilerin daha başarılı olması beklenirken sonuç bundan farklı çıkmıştır. Literatüre baktığımızda da ortak bir kanı olmadığını görmekteyiz. Yapılan önceki çalışmaların bir kısmında geçmiş deneyimlerin programlama başarıları üzerinde olumlu etkisi olduğu söylenirken (Byrne ve Lyons, 2001; Konvalina ve diğerleri, 1983), bunun aksini, yani önceki deneyimlerin programlama başarıları üzerinde etkisinin olmadığını söyleyen çalışmalarda mevcuttur

(Holden ve Weeden, 2003; Pillay ve Jugoo, 2005). Bu durumu hazır bulunuşluk açısından ele aldığımızda, önceden programlama deneyimine sahip olan öğrencilerin daha başarılı olacağı düşünülmektedir. Literatürde ortaya çıkan farkın çalışmaların araştırma yöntemleriyle açıklamamız mümkün. Çünkü Bryne ve Lyons (2001) ve Konvalina ve diğerleri (1983) çalışmalarını başlangıç düzeyinde tek bir ders kapsamında yürütmüşler ve deneyimin başarıyı etkilediğini bulmuşlardır. Holden ve Weeden (2003) ve Pillay ve Jugoo (2005) ise çalışmalarını daha uzun bir süreç içerisinde yapmışlar ve deneyimin programlama başarısı üzerinde herhangi bir etkisinin olmadığını tespit etmişlerdir. Bu çalışma da ise programlama başarısı tespit edilirken tek bir ders bazında çalışılmaması ve çalışmanın 2. sınıf öğrencileri üzerinde yürütülmesi zamanla öğrenciler arasındaki farkın kapanıp, önceden programlama deneyimi olan öğrencilerle, olmayan öğrencilerin aralarında herhangi bir fark bulunmamasını sağlamıştır. Nitekim MYO Bilgisayar Programcılığı Programında dersler, öğrenciler ilk defa programlama eğitimi alıyormuş gibi düzenlenir. Bu da zamanla öğrencilerin aralarındaki farkın kapanmasını sağlamış olabilir.

Öğrencilerin geçmiş deneyimleriyle ilişkili olarak incelenebilecek bir diğer değişken olan mezun olunan lise türüne baktığımızda ise teknik lise mezunlarının daha başarılı olduğu görülmektedir. Teknik lise mezunlarının, lise dönemlerinde programlama eğitimi almış olmalarının onları MYO düzeyinde avantajlı hale getirerek daha başarılı olmalarını sağladığını söyleyebiliriz. Ancak bu durum önceki deneyimlerin programlama başarıları üzerinde etkisi olmaması sonucuyla çelişmektedir. Bunun yanı sıra normal meslek lisesi mezunları ile genel lise mezunları arasında herhangi bir fark görülmemektedir. Bu durumu teknik lise ve meslek lisesi bilişim teknolojileri bölümlerinin dal programlarıyla alakalı olduğunu söyleyebiliriz. Bu liselerde bilişim teknolojileri bölümü altında 4 dal (ağ işletmenliği, bilgisayar teknik servisi, veri tabanı programcılığı, web programcılığı) bulunmaktadır. Bu bölümlerin bazılarında (ağ işletmenliği, bilgisayar teknik servisi) yoğun olarak programlama eğitimi verilmemektedir. Araştırma örneklemimiz içerisinde yer alan teknik lise mezunlarının programlama derslerinin daha yoğun verildiği alanlardan (veri tabanı programcılığı, web programcılığı) gelmiş olması onların daha başarılı olmalarını sağlamış olabilir.

Öğrencilerin genel akademik not ortalamaları ile sınav başarısı arasında pozitif bir ilişki görülmüştür. GANO'su yüksek olan öğrencilerin sınav başarısı daha yüksek olmuştur. Müfredatlarında sadece programlamayla ilgili dersler olmayan bilgisayar programcılığı programlarında, başarılı olan öğrencilerin tüm derslerinde başarılı olduğunu söyleyebiliriz.

5.2 MYO Öğrencilerinin Programlama Başarıları İle Demografik Bilgileri (Cinsiyet, Mezun Oldukları Okul Türü, Programlamayla İlgili Geçmiş Deneyimleri, Genel Akademik Not Ortalamaları), Problem Çözme Ve Analitik Düşünme Becerisi, Kişisel Öz Yeterlilik, Motivasyonları Arasındaki İlişkiye Ait Sonuçlar

Öğrencilerin başarısını etkileyebileceği düşünülen demografik bilgileri ve bunun yanı sıra problem çözme ve analitik düşünme becerisi, kişisel öz yeterlilik ve motivasyonları incelendiğinde, bu değişkenlerin toplam varyansın %23'ünü açıkladığı görülmektedir. Regresyon katsayılarının anlamlılığına ilişkin T-testi sonuçları incelendiğinde ise cinsiyet, mezun olunan okul türü, GANO ve motivasyonun başarı üzerinde etkili bir yordayıcı olduğu görülmektedir. Diğer değişkenlerin programlama başarısı üzerinde anlamlı bir etkisi bulunamamıştır.

Demografik bilgilere ait sonuçları detaylı olarak ele almıştık. Diğer değişkenleri ele aldığımızda motivasyonun başarı üzerinde etkili olduğu görülmektedir. Elde ettiğimiz bu sonuç literatürde yapılan çalışmalarla paralellik göstermektedir (Jiau ve diğerleri, 2009; Matthíasdóttir, 2006). Hayatın her alanında başarının önemli bir parçası olan motivasyon, programlama başarısının da önemli bir değişkeni durumundadır.

Çalışmamızda elde ettiğimiz sonuçlara göre problem çözme ve analitik düşünme becerisinin programlama başarısı üzerinde herhangi bir etkisi olmadığı görülmüştür. Elde ettiğimiz sonuç literatürdeki diğer çalışmalarla çelişmektedir. Yapılan diğer çalışmalarda problem çözme ve analitik düşünme becerisinin programlama başarısı üzerinde etkili olduğu vurgulanmıştır (Grant, 2003; Ismail ve diğerleri, 2010; Pillay ve Jugoo, 2005).

