

**PROGRAMLAMA PERFORMANSINI ETKİLEYEN
FAKTÖRLERİN BİLİŞSEL TABANLI BİREYSEL
FARKLILIKLAR TEMELİNDE MODELLENMESİ**

**MODELING THE INFLUENCE OF COGNITIVE BASED
INDIVIDUAL DIFFERENCES ON PROGRAMMING
PERFORMANCE**

SACİDE GÜZİN MAZMAN

PROF. DR. ARİF ALTUN

Tez Danışmanı

Hacettepe Üniversitesi
Lisansüstü Eğitim-Öğretim ve Sınav Yönetmenliğinin
Bilgisayar ve Öğretim Teknolojileri Eğitim Anabilim Dalı için Öngördüğü
DOKTORA TEZİ olarak hazırlanmıştır.

2013

SACİDE GÜZİN MAZMAN'ın hazırladığı “**Programlama Performansını Etkileyen Faktörlerin Bilişsel Tabanlı Bireysel Farklılıklar Temelinde Modellenmesi**” adlı bu çalışma aşağıdaki jüri tarafından **BİLGİSAYAR VE ÖĞRETİM TEKNOLOJİLERİ EĞİTİMİ ANABİLİM DALI**'nda **DOKTORA TEZİ** olarak kabul edilmiştir.

Başkan:

(Prof. Dr. Petek AŞKAR)

Danışman:

(Prof. Dr. Arif ALTUN)

Üye:

(Prof. Dr. Banu CANGÖZ)

Üye:

(Prof. Dr. Zahide YILDIRIM)

Üye:

(Doç. Dr. Hakan TÜZÜN)

Bu tez Hacettepe Üniversitesi Eğitim Bilimleri Enstitüsü tarafından **DOKTORA TEZİ** olarak onaylanmıştır.

Prof. Dr. Berrin AKMAN.

Eğitim Bilimleri Enstitüsü Müdürü

Bu tez Türkiye Bilimsel ve Teknik Arařtırmalar Kurumu (TÜBİTAK) tarafından 112K615 no'lu proje kapsamında kısmen desteklenmiştir.

ETİK

Hacettepe Üniversitesi Eğitim Bilimleri Enstitüsü, tez yazım kurallarına uygun olarak hazırladığım bu tez çalışmasında,

- tez içindeki bütün bilgi ve belgeleri akademik kurallar çerçevesinde elde ettiğimi,
- görsel, işitsel ve yazılı tüm bilgi ve sonuçları bilimsel ahlak kurallarına uygun olarak sunduğumu,
- başkalarının eserlerinden yararlanılması durumunda ilgili eserlere bilimsel normlara uygun olarak atıfta bulunduğumu,
- atıfta bulunduğum eserlerin tümünü kaynak olarak gösterdiğimi,
- kullanılan verilerde herhangi bir tahrifat yapmadığımı,
- ve bu tezin herhangi bir bölümünü bu üniversitede veya başka bir üniversitede başka bir tez çalışması olarak sunmadığımı

beyan ederim.

25/10/2013

Sacide Güzin MAZMAN

ÖZET

PROGRAMLAMA PERFORMANSINI ETKİLEYEN FAKTÖRLERİN BİLİŞSEL TABANLI BİREYSEL FARKLILIKLAR TEMELİNDE MODELLENMESİ

Sacide Güzin MAZMAN

Doktora, Bilgisayar ve Öğretim Teknolojileri Eğitimi Bölümü

Tez Danışmanı: Prof. Dr. Arif ALTUN

Ekim 2013, 105 Sayfa

Bu çalışmanın amacı programlama performansına etki eden bilişsel faktörlere ilişkin bir model geliştirerek, bu performans üzerindeki bilişsel tabanlı bireysel farklılıkların etkisini diğer demografik özelliklerin etkisi ile birlikte ortaya koymaktır. Çalışmada bilişsel tabanlı bireysel farklılıklardan uzamsal beceri ve çalışma belleği ele alınarak, bu becerilerin öz yeterlilik, cinsiyet, ön deneyim ve üniversite değişkenleri ile birlikte programlama performansının ne kadarını yordadığı ve bu sürece anlamlı katkıda bulunan değişkenlerin önem sıraları ortaya konulmuştur.

Çalışma üç farklı üniversitedeki Bilgisayar ve Öğretim Teknolojileri Öğretmenliği (BÖTE) bölümü 2. sınıf öğrencilerinden “Programlama Dilleri-I” dersine kaydolan toplam 129 lisans öğrencisi ile yürütülmüştür. Çalışmada uzamsal beceriler zihinsel döndürme ve uzamsal yönelim becerilerin ölçülmesi ile elde edilirken, çalışma belleği ise görsel-uzamsal çalışma belleği ve sözel çalışma belleği olmak üzere iki alt boyutun ölçülmesi ile elde edilmiştir. Uzamsal yönelim, görsel uzamsal bellek ve zihinsel döndürme E-prime 2.0 yazılımı ile araştırmacılar tarafından geliştirilmiş bilgisayar tabanlı testler ile, sözel bellek ise Çevik (2012) tarafından Türkçeye derlenmiş “Brain Workshop” adlı n-geri görev yazılımı ile toplanmıştır. Bilişsel testlerin puanlanmasında doğruluk ve tepki sürelerini birlikte işe koşan bir etkililik puan indeksi hesaplanmıştır. Öz yeterlilik algısının belirlenmesinde, “Programlamaya İlişkin Öz Yeterlilik Algısı Ölçeği (Altun ve Mazman, 2012)” kullanılmıştır. Bağımlı değişken olan programlama performansı için ise öğrencilerin programlama dersine ilişkin yılsonu başarı notları alınmıştır. Veri toplama tüm katılımcılarda bireysel olarak, aynı araştırmacı ile gerçekleştirilmiştir.

Verilerin analizinde veri madenciliğine dayalı doğrusallık varsayımı gözetmeyen, tahminleyici non-parametrik yöntemlerden “Artırılmış Regresyon Ağaçları” ve

“Rastgele Ormanlar” teknikleri kullanılmıştır. Programlama performansının bilişsel ve bilişsel olmayan değişkenlerle tahmin edilmesine ilişkin kurulan model her iki yöntemle de analiz edilerek değişkenlerin önem sıraları ve bağımlı değişkenin tahmin edilme oranı karşılaştırılmıştır.