5.3 MYO Öğrencilerinin Programlama Başarısı İle Öz Düzenleyici Öğrenme Stratejileri (Hedef Belirleme, İçsel Hedef, Dışsal Hedef, Öz Yeterlilik, Öz Yansırma, Deęer Verme, Üst Biliş) Arasındaki İlişkiye Ait Sonuçlar

Öğrencilerin programlama başarısı ile öz düzenleyici öğrenme stratejileri arasındaki ilişki incelendiğinde, ele alınan öğrenme stratejilerinin toplam varyansın %18'ini açıkladığı görölmektedir. Regresyon katsayılarının anlamlılığına ilişkin yapılan t testi sonuçları incelendiğinde öz yeterlilik deęişkeninin başarı üzerinde anlamlı bir etkisinin olduęu görölmektedir. Ele alınan dięer deęişkenlerin başarı üzerinde anlamlı bir etkisi görölmemektedir.

Öz düzenleyici öğrenme stratejilerinin programlama başarısına ait varyansın %18'ini açıklaması, literatürde yapılan dięer derslere yönelik çalışmalara göre düşük kalmaktadır.

5.4 Öneriler

5.4.1 Uygulamaya yönelik öneriler

1. Yapılan çalışmada öğrencilere açık uçlu sorulardan oluşan bir başarı sınavı uygulanmıştır. Sınavın açık uçlu sorulardan oluşması öğrencilerin yeterince cevap vermelerini engellemiştir. Yapılacak dięer çalışmalarda açık uçlu soruların sayısı azaltılarak, çoktan seçmeli soruların eklenmesi yararlı olacaktır.
2. Yapılan çalışmada kullanılan ölçeklerin çok olması ve aynı anda uygulanması öğrencilerin sıkılmasına neden olmuştur. Yapılacak dięer uygulamalarda imkan varsa ölçeklerin ayrı zamanlarda uygulanması faydalı olacaktır.

5.4.2 Yeni yapılacak arařtırmalara ynelik neriler

1. Bu arařtırma sonularının genellenebilirliđini artırmak iin benzer alıřmaların farklı rneklem grupları zerinde tekrarlanması yararlı grlebilir.
2. đrencilerin programlama bařarısını etkilediđi grlen motivasyon diđer alıřmalarda đrencilerin programlamaya ynelik motivasyonlarını etkileyen deđiřkenler aısından incelenebilir.
3. Yapılacak yeni alıřmalarda programlama bařarısını etkileyebilecek diđer deđiřkenler kullanılabilir.
4. Yapılan bu arařtırma MYO Bilgisayar Programcılıđı Programı 2. sınıf đrencileri zerinde gerekleřtirilerek tek seferde yapılmıřtır. Yapılacak olan yeni alıřmalarda, programa yeni bařlayan đrenciler eđitimleri srecince takip edilerek, zaman ierisinde incelenen deđiřkenlerin durumları takip edilebilir.
5. Arařtırma sonularında đrenci nbilgisinin ve đrenci zgveninin bařarıda etkisi olduđu gze arpmaktadır. Bu durumda đrenci nbilgisini dikkate alan ve z gvenini artıran đretim uygulamalarının bilgisayar programcılıđında bařarıya etkisi gelecek arařtırmalarda alıřılabilir.
6. Programlama mfredatını btncl olarak basitten zora dođru sıralama yaklařımı ile đreten đretim stratejileri kullanarak, rneđin Reigeluth'un ayrıntılandırma kuramı kullanarak, tasarlanan đretimin đrenci bařarısına olan etkisi incelenebilir. Reigeluth'un ayrıntılandırma kuramı đretimde tm konuları sırasıyla vermek yerine mfredatın btncl olarak basitten zora dođru tekrarlar halinde verilmesine dayanmaktadır, bylelikle đrencinin okutulan konular hakkında her zaman nbilgisi bulunmaktadır.
7. Yine bir diđer đretim yaklařımı olan problem temelli đretim ile tasarlanmış đrenme ortamlarında đrencilerin programlama derslerindeki bařarısı incelenebilir. Problem temelli đrenme ortamları đrenciye sadece bilgisayar programcılıđının teknik bilgisini deđil aynı zamanda đrenciye analitik dřnme ve problem zme becerileri ile beraber zgvenine ve motivasyonuna katkı

sağlayacağından, bilgisayar programcılığı derslerindeki akademik başarısını da arttırabilir.

8. Öğrencilerin bilgisayar programcılığı konusunda başarılarını arttırmaya yönelik olarak tasarlanan her iki öğretim yaklaşımında da tam deneysel veya yarı deneysel bir yaklaşım ele alınabilir. Ayrıntılandırma kuramı veya problem temelli öğrenme yaklaşımı ile tasarlanmış öğrenme ortamları, meslek yüksek okullarında mevcut yapıda öğretilen bilgisayar programcılığı dersleri ile öğrenci başarısı ve özgüven boyutlarında karşılaştırılabilir.

KAYNAKÇA

- Akay, H. (2006). *Problem kurma yaklaşımı ile matematik öğretiminin öğrencilerin akademik başarısı, problem çözme becerisi ve yaratıcılığı üzerindeki etkisinin incelenmesi*. Doktora Tezi, Gazi Üniversitesi Eğitim Bilimleri Enstitüsü, Ankara.
- Akbaba, S. (2006). Eğitimde motivasyon. *Kazım Karabekir Eğitim Fakültesi Dergisi*, 13.
- Ala-Mutka, K. (2004). Problems in learning and teaching programming. *Institute of Software Systems, Tampere University of Technology*.
- Alcı, B. ve Altun, S. (2007). Lise öğrencilerinin matematik dersine yönelik özdüzenleme ve bilişüstü becerileri, cinsiyete, sınıfa ve alanlara göre farklılaşmakta mıdır? *Çanakkale Üniversitesi Sosyal Bilimler Enstitüsü Dergisi*, 16(1), 33-44.
- Alican, F. (2008). Yazılım sektörü eylem koordinasyonu: Dünyadaki başarılı örneklerin incelenmesi. İstanbul. Web: <http://www.yasad.org.tr/UserFiles/File/Yazilim%20Eylem%20Koordinasyonu%20-%20Fuat%20Alican.pdf> adresinden 23 Ekim 2011'de alınmıştır.
- Arabacıoğlu, T., Bülbül, H. İ. ve Filiz, A. (2007). *Bilgisayar programlama öğretiminde yeni bir yaklaşım*. Akademik Bilişim 2007 sempozyumunda sunuldu, Kütahya Dumlupınar Üniversitesi.
- Aypay, A. (2010). The adaptation study of general self-efficacy (GSE) scale to Turkish. *Inonu University Journal Of The Faculty Of Education*, 11(2), 113-131.
- Bandura, A. (1982). Self-efficacy mechanism in human agency. *American Psychologist*, 37(2).
- Başar, E. (2001). *Genel öğretim yöntemleri*. Samsun: Kardeşler Ofset ve Matbaa.
- Bayraktar, G., Güngörmüş, H. A., Gülbahçe, Ö., Şahin, S. ve Bastık, C. (2011). Beden eğitimi ve türkçe öğretmen adaylarının problem çözme becerisi algı düzeyleri açısından karşılaştırılması. *New World Sciences Academy*, 6(2).
- Bellardinelli, M. (1971). *Criteria and educational tools in the training of programmers*.
- Bland, L. S. (2005). *The effects of a self-reflective learning process on student art performance*. Doktora Tezi.
- Bloom, B. (1995). *İnsan nitelikleri ve okulda öğrenme* (çev. D. A. Özçelik, 2. Baskı). Ankara: Milli Eğitim Basımevi.
- Boulay, B. (1989). *Some difficulties of learning to program*. Hillsdale, NJ: Lawrence Erlbaum.

- Büyüköztürk, Ş. (2011). *Sosyal bilimler için veri analizi el kitabı*. Ankara: Pegem Akademi.
- Büyüköztürk, Ş., Akgün, Ö. E., Kahveci, Ö., & Demirel, F. (2004). Güdülenme ve öğrenme stratejileri ölçeğinin Türkçe formunun geçerlik ve güvenirlik çalışması. *Kuram ve Uygulamada Eğitim Bilimleri*, 4(2), 207-239.
- Byrne, P. ve Lyons, G. (2001). *The effect of student attributes on success in programming*. Proceedings of The 6th Annual Conference On Innovation And Technology In Computer Science Education sempozyumunda sunuldu, İngiltere.
- Cassady, J.C. (2001). Self-reported GPA and SAT: a methodological note. *Practical Assessment, Research & Evaluation*, 7(12).
Web:<http://PAREonline.net/getvn.asp?v=7&n=12> adresinden 26 Eylül 2007'de alınmıştır.
- Chalk, P. D. (2000). Community of practice: Learning the craft of programming. *Association for Computer Machinery*.
- Çamoğlu, K. (2009). *Programlama ve veri tabanı mantığı*. İstanbul: Kodlab.
- Çiltaş, A. (2011). Eğitimde öz-düzenleme öğretiminin önemi üzerine bir çalışma. *Mehmet Akif Ersoy Üniversitesi Sosyal Bilimler Enstitüsü Dergisi*, 3(5).
- Deci, E. L., Koestner, R. ve Ryan, R. M. (1999). A meta-analytic review of experiments examining the effects of extrinsic rewards on intrinsic motivation. *Psychological Bulletin*, 125(6).
- Delialioğlu, Ö., Öncü, S. ve Çakır, H. (2008). Comparison of students' success in an international technology certificate program. *H. U. Journal of Educational*, 35, 75-86.
- Deniz, S., Görgen, İ. ve Şeker, H. (2006). Tezsiz yüksek lisans öğretmen adaylarının teknolojiye yönelik tutumları. *Eurasian Journal of Educational Research*, 23, 62-71.
- Erdoğan, B. (2005). *Programlama başarısı ile akademik başarı, genel yetenek, bilgisayara karşı tutum, cinsiyet ve lise türü arasındaki ilişkilerin incelenmesi*. Yayınlanmış Yüksek Lisans Tezi, Marmara Üniversitesi Eğitim Bilimleri Enstitüsü, İstanbul.
- Erdoğan, Y., Emin, A. ve Kabaca, T. (2008). Exploring the psychological predictors of programming achievement. *Journal of Instructional Psychology*, 35(3), 264-270.
- Ersoy, H., Madran, R. O. ve Gülbahar, Y. (2011). *Programlama dilleri öğretimine bir model önerisi: Robot programlama*. Akademik Bilişim 2011 sempozyumunda sunuldu, İnönü Üniversitesi.
- Eryılmaz, S. (2003). *Algoritma tasarlama ve programlamaya giriş*. Ankara: Detay Yayıncılık.