Artırılmış Regresyon Ağacı analizi sonuçlarına sekiz değişken (zihinsel döndürme, uzamsal yönelim, görsel-uzamsal bellek, sözel bellek, programlamaya ilişkin öz yeterlilik algısı, ön deneyim, üniversite ve cinsiyet) birlikte programlama performansının %65,’ini tahmin etmiştir. Bu değişkenler arasından önem ağırlığı 0,4’ün üzerinde bulunarak modele anlamlı katkıda bulunan değişkenlerin sıralaması ise şu şekildedir; görelî önem ağırlığı en yüksek değişken uzamsal yönelim becerisi olurken (önem derecesi= 1), uzamsal bellek (önem derecesi= 0.965) ikinci sırada, öz yeterlilik algısı (önem derecesi= 0.929) üçüncü sırada, zihinsel döndürme becerisi (önem derecesi= 0,876) dördüncü sırada yer almakta olup, bunu sözel bellek (önem derecesi= 0,830) izlemiş ve altıncı sırada ise üniversite (0,809) değişkeni yer almıştır. Cinsiyet (önem derecesi= 0,365) ve ön deneyimin (önem derecesi= 0,165) ise programlama performansını tahmin etmedeki modele katkıları anlamlı olarak bulunmamıştır.

Rastgele orman analizi sonuçlarına göre sekiz değişken birlikte programlama performansının %60’ını tahmin etmiştir. Rastgele ormanlar tekniğı ile kurulan modelde önem ağırlı 0.4’ün üzerinde olarak modele anlamlı katkıda bulunan değişkenler şu şekildedir; görelî önem ağırlığı en yüksek, üniversite değişkeni olurken (önem derecesi= 1), özyeterlilik (önem derecesi= 0.952) ikinci sırada, zihinsel döndürme (önem derecesi= 0,483) üçüncü sırada, uzamsal bellek (önem derecesi= 0,474) dördüncü sırada yer almış, bunu uzamsal yönelim becerisi (önem derecesi= 0,470) izlemekte olup, altıncı sırada ise sözel bellek (önem derecesi= 0,416) değişkeni yer almıştır. Ön deneyim (önem derecesi= 0,215) ve cinsiyet (önem derecesi= 0,186) değişkenleri ise programlama performansını tahmin etmedeki modele anlamlı katkıda bulunmamıştır. Modele anlamlı katkıda bulunan değişkenler her iki yöntemde de aynı olmakla birlikte, iki tekniğın kullandığı algoritmaların farklılaşması ile değişkenlerin önem sıraları da farklılaşmaktadır.

Anahtar Sözcükler: Programlama performansı, bilişsel beceriler, bireysel farklılıklar, artırılmış regresyon ağacı, rastgele ormanlar

ABSTRACT
MODELING THE INFLUENCE OF COGNITIVE BASED INDIVIDUAL
DIFFERENCES ON PROGRAMMING PERFORMANCE

Sacide Güzin MAZMAN

Doctor of Philosophy, Department of Computer Education and Instructional
Technology

Supervisor: Prof. Dr. Arif ALTUN

October 2013, 105 Pages

This study is aimed to determine the effects of cognitive based individual differences along with the other demographic characteristics on undergraduate students' programming performance and to model the programming performance based on the cognitive factors related to programming. Considering the spatial ability and the working memory variables as cognitive and self-efficacy, gender, prior experience and university variables as non-cognitive individual differences, this study is designed to address how much of variance of programming performance is predicted and what the importance rank of those variables are.

The study group included 129 sophomores who were enrolled in Programming Language-I course at the Computer Education and Instructional Technologies Department in three different universities located in Ankara. Spatial ability factor included spatial orientation and mental rotation as sub-factors, while working memory had visuo-spatial working memory and verbal working memory sub-factors. Spatial orientation, mental rotation and visual-spatial working memory data were collected via computerized tests which were developed through E-Prime 2.0 software by researchers. Verbal memory data was collected through n-back task translated into Turkish by Çevik (2012). An efficiency score index was calculated which is comprised of both accuracy and reaction times. "Programming Self Efficacy Scale" (Altun and Mazman, 2012) was used to measure self-efficacy. Students' programming course grades at the end of the year was taken as dependent variable. Data collection process was conducted individually for each participant and by the same researcher.

"Boosted Regression Trees" and "Random Forests" are used in data analyses which are non-parametric predictive data mining methods without linearity assumption. The

model which was developed to predict programming performance based on cognitive and non-cognitive variables, tested by both of the methods and the importance ranks of variables and proportion of predicted variance of programming performance was compared.

The result of Boosted Regression Tree revealed out that, eight variables (mental rotation, visuo-spatial working memory, verbal memory, programming self-efficacy, prior experiences, university and gender) predicted 65% of the programming performance. The variables, of which importance value was found higher than 0.4 and contributed the model significantly ranked as; having the highest relative importance was spatial orientation ability (importance=1), visuo-spatial working memory was at the second rank (importance=0,965), programming self-efficacy was at the third rank (importance=0,929), mental rotation ability was at the fourth rank (importance=0,876) followed by verbal working memory which was at the fifth rank (importance=0,830) and the university variable was at the sixth rank (importance=0,809). Neither gender (importance=0,365), nor prior experience (importance=0,165) were found to be significant predictors in the model.

The result of Random Forests analyses showed that eight variables predicted 60% of the programming performance. The variables, of which importance value was found higher than 0.4 and contributed to the model significantly ranked as; having the highest relative importance was university (importance=1), programming self efficacy was at the second rank (importance=0,952), mental rotation was at the third rank (importance=0,483), visuo-spatial working memory was at the fourth rank (importance=0,474) followed by spatial orientation ability which was at the fifth rank (importance=0,470) and the verbal memory variable was at the sixth rank (importance=0,416). Neither gender (importance=0,186), nor prior experience (importance=0,215), were found to be significant predictors in the model. While the variables were the same in both of the models to predict programming performance, the importance rank of the variables were slightly differed due to the algorithms used in techniques.

Keywords: Programming performance, cognitive abilities, individual differences, boosted regression tree, random forests

TEŞEKKÜR

Hacettepe'ye geldiğimden beri aldığım derslerde, yaptığımız çalışmalarda ve en önemlisi de tez sürecimde hiçbir zaman bildiklerini esirgemeyen, her zaman yol gösteren, her türlü zorlukla başa çıkmada beni motive eden, zor anlarımda desteğini hissettiren ve hiçbir zaman tükenmeyen alternatif fikirleriyle problemleri çözen, akademisyenlik adına çok şey öğrendiğim değerli danışmanım ve hocam Prof. Dr. Arif Altun'a sonsuz teşekkürlerimi sunarım.

Tez süreci boyunca her daim derin bilgileri ile tezime destek olan ve yön veren, her tez izleme toplantısından motive olmuş şekilde çıkmamı sağlayan değerli hocalarım Prof. Dr. Banu Cangöz ve Prof. Dr. Zahide Yıldırım'a katkılarından dolayı teşekkür ederim. Çalışmamı inceleyerek değerli katkılarını esirgemeyen ve önerileri ile tezin şekillenmesini sağlayan, sundukları farklı bakış açıları ile tezi zenginleştiren değerleri hocalarım Prof. Dr. Petek Aşkar'a ve Doç. Dr. Hakan Tüzün'e teşekkürlerimi sunarım.

Veri toplama sürecinde verdikleri destekle ve ev sahiplikleriyle süreci daha kolay geçirmemi sağlayan Gazi Üniversitesi'nden Dr. Sibel Somyürek'e ve Doç. Dr. Tolga Güyer'e, Ankara Üniversitesi'nden arkadaşım Funda Kamel Erdoğan'a, değerli hocam Prof. Dr. Nurettin Şimşek'e ve Yrd. Doç. Dr. Özlem Çakır'a, uygulama yaptığım ders kapsamında istediğim bilgileri esirgemeyen Erinç Karataş ve Salih Demir'e ve uygulamaya sabırla ve özveriyle katılan tüm öğrencilere teşekkür ederim.

Tez sürecinde analizlerde kullandığım programları öğrenmem konusunda benden yardımlarını esirgemeyen ve her başım sıkıştığımda sabırla beni cevaplayan değerli arkadaşım Erdal Coşgun'a teşekkür ederim. Yine programlar konusunda sık sık kapısını çalarak fikirlerini aldığım arkadaşım Gökhan Akçapınar'a teşekkür ederim.

Tez sürecinde beni her daim motive eden bana hep destek olan, sık sık birbirimize dert yanarak yine birbirimizi avuttuğumuz kader arkadaşım, canım dostum Hale Ilgaz'a teşekkür ederim. Tez sürecinde desteğini hep yanımda hissettiğim beni yüreklendiren bana hep inanan değerli arkadaşlarım Yasemin Demiraslan Çevik, Filiz Kuşkaya Mumcu, Tülin Haşlaman, Fatih Özdiñç ve Esra Telli'ye teşekkür ederim. Ayrıca tüm tez yazma sürecime şahit olan, bana destek olan oda arkadaşlarım Ferhat Kadir Pala'ya ve Turgay Baş'a, ve tüm araştırma görevlisi arkadaşlarıma teşekkür ederim.

Doktora öğrenimim boyunca bana maddi destek sunan Türkiye Bilimsel ve Teknolojik Araştırma Kurumu'na (TÜBİTAK) teşekkür ederim

Hayatımın her anında bana olan inançlarını ve desteklerini sevgileriyle birlikte hissettiren, koşulsuz ve karşılıksız emeklerini esirgemeyen başta canım kardeşim Ahmet Salih Mazman'a ve aileme sonsuz teşekkürlerimi sunuyorum. İyi ki varsınız.

İÇİNDEKİLER

ÖZET	i
ABSTRACT.....	iii
TEŞEKKÜR.....	v
ÇİZELGELER DİZİNİ	ix
ŞEKİLLER DİZİNİ	x
KISALTMALAR	xi
1. GİRİŞ.....	1
1.1 Problem Durumu	1
1.2 Programlama Performansı	4
1.3 Bilişsel Bireysel Farklılıklar	7
1.3.1 Uzamsal Beceri	8
1.3.2 Çalışma Belleği	10
1.4 Programlamaya İlişkin Özyeterlilik Algısı.....	12
1.5 Programlama Ön Deneyimi	13
1.6 Programlama Performansı ve Cinsiyet.....	14
1.7 Çalışmanın Önemi.....	15
1.8 Çalışmanın Amacı	16
1.9 Araştırma Soruları	16
1.10 Çalışmanın Sınırlılıkları	17
1.11 Tanımlar	17
2. İLGİLİ ÇALIŞMALAR.....	19

2.1	Çalışma Belleği ve Programlama Performansına İlişkin Çalışmalar.....	20
2.2	Uzamsal Beceriler ve Programlama Performansına İlişkin Çalışmalar	24
2.3	Bilişsel Olmayan Faktörler ve Programlama Performansına İlişkin Çalışmalar (Öz yeterlilik, Cinsiyet ve Ön deneyim).....	27
3.	YÖNTEM	34
3.1	Araştırma Modeli	34
3.2	Çalışma Grubu	34
3.3	Veri Toplama Süreci.....	35
3.4	Veri Toplama Araçları.....	38
3.4.1	Uzamsal Yönelim Testi.....	38
3.4.2	Görsel-Uzamsal Bellek- Sayı Döndürme Testi	40
3.4.3	Sözel Çalışma Belleği Testi.....	43
3.4.4	Programlamaya İlişkin Öz Yeterlilik Ölçeği.....	45
3.4.5	Programlama Performansı	46
3.5	Ölçme Araçlarının Puanlanması.....	46
3.6	Verilerin Analizi	47
3.6.1	Artırılmış Regresyon Ağaçları (Boosted Regression Trees).....	48
3.6.2	Rastgele Ormanlar (Random Forests)	51
4.	BULGULAR	56
4.1	Bireylerin Programlama Performansı, Öz Yeterlilik Algıları, Uzamsal Becerileri, Çalışma Belleği Düzeyleri ve Aralarındaki İlişkiye İlişkin Bulgular	56
4.2	Programlama Performansının Bireylerin Ön Deneyimleri, Cinsiyet ve Üniversitelerine Göre İncelenmesine İlişkin Bulgular.....	59

4.3	Programlama Performansının Yordanmasına İlişkin Kurulan Modelin Ağaç Tabanlı Yöntemlerle Test Edilmesine İlişkin Bulgular	60
4.3.1	Artırılmış Regresyon Ağacı Sonuçlarına İlişkin Bulgular	60
4.3.2	Rastgele Orman (Random Forest) Analizine İlişkin Bulgular.....	65
5.	TARTIŞMA VE SONUÇ	70
6.	ÖNERİLER	80
6.1	Araştırmaya Dönük Öneriler	80
6.2	Uygulamaya Dönük Öneriler	82
	KAYNAKÇA.....	83
	EKLER	94
	ÖZGEÇMİŞ	105