- Fidan, N. (1986). *Okulda öğretme ve öğrenme*. Ankara: Kadioğlu Matbaası.
- Garner, S. (2003). Learning resources and tools to aid novices learn programming. *Information Science InSITE*.
- Gawith, G. (1995). A serious look at self-efficacy: Or waking beeping Slooty. Web: from <http://www.cegsa.sa.edu.au/conference/acec98.htm> adresinden 10 Ekim 2011'de alınmıştır.
- Goold, A. ve Rimmer, R. (2000). Factors affecting performance in first-year computing. *ACM SIGCSE Bulletin*, 32(2), 39-43.
- Grant, N. S. (2003). *A study on critical thinking, cognitive learning style, and gender in various information science programming classes*. The 4th Conference On Information Technology sempozyumunda sunuldu, Indiana, USA.
- Gültekin, K. (2006). *Çoklu ortamın bilgisayar programlama başarısı üzerine etkisi*. Yayınlanmış Yüksek Lisans Tezi, Hacettepe Üniversitesi Fen Bilimleri Enstitüsü, Ankara.
- Harman, G. ve Çelikler, D. (2012). Eğitimde hazır bulunuşluğun önemi üzerine bir derleme çalışması. *Eğitim ve Öğretim Araştırmaları Dergisi*, 1(3).
- Haşlaman, T. (2005). *Programlama dersi ile ilgili öz düzenleyici öğrenme stratejileri ile başarı arasındaki ilişkilerin incelenmesi: bir yapısal eşitlik modeli*. Yüksek Lisans Tezi, Hacettepe Üniversitesi Fen Bilimleri Enstitüsü, Ankara.
- Haşlaman, T. ve Aşkar, P. (2007). Programlama dersi ile ilgili öz düzenleyici öğrenme stratejileri ile başarı arasındaki ilişkilerin incelenmesi. *Investigating the Relationship Between Self-Regulated Learning Strategies and Achievement in a Programming Course*(32), 110.
- Hawkins, J. (1985). Computers and girls: Rethinking the issues. *Sex Roles*, 13(3/4), 165-180.
- Henderson, P. B. (1986). *Proceedings of the 17th SIGCSE '86*: Technical symposium on Computer Science Education sempozyumunda sunuldu, New York.
- Heppner, P. P. ve Peterson, C. H. (1982). The development and implications of a personal-problem solving inventory. *Journal Of Counseling Psychology*, 29.
- Holden, E. ve Weeden, E. (2003). *The impact of prior experience in an information technology programming course sequence*. Conference On Information Technology Education sempozyumunda sunuldu, New York.
- Housten, D. M. (1993). *An exploration and analysis of the relationship among learning styles, teaching styles, gender and performance in a college computer science course*. Yayınlanmış Doktora Tezi, Kansas State University, Manhattan, Kansas.

- Ikolo, V. E. ve Okiy, R. B. (2012). Gender differences in computer literacy among clinical medical students in selected southern nigerian universities. *Library Philosophy and Proctice*.
- İmal, N. ve Eser, M. (2005). Programlama dili öğrenmedeki zorluklar ve çözüm yaklaşımları.
- Ismail, M. N., Ngah, N. A. ve Umar, I. N. (2010). Instructional strategy in the teaching of computer programming: A need assessment analyses. *The Turkish Online Journal of Educational Technology*, 9(2).
- 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.
- Kak, A. (2008). Teaching programming. Web: <https://engineering.purdue.edu/kak/TeachingProgramming.pdf> adresinden 15 Aralık 2012'de alınmıştır.
- Karasar, N. (1999). *Bilimsel araştırma yöntemi*. Ankara: Nobel Yayın Dağıtım.
- Kauffman, D. F. (2004). Self-regulated learning in web-based environments: Instructional tools designed to facilitate cognitive strategy use, metacognitive processing and motivational beliefs. *J. Educational Computing Research*, 30, 139-161.
- Keskinsoy, A. (2010). *Mesleki liselerde görsel programlama başarısını etkileyen faktörler*. Yayınlanmış Yüksek Lisans Tezi, Marmara Üniversitesi Eğitim Bilimleri Enstitüsü, İstanbul.
- Konuk, M. ve Öztürk, A. (2010). *Üniversite - sanayi işbirliği ve teknokentlere bakış*. Paper presented at the Cumhuriyetimizin 100. Yılına Doğru Üniversite Vizyonumuz, Ankara.
- Konvalina, J., Stephens, L. ve Wileman, S. (1983). Identifying factors influencing computer science aptitude and achievement. *AEDS I*, 16(2), 106 - 112.
- Köse, S., Gencer, A. S. ve Gezer, K. (2007). Meslek yüksekokulu öğrencilerinin bilgisayar ve internet kullanımına yönelik tutumları. *Pamukkale Üniversitesi Eğitim Fakültesi Dergisi*, 21(1), 44-54.
- Krouse, J. H. ve Krouse, H. J. (1981). Toward a multimodal theory of academic achievement. *Educational Psychologist*, 16, 151-164.
- Kuncel, N.R., Crede, M., & Thomas, L.L. (2005). The validity of self-reported grade point averages, class ranks, and test scores: A meta-analysis and review of the literature. *Review of Educational Research*, 75(1), 63-82.
- Kurbanoglu, S. S. (2004). Öz-yeterlik inancı ve bilgi profesyonelleri için önemi. *Bilgi Dünyası*, 5(2), 137.

- Lai, E. R. (2011). Motivation: A literature review. Web: https://pearsonassessments.com/hai/images/tmrs/Motivation_Review_final.pdf adresinden 15 Ocak 2013'te alınmıştır.
- Lau, W. W. F. ve Yuen, A. H. K. (2009). Exploring the effects of gender and learning styles on computer programming performance: Implications for programming pedagogy. *British Journal of Educational Technology*, 40(4).
- Marciniak, J. J. (2002). Software engineering, a historical perspective. *Encyclopedia of Software Engineering*, 2, 1479-1483.
- Matthíasdóttir, Á. (2006). *How to teach programming languages to novice students? Lecturing or not?* International Conference on Computer Systems and Technologies sempozyumunda sunuldu, Bulgaristan.
- McDowell, C., Werner, L., Bullock, H. E. ve Fernald, J. (2003). *The impact of pair programming on student performance, perception and persistence*. 25. International Conference on Software Engineering sempozyumunda sunuldu, Portland.
- MEB. (2011). Bilişim teknolojileri alanı çerçeve öğretim programı. Web: http://mebk12.meb.gov.tr/meb_iys_dosyalar/35/25/966426/dosyalar/2012_12/05121710_bc10ae0d38334ff88785a070d294f0c4.pdf adresinden 8 Ocak 2013'te alınmıştır.
- Miliszewska, I., Venables, A. ve Tan, G. (2008). Improving progression and satisfaction rates of novice computer programming students through ACME - analogy, collaboration, mentoring, and electronic support. [Article]. *Issues in Informing Science & Information Technology*, 5, 311-323.
- Miller, M. ve Nunn, G. D. (2001). Using group discussions to improve social problem-solving and learning. *Education*, 121(3).
- Mims, M. (2012). The importance of technology usage in the classroom, does gender gaps exist. *Contemporary Issues in Education Research*, 5(4), 271 - 278.
- Naur, P. ve Randall, B. (1968). Paper presented at the Software Engineering: A Report on a Conference Sponsored by the NATO Science Committee, Germany.
- NCREL. (2005). Using technology to improve student achievement. Web: <http://www.ncrel.org/sdrs/areas/issues/methods/technlgy/te800.htm> adresinden 22 Aralık 2012'de alınmıştır.
- Norvell, T. S. ve Bruce-Lockhart, M. P. (2004). *Teaching computer programming with program animation*. Paper presented at the Computer and Software Engineering Education, Canadian Conference.
- OSYM. (2011a). Merkezi yerleştirme ile öğrenci alan yükseköğretim lisans programları. Web: <http://www.osym.gov.tr/dosya/1-57873/h/tablo-4.pdf> adresinden 22 Ekim 2011'de alınmıştır.