ÇİZELGELER DİZİNİ

Çizelge 2.1. İlgili çalışmalar için taranan kaynaklar ve anahtar kelimeleri.....	20
Çizelge 3.1. Çalışma grubuna ait demografik bilgiler	35
Çizelge 3.2. Rastgele Ormanlar ve Artırılmış Regresyon Ağacı teknikleri karşılaştırma tablosu	55
Çizelge 4.1. Programlama performansı, uzamsal beceri, çalışma belleği ve öz yeterlilik algısına ilişkin betimsel istatistikler	57
Çizelge 4.2. Değişkenler arasındaki korelasyon matrisi.....	57
Çizelge 4.3. Cinsiyet ve ön deneyime göre programlama performansına ilişkin t testi sonuçları.....	59
Çizelge 4.4 Üniversitelere göre programlama performansına ilişkin tek yönlü varyans analizi sonuçları.....	60
Çizelge 4.5. Artırılmış Regresyon Ağacı analizi açıklayıcı değişkenlere ilişkin değişken sıraları ve önem ağırlıkları.....	63
Çizelge 4.6 Rastgele Orman analizi açıklayıcı değişkenlere ilişkin değişken sıraları ve önem ağırlıkları	67

ŞEKİLLER DİZİNİ

Şekil 3.1. Uygulama sürecine ilişkin akış diyagramı	37
Şekil 3.2. Uzamsal yönelim testi örnek ekran görüntüsü	40
Şekil 3.3. Görsel-Uzamsal bellek testi örnek sayı görüntüleri	42
Şekil 3.4 Görsel uzamsal bellek testi örnek soru ekranları	43
Şekil 3.5. Sözel çalışma belleği testi örnek ekran görüntüsü	44
Şekil 4.1 Bağımlı değişken ve sürekli bağımsız değişkenlere ilişkin saçılım grafikleri	58
Şekil 4.2 Artırılmış Regresyon Ağacı analizi optimizasyon grafiği.....	62
Şekil 4.3. Artırılmış Regresyon Ağacı analizi açıklayıcı değişkenlere ilişkin görelî önem katsayıları grafiği	63
Şekil 4.4 Artırılmış Regresyon Ağacı analizi saçılım grafiği	64
Şekil 4.5 Ortalama hatalar karesi tahmin grafiği	66
Şekil 4.6 Rastgele Orman analizi açıklayıcı değişkenlere ilişkin görelî önem katsayıları grafiği	67
Şekil 4.7 Rastgele Orman analizi saçılım grafiği.....	69

KISALTMALAR

SS: Standart Sapma

SH: Standart Hata

Sd: Serbestlik derecesi

N: Sayı

\bar{X} : Aritmetik Ortalama

r: Korelasyon katsayısı

p: Anlamlılık düzeyi

BÖTE: Bilgisayar ve Öğretim Teknolojileri Eğitimi

Oob: Out of bag

lr: (learning rate) Öğrenme oranı-küçülme parametresi

nat: (number of additive terms) Toplanabilir terim sayısı

nmn: (maximum number of nodes) Maksimum düğüm sayısı

1. GİRİŞ

Bu kısımda çalışmanın problem durumu, çalışmanın amacı, çalışmanın önemi ve araştırma sorularına yer verilecektir.

1.1 Problem Durumu

Programlama performansının bireyler açısından oldukça önemli bir beceri olduğu bilinmekte ve yapılan çalışmalarda da bu performansa etki eden faktörler ortaya konulmaya çalışılmaktadır. Nitekim hangi bireylerin programlamada başarılı olduklarına hangilerinin başarısız olduklarına, programlama performansına etki eden bireysel ve diğer faktörlerin etkisine ilişkin farklı bulgular söz konusudur.

Programlama karmaşık bilişsel davranışların da en önemlilerinden birisi olarak görülmekte ve iyi bir programcının hem içerik hem de uygulama bilgisine sahip olduğu ifade edilmektedir (Bergersen ve Gustafsson, 2011). Diğer yandan öğrencilerin programlama öğrenmeye başlamadan önce bazı temel becerilere sahip olması gerektiği ifade edilmekte, fakat bu becerilerin neler olması gerektiğine ilişkin ise farklı çalışmalarda farklı bulguların olduğu görülmektedir (Holvikivi, 2010).

Programlama performansını belirleyen ya da bu performans ile ilişkili olan değişkenleri bulmaya dönük olarak yapılan çalışmalar 1950'lerden günümüze kadar devam etmektedir. Bu konuda yapılan çalışmalar ele aldıkları değişkenler, ölçme araçları, yöntem ya da bağlam bakımından değişse de, temelde iyi programcı olmanın nedenlerini ya da bunu etkileyen faktörleri ve bu faktörler arasındaki ilişkiyi bulmaya dönüktür. Bu çalışma kapsamında erişilebilen ilk çalışmalardan birisi olan, Rowan (1956) "*Psychological Tests and Selection of Computer Programmers*" başlıklı çalışmada bilgisayar programcılığı için personel seçiminde kullanılan testleri incelemiş ve bu testlerin kullanımında deneysel geçerliliğin önemi ile birlikte objektif değerlendirmenin önemini vurgulamıştır. Daha sonraki çalışmaların bazılarında programlama performansı ile matematik becerisi, kişilik özellikleri ve sözel beceri, akıl yürütme, eleştirel düşünme gibi çeşitli yetenekler arasındaki ilişki incelenirken (Alspaugh, 1972; Austin, 1987), bazı çalışmalarda programlama performansına etki edebilecek olası faktörlerle modeller kurularak test edilmiştir (Glorfeld ve Fowler, 1982). Bunlar dışında bireylerin farklı programlama dillerindeki başarısını etkileyen faktörleri inceleyen

(Capstick, Gordon, ve Salvadori, 1975), ayrıca programlama performansının etkili bir şekilde geliştirilmesi ve transferi için öğretim tasarımı ilkeleri (Merrienboer ve Paas, 1990) ve çeşitli öğrenme yaklaşımlarına yönelik hazırlanmış programlama öğretimi ortamları öneren çalışmalar (Emurian, Hu, Wang, ve Durham, 2000) da bulunmaktadır. Son zamanlarda yapılan çalışmalara bakıldığında ise halen benzer şekilde demografik özellikler (cinsiyet, yaş, lise türü vb.), ön deneyim, kişilik özellikleri, sayısal ve sözel yetenek testleri, matematik becerisi, problem çözme becerileri gibi çeşitli değişkenlerle programlama performansı arasındaki ilişkinin ortaya konularak programlama performansının yordanmasına ilişkin araştırmalar yürütülmektedir (Bergersen ve Gustafsson, 2011; Martin ve Beena, 2012; Mauai ve Temese, 2012; Yurdugül ve Aşkar, 2013).

Geçmişten günümüze devam eden bu çalışmalar, her ne kadar yapılan araştırmalarda programlama performansına ilişkin bireyler arası farklılıkların devam eden bir şekilde çalışılmış olsa da, bu farklılıkların yeteri kadar nicel olarak belirtilememiş olduğunu (Bergersen ve Gustafsson, 2011) göstermektedir. Bunun nedeninin ise programlama performansına ilişkin yapı geçerliliği sorununun henüz çözülememiş olduğu ifade edilmektedir (Hannay, Arisholm, Engvik, ve Sjoberg, 2010). Ayrıca programlama konusunu bazılarını eğlenceli kılan ve motive eden bazılarını ise eziyet haline getiren faktörlerin henüz net olarak bilinmemekte olduğu (Ramalingam, LaBelle, ve Wiedenbeck, 2004), bireyler tarafından genellikle zor olarak kabul edilen programlamanın, genel faktörler dışında doğrudan programlama performansına özgün bağlamlarda ele alınması gerektiği (Zin, Idris, ve Subramaniam, 2006) ileri sürülmektedir. Irons (1982) ise bireylerin bir programlama ortamına ilişkin bilgiyi işleme özellikleri ve bilişsel becerileri gibi temel psikolojik süreçlerinin değerlendirilmesi gerektiğini, fakat araştırmalarda programlama performansı üzerindeki bilişsel becerilerin etkisinin çoğu zaman ihmal edildiğini vurgulamıştır.

Programlama eğitimi tüm dünyada olduğu gibi ülkemizde de genel olarak lisans düzeyinde verilmekte iken, yazılımı sevdirmek ve lisans seviyesine gelmeden önce yazılım mantığını öğretmek amacıyla lisans öncesi eğitimde programlama içeren yazılım dersleri müfredatta yerini almıştır (Karabaş ve Güneş, 2013). İlköğretim ve ortaöğretim seviyesinde programlama eğitimi veren eğiticiler ise bilim teknolojileri öğretmenleri olup, bu öğretmenleri yetiştiren en önemli bölümlerden birisi ise

Bilgisayar ve Öğretim Teknolojileri Eğitimi bölümleridir. Bilgisayar ve Öğretim Teknolojileri Eğitimi Bölümü'nde (BÖTE) programlama derslerinin hemen hemen her yarıyıl temel ders olarak, seçmeli ve zorunlu verildiği, ayrıca belirli derslerin içeriğinde de entegre olarak da programlama bilgisi gereksiniminin yer aldığı görülmektedir. BÖTE bölümlerinde III. yarıyıldan itibaren “Programlama Dilleri-I”, IV. yarıyıldan itibaren “Programlama Dilleri-II”, V. yarıyıldan itibaren “İnternet Tabanlı Programlama” zorunlu olarak yer alırken, “Nesneye Yönelik Programlama” ve “Görsel Programlama” ise birçok üniversitenin öğretim programında seçmeli olarak verilmektedir. Programlama dersleri dışında veri tabanı, işletim sistemleri ve grafik canlandırma gibi derslerin içeriğinde ise yine PHP, ASP, Action Script programlama ve Java programlama bilgilerinden etkin olarak yararlanılmaktadır. Bu şekilde programlama performansının BÖTE alanında da oldukça önemli olduğu ifade edilebilir.

BÖTE bölümleri özellikle meslek lisesi, teknik liseler ve liseler olmak üzere farklı lise türlerinden öğrenci alan bir bölüm olmasıyla bilgisayar dersleri bakımından farklı alt yapıya sahip öğrenciler bir araya gelmektedir. Diğer yandan programlama derslerine ilişkin başarının BÖTE bölümlerinde de düşük olduğu görülmektedir. BÖTE bölümünde “Programlama Dilleri I” dersi kapsamında yapılan bir çalışmada, dersi o dönem alan toplam 81 öğrenciden 45'nin derse ilk defa kaydolarken, 36'sının ise dersi alttan aldığı yani programlama dersine kaydolmuş öğrencilerin yarısının bir önceki dönemden başarısız olarak dersi tekrarladığı ifade edilmiştir (Tüzün, 2007).

Erdoğan (2005) programlama başarısına ait yapılmış araştırmaların toplam varyansı açıklamada yetersiz olduğunu ifade etmiş ve bu alanda ülkemizde yapılmış olan ilk araştırma olarak ifade ettiği çalışmasında BÖTE Bölümü öğrencileri ile çalışarak programlama başarısına ilişkin varyansın %34,1'ini açıklamıştır. Bu da ülkemizde halen, programlama performansının açıklanmasına dönük farklı değişkenlerin ele alınarak açıklanan varyansın artırılmasına ihtiyaç olduğunu göstermektedir.

Programlama her bir alt görevi için oldukça karmaşık bilişsel süreçler gerektiren bir beceri olduğundan, bireylerin programlama için gerekli bilişsel becerilerinin bilinmesi ve bu becerilerin diğer faktörlerle ele alınarak ortak etkisinin ortaya

konulması gerekmektedir. Ülkemizde programlama performansını açıklayan faktörleri bütünsel bir bakış açısıyla belirlemeye dönük yapılan çalışmaların sınırlılığı dikkat çekmektedir. Var olan çalışmalarda ise bireylerin bilişsel bireysel farklılıklarından çok genel akademik başarı, bilgisayara karşı tutum, cinsiyet, lise türü gibi bilişsel faktörlerden çok bilişsel olmayan faktörler ve demografik özelliklerin ele alındığı (Erdoğan, 2005), bilişsel faktörlerin ise problem çözme becerileri, yaratıcılık ve genel yetenek ölçümleri ile sınırlandırıldığı (Erdoğan, Aydın, ve Kabaca, 2008) görülmektedir. Bu nedenle programlama performansındaki bireysel farklılıkların ortaya konulmasında bilişsel farklılıkların temel alınarak, bu bilişsel farklılıkların bireysel olarak ölçümlendiği ve programlama sürecine etki eden bilişsel olmayan faktörlerle birlikte ele alınarak programlama performansının açıklanmasına ilişkin çalışmalara gereksinim olduğu görülmektedir.