- OSYM. (2011b). Merkezi yerleştirme ile öğrenci alan yükseköğretim önlisans programları. Web: <http://www.osym.gov.tr/dosya/1-57873/h/tablo-4.pdf> adresinden 22 Ekim 2011'de alınmıştır.
- Özevin, B. (2006). *Oyun, dans ve müzik dersine ilişkin motivasyon ölçeği*. Paper presented at the Ulusal Müzik Eğitimi Sempozyumu, Denizli.
- Pillay, N. ve Jugoo, V. R. (2005). An investigation into student characteristics affecting novice programming performance. *ACM SIGCSE*, 37(4), 107 - 110.
- Pintrich, P. R. ve De Groot, E. V. (1990). Motivational and self regulated learning componenets of classroom academic peformance. *Journal of Educational Psychology*, 82(1), 33-40.
- Pintrich, R. R. (2000). *The role of goal orientation in self-regulated learning*. San Diego, CA: Academic Pres.
- Pioro, B. T. (2004). *Performance in an introductory computer programming course as a predictor of future success for engineering and computer science majors*. International Conference on Engineering Education sempozyumunda sunulmuştur, Gainesville.
- Rizemberg, R. ve Zimmerman, B.J. . (1992). Self-regulated learning in gifted students. *Roeper Review*, 15(1), 98-101.
- Robins, A., Rountree, J. ve Rountree, N. (2003). Learning and teaching programming: A review and discussion. *Computer Science Education*, 13(2), 137–172.
- Ruban, L. ve Reis, S. M. (2006). Patterns of self-regulatory strategy use among low-achieving and high-achieving university students. *Roeper Review*, 28(3).
- Schubert, J. G. (1986). Gender equity in computer learning. *Theory Into Practice*, 25(4), 267.
- Senemoğlu, N. (2009). *Gelişim, öğrenme ve öğretim kuramdan uygulamaya*. Ankara: Pegem Akademi.
- Sureau, D. (2001). History of programming languages and their evolution. Web: <http://www.scriptol.com/programming/history.php> adresinden 15 Aralık 2012'de alınmıştır.
- Taşkın, M. M. (2002). Yazılım sektörü. Web: http://scholar.googleusercontent.com/scholar?q=cache:qeETH_d8h0cJ:scholar.google.com/+yaz%C4%B1l%C4%B1m+sekt%C3%B6r%C3%BCnde+insan+g%C3%BCc%C3%BCn%C3%BCn+%C3%B6nemi&hl=tr&as_sdt=0,5 adresinden 22 Ekim 2011'de alınmıştır.
- Taylan, S. (1990). *Heppner'in problem çözme envanterinin uyarlama, geçerlik ve güvenilirlik çalışmaları*. Yayınlanmamış Yüksek Lisans Tezi, Ankara Üniversitesi Sosyal Bilimler Enstitüsü, Ankara.

- Thomas, L., Ratcliffe, M., Woodbury, J. ve Jarman, E. (2002). *Learning styles and performance in the introductory programming sequence*. The 33rd SIGCSE Technical Symposium On Computer Science Education sempozyumunda sunulmuştur, New York, USA.
- Tukiainen, M. ve Mönkkönen, E. (2002). *Programming aptitude testing as a prediction of learning to program*. 14th Workshop of the Psychology of Programming Interest Group sempozyumunda sunulmuştur, Brunel University.
- Üredi, I. ve Üredi, L. (2005). İlköğretim 8. sınıf öğrencilerinin öz-düzenleme stratejileri ve motivasyonel inançlarının matematik başarısını yordama gücü. *Mersin Üniversitesi Eğitim Fakültesi Dergisi*, 1(2), 250-260.
- Wilder, G., Mackie, D. ve Cooper, J. (1985). Gender and computers: Two surveys of computer-related attitudes. *Sex Roles*, 13(3/4), 215-228.
- Winslow, L. E. (1996). Programming pedagogy – A psychological overview. *SIGCSE Bulletin*, 28.
- Yalçın, B., Tetik, S. ve Açıkgöz, A. (2010). Yüksekokul öğrencilerinin problem çözme becerisi algıları ile kontrol odağı düzeylerinin belirlenmesine yönelik bir araştırma. *Organizasyon ve Yönetim Bilimleri Dergisi*, 2(2).
- Yaşar, E. (2011). *Algoritma ve programlamaya giriş* (3. baskı). Trabzon: Murathan Yayınevi.
- Yılmaz, Ş. (2007). *Yazılım sektörünün gelişme dinamikleri : Dünyada ve Türkiye'de sektörel gelişme stratejileri*. Yayınlanmış Yüksek Lisans Tezi, Marmara Üniversitesi Sosyal Bilimler Enstitüsü, İstanbul.
- Zimmerman, B. J., Bonner, S. ve Kovach, R. (1996). Developing self-regulated learners, beyond achievement to self-efficacy. *American Psychological Association*.