1.2 Programlama Performansı

Programlama, alt görevlerinin her birisi farklı bir bilgi alanı ve farklı bilişsel süreçler gerektiren oldukça karmaşık bir süreç olup (Ambrosio, Costa, Almeida, Franco, ve Macedo, 2011) eğitim ve öğrenme açısından önemli olduğu varsayılan birçok beceri programlama öğretiminin doğasında yer almaktadır (Howard, 2002; Lehman, Bruning, ve Horn, 1983). Örneğin okuduğunu anlama, eleştirel akıl yürütme ve sistemik düşünme; problem tanımlamadaki bilişsel bileşenlere sahip olma, planlama ve çözüm üretme, yaratıcılık ve entelektüel merak, matematik becerisi ve durumsal akıl yürütme, prosedürel düşünme ve geçici akıl yürütme, analitik ve nicel akıl yürütme, farklı kaynaklardan yararlanabilme, yeni çözümler üretmede yaratıcı ve esnek olabilme bu becerilerden bazılarıdır (Ambrosio ve diğ., 2011; Lau ve Yuen, 2011).

Programlama okuryazarlığı kavramı da, 21. yüzyıl okuryazarlıkları olarak adlandırılan ve bireylerin elektronik ortamlarda okur yazar olarak nitelendirilebilmeleri için e-okuma, e-yazma ve e-metin gibi temel becerileri vurgulayan e-okuryazarlığın (Altun, 2003) ve teknoloji okur yazarlığının (Lau ve Yuen, 2011) bir boyutu olarak ele alınmaya başlamıştır (Prensky, 2008). Programlama okuryazarlığı, bilgisayar programı yazabilmenin yanında çağımızda bireylerin sahip olması gereken üst düzey düşünme becerilerini de kullanmayı gerektiren bir beceri olarak tanımlanmakta ve problemi anlama, uygulanabilir bir

algoritma oluřturma, bu algoritmaya uygun kod yazma ve kodlardaki hataları dzeltme gibi biliřsel becerileri gerektirdiđi ifade edilmektedir (Akıncı ve Tzn, 2012). UNESCO'nun raporunda da Bilgi ve İletiřim Teknolojilerini (BİT) okullarda kullanmaya iliřkin ařamalardan ilkinin BİT okuryazarlıđı oluřtururken, son ařamayı ise BİT alanında uzmanlařma yani temel ve ileri dzeyde programlama tekniklerini bilme oluřturmaktadır (UNESCO, 2002).

Diđer yandan programlama konusunda birok bireyin bařarisız olduđu (Ambrosio ve diđer., 2011; Mancy ve Reid, 2004; Mead ve diđer., 2006) ve programlama performansı yksek bireylerin ise genel olarak analitik dřnebilen, zeki ve entelektel bireyler olduđuna iliřkin genel yargılar sz konusudur (Byrne ve Lyons, 2001). Farklı alanlarda olduka bařarılı olabilen bireylerin programlama konusunda yetersiz olabildiđi de belirtilmiřtir (Byrne ve Lyons, 2001; Fincher ve diđer., 2005; Jones ve Burnett, 2008).

Programlama yeterlilikleri sz konusu olduđunda bilgi ve becerinin ayırt edilmesi gerektiđi, programlama bilgisinin tanımlar, gerekler, dil yapısı ve spesifik algoritmalarla oluřturulurken, programlama becerisinin ise bilgiyi kullanmadaki gerekli stratejilerden oluřtuđu belirtilmektedir (Caspersen, 2007). Her ne kadar bilgi program yazma iin gerekli olsa da programlamaya iliřkin bir uygulama iin yeterli bir n kořul olmayıp programın geliřtirilebilmesi iin bu bilginin iře kořulmasını sađlayacak becerilere ve stratejilere ihtiya duyulmaktadır. Programlamaya iliřkin bilgi her ne kadar tek olabilse de, bilginin kullanımını ortaya koyan beceriler ve iře kořulan stratejiler noktasında bireyler farklılařmakta ve bu noktada ise bireysel farklılıklar n plana ıkmaktadır.

Michelson (2004) programlama performansına katkıda bulunan ve bu becerisinin gstergesi olan faktrlere iliřkin yapılacak alıřma konuları iin řu nerilerde bulunmuřtur; 1) hem bařarılı hem de bařarisız đrencilerin beceri ve niteliklerine iliřkin geniř lekli niceliksel olarak alıřmalar 2) programlamayı farklı ortamlarda ve farklı yaklařımlarla alıřan gruplara iliřkin kk lekli boylamsal alıřmalar; 3) programlama đretimine iliřkin iyi ve kt uygulama rneklerinin tanımlanması ve incelenmesine iliřkin alıřmalar.

Yapılan alıřmalarda en ok ele alınan faktrlerin cinsiyet, kiřilik, zeka, bilgisayara karřı tutum, deneyim, rahatlık seviyesi, matematik arka planı, alınan dersler ve

oyun oynama vb. faktörler olduğu görülmektedir (Charlton ve Birkett, 1999; Wilson, 2002).

Alanyazında programlama performansına ilişkin yapılan araştırmaların bir boyutunda ise uzman ya da acemi olan bireylerin sahip olduğu özellikleri belirlemeye ya da uzman-acemi farklarını ortaya koymaya dönük çalışmalar bulunmaktadır (Vainio ve Sajaniemi, 2007). Bu araştırmalarda da benzer şekilde programlama konusunda uzman ve acemi olan bireyler ayrı ayrı incelenerek uzmanların özellikleri ve acemilerin özellikleri belirlenip, programlama performansına etki eden faktörler ortaya konulmuş, bu özelliklerden uzman ve acemileri ayıran en önemli faktörlerin ön deneyim, öz yeterlilik, bilginin organizasyonu, zihinsel modeller ve bilgisayara karşı ilgi olduğu bulunmuştur (Ramalingam ve diğ., 2004; Wiedenbeck, LaBelle, ve Kain, 2004; Wiedenbeck, Sun, ve Chintakovid, 2007).

Son zamanlarda ise programlama performansını belirlemeye yönelik olarak akademik arka plan ve psikolojik faktörlerle (Bergin ve Reilly, 2006), bilişsel, davranışsal ve tutumsal faktörlerle (deRaadt ve diğ., 2005; Simon, Cutts, ve diğ., 2006; Simon, Fincher, ve diğ., 2006), bilişsel beceriler ve ön bilgi ile (Bergersen ve Gustafsson, 2011), demografik faktörler, akademik beceri, öğretim ortamı ve zihinsel model faktörleri (Lau ve Yuen, 2011; Ramalingam ve diğ., 2004) süreçte etken rol oynayan değişkenleri modellenmesine dönük akımlar söz konusudur. Bu çalışmalarla programlama performansına ilişkin yapısal modeller kurularak, bu performansa etki eden faktörlerin doğrudan ve dolaylı etkileri birlikte incelenmekte ve aynı zamanda değişkenler arasındaki ilişkilerin büyüklüğü de ölçülebilmektedir. Programlama doğası gereği bilişsel bir etkinlik olduğundan, bireyin programlama sürecinin anlaşılması için bilişsel yapıların incelenmesi gerektiği önerilmektedir (Caspersen, 2007).

Dolayısıyla bu çalışmada programlama performansına ilişkin faktörler, bilişsel tabanlı bireysel farklılıklar çerçevesinde incelenecektir. Bilişsel farklılıklardan çalışma belleği ve uzamsal beceriler ele alınarak, bu değişkenlerle birlikte alanyazında programlama performansı ve ele alınan bilişsel faktörlerle ilgili olduğu ortaya konulan öz yeterlilik algısı, cinsiyet, ön deneyim değişkenleri ile birlikte çalışma grubunun üç farklı üniversitedeki öğrencilerden oluşması ile üniversite faktörü ile programlama performansını yordayan değişkenler modellenmiştir.

Bu deęişkenlerin tanımları ve programlama performansına ilişkin rolleri aőađıdaki bölümde özetlenmiştir.

1.3 Bilişsel Bireysel Farklılıklar

Bireyler hem genel becerileri, yetenekleri, bilgiyi işleme tercihleri, bilgiden anlam çıkarmaları ve bu anlamları yeni durumlara uygulamaları bakımından hem de öğrenme görevleri ya da gerçek yaşam görevleri ve ürünlerine göre de farklılaşmaktadır. Çünkü farklı görevler, farklı ortamlar ve farklı çıktılar farklı beceri ve yetenek gerektirmektedir (Jonassen ve Grabowski, 1993). Öğrenme ortamları açısından önemli bireysel farklılıklardan birisi bilişsel tabanlı bireysel farklılıklardır.

Bilişsel bireysel farklılıkların etkisinin kontrol edilmediđi durumlarda öğrenme süreçlerinin etkililiđinin de net olarak ortaya koyulamayacağı (Stalcup, 2005) ifade edilmekte olup, öğretsel manipölasyonlarda ve bireyin performansında bilişsel farklılıklar vurgulanmış (Ackerman, Beier, ve Bowen, 2002) ve bireylerin bilişsel becerilerindeki farklılıkların göz önüne alınmasının tüm yaşam boyunca önemli olduđu ifade edilmiştir (Alwin, 1994)

Programlama, problem çözme ve dil kazanımı gibi birçok üst düzey bilişsel süreci içermesiyle, bireyin göreve ilişkin algı, dikkat, bellek kapasitesi gibi bilişsel becerileri ve bilgiyi işleme özellikleri gibi temel psikolojik süreçlerin dikkate alınması gerektiđi vurgulanmaktadır (Irons, 1982; Jenkins, 2002; Ormerod, 1990). Programlama bilişsel bir süreç olduğundan bireylerin bilişsel bileşenleri programlama sürecinin bütünsel bir parçası olup (Leverington, 2010), bireyin bilişsel özellikleri bir programlama dili için gerekli bilişsel gereksinimlerin altındaysa bireyin motivasyonu düşebilir, eđer tam tersi şekilde üstüneyse birey görevden sıkılabilir (White ve Sivitanides, 2002).

Bireylerin bilişsel süreçlerine ilişkin bireysel farklılıkları bellek kapasiteleri, dikkat süreçleri, uzamsal becerileri, algıları, dil kazanımı, zihinsel modelleri, problem çözme, akıl yürütme vb. birçok bilişsel olarak yürütölen faktörlerden kaynaklanmaktadır. Bu çalışma kapsamında ele alınan çalışma belleđi ve uzamsal beceriler ise bilgisayar tabanlı ortamlardaki görevlere ilişkin sıklıkla çalışılan ve birçok araştırma bulgusunda tutarlı olarak etkileri tekrarlanan bilişsel farklılıklardır (Campbell ve Norman, 2010; Pak, Rogers, ve Fisk, 2006).

1.3.1 Uzamsal Beceri

Programlama performansı ile ilişkisi olan bireysel farklılıklardan birisi de uzamsal beceriler olup, bu beceri sözel beceri, matematik becerisi ve akıl yürütme becerisinden farklı olarak zekânın bir boyutu, aynı zamanda heterojen beceriler kümesi olarak ifade edilmektedir (Jones ve Burnett, 2008).

Uzamsal beceriler bilişsel farklılıklar alanyazınında önemli bir yer tutmaktadır (Alonso, 1998; Blustein ve Satel, 2003; Stanney ve Salvendy, 1995; Vicente ve Williges, 1988). Uzamsal beceri en genel anlamıyla sembolik ve dilsel olmayan bilginin yani uzamsal örüntülerin ya da zihinsel imgelerin algılanması, kodlanması, hatırlanması, dönüştürülmesi, ayırt edilmesi, bunların zihinsel temsili ve sonrasında bu yapıların özelliklerinin değerlendirilmesi olarak tanımlanmakta olup, konum, boyut, uzaklık, yön, şekil ve hareket gibi uzamsal özelliklerin bilşi olduğu ifade edilmektedir (Ahmed ve Blustein, 2005; Lawton, 2010; Linn ve Petersen, 1985; McGee, 1979).

Uzamsal becerinin günlük yaşam deneyimleri ile yakından ilişkili olup günlük hayattaki çeşitli araçlarla etkileşim, yer yön hatırlama - tarif etme, akılda canlandırma, plan yapma vb. birçok süreçte gerekli olduğu (Baran, Dogusoy, ve Cagiltay, 2007; Lawton, 2010; Rizzo ve diğ., 1998) matematik, fizik, kimya, mühendislik, mimarlık, tıp, grafik, sanat, bilgisayar bilimleri vb. birçok akademik alandaki başarı ile ve bu alanlara ilişkin meslek seçimi ile de ilişkili olduğu (McGee, 1979; Sorby, Drummer, Hungwe, ve Charlesworth, 2005; Wright, Thompson, Ganis, Newcombe, ve Kosslyn, 2008) ayrıca, bilgisayar tabanlı ortamlardaki çeşitli görevlerde ve ortamla etkileşimlerle de önemli bir faktör olarak bireysel farklılıkların bulunduğu bilişsel bir beceri olarak ele alınmaktadır (Chen, 2000; Vicente ve Williges, 1988). Programlama becerisi gereken bilgisayar bilimleri, bilgi ve iletişim teknolojileri, mühendislik bilimleri gibi birçok alanda da uzamsal beceriler bireysel farklılıkların önemli bir kaynağı olarak ortaya çıkmaktadır. Neden sonuç ilişkisi midir kesin olmamakla birlikte, uzamsal becerisi yüksek bireylerin programlama becerisi kullanımı gerektiren meslekleri (mimarlık, mühendislik vb.) seçme eğiliminde olduğu da ifade edilmektedir (Jones ve Burnett, 2008).