EKLER

Ek1 Çalışmada kullanılan anket ve ölçekler

Sevgili Öğrenciler,

Yapılan bu çalışmada sizlerin derslerinizdeki başarı ya da başarısızlıklarınızı etkileyen faktörleri belirleyerek, elde edilen sonuçlara göre başarınızı artırmaya yönelik tedbirler alınması amaçlanmaktadır. Cevaplayacağınız anketler ve programlama başarı sınavı ile yapılacak olan bu çalışmaya katkı sağlamış ve daha iyi bir eğitim almak için alınacak tedbirlerde sizler de fikirlerinizi belirtmiş olacaksınız. Çalışma esnasında sizlerden alınacak olan bilgiler kesinlikle sizi değerlendirmek veya yargılamak amaçlı değildir. Bu yüzden araştırmanın amacı dışında sizlerden isim, okul numarası gibi kimliğinizi belli edecek bilgiler istenmeyecektir. Vereceğiniz cevaplar bilimsel bir araştırmaya ışık tutacağından samimi cevaplar vermeniz önem taşımaktadır.

Samimiyetiniz için teşekkür ederim.

Lütfen aşağıdaki soruları size uygun şekilde doldurunuz.

1. Cinsiyetiniz : () Erkek () Kız

2. Yaşınız:

3. Mezun Olduğunuz Okul Türü :

() Genel Lise () Meslek Lisesi () Teknik Lise

() Anadolu Teknik Lisesi () Anadolu Meslek Lisesi

4. Mezun Olduğunuz Bölüm (Mezun Olduğunuz Okul Türüne Göre):

Meslek Lisesi : () Yazılım () Donanım () Ağ () Web Tasarım () Diğer.....

Genel Lise : () Sayısal () Sözel () Eşit Ağırlık () Diğer

5. Lise Mezuniyet Dereceniz :

() 2 – 2,49 () 2,5 – 2,99 () 3 - 3,49 () 3,5 – 3,99 () 4 – 4,49 () 4,5 - 5

6. Bu bölüme başlamadan önce herhangi bir programlama eğitimi aldınız mı?

() Evet () Hayır

PROBLEM ÇÖZME ENVANTERİ

Sevgili Öğrenciler;

İnsanlar günlük sorunlara farklı tepkilerde bulunurlar. Bu ölçekte verilen ifadeler de insanların günlük olaylar ve kişisel problemlere nasıl tepkide bulunduklarını belirlemeye yöneliktir. Aşağıdaki 12 maddeyi dürüst ve samimi olarak, sizin buna benzer problemleri nasıl halletmeye çalıştığınızı gösterecek şekilde işaretleyiniz. Her bir ifadeyi dikkatlice okuduktan sonra verilen ifade ile ne kadar uzlaştığının veya ayrıldığının derecesini ifadenin karşısına (x) sembolü koyarak işaretleyiniz.

Madde No	PROBLEM ÇÖZME ÖLÇEĞİ	Tamamen Katılıyorum	Kısmen Katılıyorum	Çok Az Katılıyorum	Çok Az Katılmıyorum	Kısmen Katılmıyorum	Hiç Katılmıyorum
1	Bir sorunu çözmek için plan yaptığımda bu planın işe yarayacağından oldukça emin olurum						
2	Değişik bir durumla karşılaşsam da ortaya çıkabilecek problemleri halledeceğimden eminim						
3	Yeni ve zor sorunları çözme yeteneğime güvenirim						
4	Yeterli çaba gösterdiğimde ve zamanım olduğunda karşılaştığım bütün sorunları çözebileceğime inanırım						
5	Bir karar verdikten sonra, beklediğim sonuçla gerçekleşen sonuç genellikle aynıdır						
6	Başlangıçta çözümü mümkün gibi görünmese bile pek çok sorunu çözebilme yeteneğim vardır						
7	Genellikle sorunlarımı çözebilmek için yaratıcı ve etkili seçenekler bulabilirim						
8	Bir problemi çözerken kararlar alırım ve sonunda bunlardan mutlu olurum						
9	Belli bir davranışın sonucunu tahmin etmeye çalışırım						
10	Seçenekleri karşılaştırmak ve karar vermek için sistematik bir yöntem kullanırım						
11	Bir sorun olduğunu fark ettiğimde, yaptığım ilk şeylerden birisi sorunun ne olduğunu tam olarak anlamaya çalışmaktır.						
12	Bir sorunu çözebilmek için belli bir yol izledikten sonra beklediğim sonuçla ortaya çıkan sonucu karşılaştırırım						

MOTİVASYON VE ÖZ YETERLİLİK ÖLÇEĞİ

Sevgili Öğrenciler;

İnsanlar günlük yaşamlarında birçok problemle karşılaşır. Bunlardan kimisinin üstesinden başarıyla gelirken kimisinde ise başarısız olunmaktadır. Hazırlanmış olan bu ölçekte sizlerin herhangi bir durumla karşılaştığınızda nasıl tepkiler verdiğiniz ortaya çıkarmaya yöneliktir. Aşağıdaki 17 ifadeyi dikkatlice okuduktan sonra verilen ifade ile ne kadar uzlaştığınızın veya ayrıldığınızın derecesini ifadenin karşısına (x) sembolü koyarak işaretleyiniz.

Madde No	Motivasyon ve Öz Yeterlilik Ölçeği	Doğru Değil	Biraz Doğru	Kararsızım	Daha Doğru	Tümüyle Doğru
1	Yeni bir durumla karşılaştığımda ne yapmam gerektiğini bilirim	Öz yeterlilik				
2	Beklenmedik durumlarda nasıl davranmam gerektiğini her zaman bilirim					
3	Bana karşı çıkıldığında kendimi kabul ettirecek çare ve yolları bulurum					
4	Ne olursa olsun, üstesinden gelirim					
5	Güç sorunların çözümünü eğer gayret edersem her zaman başarırım					
6	Tasarılarımı gerçekleştirmek ve hedeflerime erişmek bana güç gelmez					
7	Bir sorunla karşılaştığım zaman onu halledebilmeye yönelik birçok fikirlerim vardır					
8	Güçlükleri soğuk kanlılıkla karşılarım, çünkü yeteneklerime her zaman güvenebilirim					
9	Ani olayların da hakkından geleceğimi zannediyorum					
10	Her sorun için bir çözümüm vardır					
11	Programlamada başarılı olacağıma inanıyorum	Motivasyon				
12	Programlama eğitimim için verilen en zor konuları anlayacağımdan eminim					
13	Programlama derslerinde verilen materyaller kolayca anlaşılabilir materyallerdi					
14	Programlama derslerinde anlatılan konular sevdiğim konulardır					
15	Programlama derslerinde öğrendiklerimi diğer derslerimde uygulayabilirim					
16	Programlama derslerinden yüksek not almak için benim için çok önemli					
17	Okuduğum alan benim için çok değerli					

BAŞARI SINAVI (C#)

Sevgili öğrenciler,

Aşağıdaki sınav sizin programlama başarınızı ölçmek amacıyla hazırlanmış ve 5 sorudan oluşmaktadır. Soruları cevaplandırırken bildiğiniz herhangi bir programlama dilini ve programlama yöntemini kullanabilirsiniz. Bu sınav sizin mevcut dersleriniz için herhangi bir ölçme niteliği taşımamaktadır ve notlarınızı etkilemeyecektir. Bu yüzden sınavı cevaplandırırken bireysel olarak yardım olmadan cevaplandırınız.

Soru 1: Aşağıdaki kavramları açıklayınız.

Programlama:

Değişken:

Döngü:

Karar Yapısı:

Dizi:

Soru 2a: Aşağıda verilen programı for döngü yapısını kullanarak tekrar yazınız.

Örnek Program	For döngüsüyle yazılmış hali
<pre>int k = 1; do { listBox1.Items.Add(k); k = k + 1; } while (k < 50);</pre>	

Soru 2b : Aşağıda switch – case yapısıyla oluşturulmuş programı if – else if – else yapısını kullanarak tekrar yazınız.

Switch – Case	If – else if – else yapısıyla yazılmış hali
<pre>switch (k) { case 1: textBox1.Text = "Bir"; break; case 2: textBox1.Text = "İki"; break; default: textBox1.Text = "Tanımsız sayı"; break; }</pre>	

Soru 3: Integer (tamsayı) türünde tanımlanmış 15 elemanlı bir tek boyutlu diziye rastgele sayılar yerleştirilmiştir. Bu sayıların en büyüğü ve en küçüğü arasındaki farkı kullanıcıya bildirecek olan programı yazınız (Kullanılacak olan dizinin önceden oluşturulduğunu var sayarak programınızı oluşturunuz).

Soru 4: Kullanıcıdan 20 tane sayı alarak bu sayılardan kaç tanesinin 50'den büyük, kaç tanesinin 50'ye eşit ve kaç tanesinin 50'den küçük olduğunu bularak, bu sayıların ortalamasını kullanıcıya bildirecek olan programı yazınız.

Soru 5: Aşağıdaki program çalıştırıldığında a, b, c, d, e, f değişkenlerinin alacağı son değerleri yazınız.

Alınacak en son değerleri buraya yazınız.

a	b	c	d	e	f

```

static void Main(string[] args)
{
    int a = 2, b = 2, c, d = 20, e = 1, f = 1;
    if (a == 1)
        a = a + 1;
    if (a == 2)
        a = a + 2;
    if (a == 3)
        a = a + 3;
    if (a == 4)
        a = a + 4;
    switch (b)
    {
        case 1:
            b = b + 1;
            break;
        case 2:
            b = b + 2;
            break;
        case 3:
            b = b + 3;
            break;
        default:
            b = b + 4;
            break;
    }

    c = 11 / Convert.ToInt32(8 / 3);
    for (int i = 0; i < 4; i++)
    {
        d = d - i;
    }
    while (f < 8)
    {
        f = f + 1;
        if (a < 4)
            e = e + f;
        else
            e = e + 2;
    }
}

```

BAŞARI SINAVI (Visual Basic)

Sevgili öğrenciler,

Aşağıdaki sınav sizin programlama başarınızı ölçmek amacıyla hazırlanmış ve 5 sorudan oluşmaktadır. Soruları cevaplandırırken bildiğiniz herhangi bir programlama dilini ve programlama yöntemini kullanabilirsiniz. Bu sınav sizin mevcut dersleriniz için herhangi bir ölçme niteliği taşımamaktadır ve notlarınızı etkilemeyecektir. Bu yüzden sınavı cevaplandırırken bireysel olarak yardım olmadan cevaplandırınız.

Soru 1: Aşağıdaki kavramları açıklayınız.

Programlama:

Değişken:

Döngü:

Karar Yapısı:

Dizi:

Soru 2a: Aşağıda verilen programı for döngü yapısını kullanarak tekrar yazınız.

Örnek Program	For döngüsüyle yazılmış hali
<pre>Dim k As Integer = 1 Do ListBox1.Items.Add(k) k = k + 1 Loop While k < 50</pre>	

Soru 2b : Aşağıda switch – case yapısıyla oluşturulmuş programı if – else if – else yapısını kullanarak tekrar yazınız.

Switch – Case	If – else if – else yapısıyla yazılmış hali
<pre>Select Case k Case 1 TextBox1.Text = "Bir" Case 2 TextBox1.Text = "İki" Case Else TextBox1.Text = "Tanımsız Sayı" End Select</pre>	

Soru 3: Integer (tamsayı) türünde tanımlanmış 15 elemanlı bir tek boyutlu diziye rastgele sayılar yerleştirilmiştir. Bu sayıların en büyüğü ve en küçüğü arasındaki farkı kullanıcıya bildirecek olan programı yazınız (Kullanılacak olan dizinin önceden oluşturulduğunu var sayarak programınızı oluşturunuz).

Soru 4: Kullanıcıdan 20 tane sayı alarak bu sayılardan kaç tanesinin 50'den büyük, kaç tanesinin 50'ye eşit ve kaç tanesinin 50'den küçük olduğunu bularak, bu sayıların ortalamasını kullanıcıya bildirecek olan programı yazınız.

Soru 5: Aşağıdaki program çalıştırıldığında a, b, c, d, e, f değişkenlerinin alacağı son değerleri yazınız.

Alınacak en son değerleri buraya yazınız.

a	b	c	d	e	f

Sub Main()

Dim a = 2, b = 2, d = 20, e = 1, f = 1, c As Integer

If a = 1 Then

a = a + 1

End If

If a = 2 Then

a = a + 2

End If

If a = 3 Then

a = a + 3

End If

If a = 4 Then

a = a + 4

End If

Select Case b

Case 1

b = b + 1

Case 2

b = b + 2

Case 3

b = b + 3

Case Else

b = b + 4

End Select

c = 11 / Int(8 / 3)

For e = 1 To 4

d = d - e

Next e

While f < 8

f = f + 1

If f < 4 Then

e = e + f

Else

e = e + 2

End If

End While

End Sub

Ek 2 Başarı Testi Değerlendirme Ölçeği

BAŞARI SINAVI (C#)

Soru 1: Aşağıdaki kavramları açıklayınız.

Programlama : Programlama her hangi bir **problemin çözümünün programlama dili** kullanılarak yazılan **kod satırlarına** verilen isimdir.

Değişken: a) Değişkenler, **girdiğimiz değerleri** alan veya programın çalışmasıyla bazı **değerlerin atandığı, veri tutucularıdır.**

Döngü : Bir programda **işlemleri tekrar etmek** için kullanılan yapılardır.

Karar Yapısı : Program içerisinde, programın **ne şekilde devam edeceğine** karar vermek amacıyla kullanılan yapılardır.

Puanlama: Her kavram 4 puan değerindedir. Kavramlar cevaplanırken koyu renkli ifadeler tanım içerisinde geçiyor ise 4 tam puan alır. Kavramların bir veya bir kaç tanımında yer almıyorsa puan verilmez.

Soru 2a: Aşağıda verilen programı for döngü yapısını kullanarak tekrar yazınız.

Örnek Program	For döngüsüyle yazılmış hali (cevap)
<pre>int k = 1; do { listBox1.Items.Add(k); k = k + 1; } while (k < 50);</pre>	<pre>int k = 1; for (int k = 0; k < 50; k++) { listBox1.Items.Add(k); }</pre>

Puanlama:

- k değişkeninin doğru biçimde tanımlanması (2 puan)
- for döngü yapısının doğru şekilde oluşturulması (5 puan)
- listbox'a atamanın yapılması (3 puan)

Soru 2b : Aşağıda switch – case yapısıyla oluşturulmuş programı if – else if – else yapısını kullanarak tekrar yazınız.

Switch – Case	If – else if – else yapısıyla yazılmış hali
<pre>switch (k) { case 1: textBox1.Text = "Bir"; break; case 2: textBox1.Text = "İki"; break; default: textBox1.Text= "Tanımsız sayı";</pre>	<pre>if(k==1) textBox1.Text = "Bir"; else if(k==2) textBox1.Text = "İki"; else textBox1.Text = "Tanımsız sayı";</pre>

<code>break;</code>	
<code>}</code>	

- Puanlama:** - Syntax'ın doğru olması (2 puan)
 - if yapısının doğru oluşturulması (3 puan)
 - else - if yapısının doğru oluşturulması (3 puan)
 - else yapısının doğru oluşturulması (1 puan)

Soru 3: Integer (tamsayı) türünde tanımlanmış 15 elemanlı bir tek boyutlu diziye rastgele sayılar yerleştirilmiştir. Bu sayıların en büyüğü ve en küçüğü arasındaki farkı kullanıcıya bildirecek olan programı yazınız (Kullanılacak olan dizinin önceden oluşturulduğunu var sayarak programınızı oluşturunuz).

```
int[] dizi = new int[15];
int enBuyuk = 0;
int enKucuk = 0;
for (int i = 0; i < dizi.Length; i++)
{
    if (dizi[i] > enBuyuk)
        enBuyuk = dizi[i];
    if (dizi[i] < enKucuk)
        enKucuk = dizi[i];
}
Console.WriteLine("Fark = " + (enBuyuk - enKucuk));
```

- Puanlama:** - dizi değişkenin doğru şekilde tanımlanması (3 puan)
 - programda kullanılacak değişkenlerin tanımlanması ve değer atamaları (4 puan)
 - döngü yapısının oluşturulması (5 puan)
 - döngü içerisindeki karşılaştırmaların yapılması (5 puan)
 - sonucun ekrana yazdırılması (3 puan)

Soru 4: Kullanıcıdan 20 tane sayı alarak bu sayılardan kaç tanesinin 50'den büyük, kaç tanesinin 50'ye eşit ve kaç tanesinin 50'den küçük olduğunu bularak, bu sayıların ortalamasını kullanıcıya bildirecek olan programı yazınız.

```
int buyuk = 0, kucuk = 0, esit = 0;
for (int i = 0; i < 20 ; i++)
{
    Console.WriteLine("Sayı = ");
    int sayi = Convert.ToInt32(Console.ReadLine());
    if (sayi > 50)
        buyuk++;
    else if (sayi < 50)
        kucuk++;
    else
        esit++;
}
Console.WriteLine("50den büyük sayıların sayısı = " + buyuk);
Console.WriteLine("50den küçük sayıların sayısı = " + kucuk);
Console.WriteLine("50ye eşit sayıların sayısı = " + esit);
```

- Puanlama:** - programda kullanılacak değişkenlerin tanımlanması ve değer atamaları (4 puan)
 - döngü yapısının oluşturulması (5 puan)
 - döngü içerisinde değerlerin alınması (4 puan)

- döngü içerisindeki karşılaştırmaların yapılması (5 puan)
- sonucun ekrana yazdırılması (2 puan)

Soru 5: Aşağıdaki program çalıştırıldığında a, b, c, d, e, f değişkenlerinin alacağı son değerleri yazınız.

Alınacak en son değerleri buraya yazınız.

a	b	c	d	e	f
8	4	5	14	15	8

- Puanlama:** - 1 doğru 3puan
- 2 doğru 6 puan
- 3 doğru 10 puan
- 4 doğru 13 puan
- 5 doğru 16 puan
- 6 doğru 20 puan