Linn ve Petersen (1985) uzamsal becerilere ilişkin alanyazında yapılan çalışmalarını dört temel kategoride toplamışlardır:

- a) *Psikometrik bakış açısı*: Uzamsal becerilerdeki faktörleri tanımlayabilmek için farklı uzamsal görevlerdeki korelasyonların karşılaştırılması ve uzamsal becerilere dönük faktör analitik çalışmalar.
- b) *Diferansiyel bakış açısı*: Uzamsal becerinin farklı gruplarda karşılaştırılması (erkek-kadın, çocuk-genç-yaşlı vb.).
- c) *Bilişsel bakış açısı*: Belirli uzamsal görevlerin çözümünden kullanılan genel süreçlerin tanımlanması.
- d) *Stratejik bakış açısı*: Farklı bireylerce çözülen bir uzamsal beceri görevinde kullanılan farklı stratejilerin niteliksel olarak tanımlanması.

Uzamsal becerilerin yapısını açıklamaya dönük olarak yapılan faktör analitik çalışmalarda uzamsal becerinin alt boyutlarına ilişkin farklı bulgular göze çarpmaktadır. Linn ve Petersen (1985) uzamsal becerilere ilişkin yaptıkları metaanaliz çalışmasında problem çözme sürecindeki gerekli bilişsel süreçlerdeki farklılıklara göre uzamsal becerilerin, zihinsel döndürme, uzamsal algı ve uzamsal görselleştirme olarak üç kategoride sınıflanabileceğini ifade etmişlerdir. McGee (1979) uzamsal görselleştirme ve uzamsal yönelim olmak üzere iki kategoride incelerken, Lohman (1988) uzamsal becerilerin uzamsal görselleştirme, uzamsal ilişkiler ve uzamsal yönelim olmak üzere üç farklı boyutu olduğunu ifade etmiştir. Carroll (1993) uzamsal becerilerin faktör yapısına ilişkin yaptığı çalışmada, uzamsal görselleştirme, uzamsal ilişkiler, kapalılık hızı, kapalılık esnekliği, algısal hız ve görsel bellek olarak altı farklı yapı ortaya koyarken, Kimura (2000) uzamsal becerilerin farklı alt boyutu üzerinde çalışarak bunları; uzamsal yönelim, uzamsal konumlandırma belleği, hedefleme, uzamsal görselleştirme, ayırıştırma ve uzamsal algı olarak sınıflamıştır.

İlgili alanyazın incelendiğinde farklı faktör analitik çalışmalarda, bazı faktörler araştırmalarda ortak ya da benzer olmakla birlikte, farklı isimde ve farklı sayıda alt faktörlerin de bulunduğu görülmektedir. Bunlar arasında zihinsel döndürme, uzamsal görselleştirme, uzamsal yönelim ve uzamsal ilişkiler ilgili çalışmalarda en çok incelenen alt beceriler olarak ortaya çıkmaktadır.

Zihinsel döndürme iki-boyutlu ya da üç-boyutlu nesnelere bireyin zihninde döndürmesi ve doğru bir çözüme ulaşmak için karmaşık uzamsal bilgileri birkaç sıralı aşamada manipüle etme becerisi (Samsudin ve İsmail, 2004; Shepard ve

Metzler, 1971); bedeninin dışındaki nesnelere hareketlerinin imgelemesini içeren uzamsal düşünmenin bir bileşeni ya da zihinsel imgelerin görsel depoda dönüştürüldüğü ve temsil edildiği görsel uzamsal bir süreç olarak tanımlanmaktadır (Khooshabeh ve Hegarty, 2010).

Uzamsal görselleştirme iki ya da üç-boyutlu görselleri zihinde kodlama ve oluşan zihinsel imge üzerinde katlama, çevirme, yönünü değiştirme, karşılaştırma gibi işlemler yapabilme becerisi olarak tanımlanmaktadır.

Uzamsal yönelim görsel bir uyarana örüntüsü içindeki öğelerin düzenini anlayabilme becerisi ve sunulan uzamsal konfigürasyon içindeki değişen yönelimlere rağmen karıştırmadan bu düzeni sürdürebilme yeteneğidir (Kimura, 2000; McGee, 1979). *Uzamsal yönelimin* genel olarak nesnelere farklı açılardan görünüşlerini imgeleyebilme becerisi olduğu bilinmekte ve (Thurstone, 1950 ss.518) tarafından da “bireyin kendi beden yöneliminin problemin temel bir parçası olduğu bir durumda uzamsal ilişkileri düşünebilme becerisi” olarak tanımlanmaktadır.

Uzamsal ilişkiler görece olarak görsel basit örüntüleri dönüştürme ya da döndürme becerisidir.

1.3.2 Çalışma Belleği

Çalışma belleği karmaşık etkinlikler sırasında bilgiyi depolayan ve bütünleştiren bilişsel bir sistem olup (Baddeley, 1992; St Clair-Thompson ve Botton, 2009), çeşitli karmaşık bilişsel görevlerin performansında önemli rolü sahip bir yapıdır (Haavisto ve Lehto, 2005).

Çalışma belleğini kavramını ilk kez Miller, Galanter ve Pribram (1960) “*Plans and the Structure of Behavior*” adlı kitaplarında bireylerin planlarının ve planlara ilişkin bileşenlerin saklandığı ve gerektiğinde geri çağrıldığı yer olarak kullanmışlardır. Baddeley ve Hitcher (1974) ise çalışma belleğini bir bellek modeli olarak ele alarak bileşenlerini tanımlamışlar ve ilk çalışma belleği modelini geliştirmişlerdir. Çalışma belleği modelinde öğrenme, akıl yürütme, karşılaştırma, kavrama vb. herhangi bir bilişsel performans anında bilginin geçici olarak tutulması ve manipülasyonunun yapıldığı kısım olarak tanımlanmıştır (Conway, Jarrold, Kane, Miyake, ve Towse,

2007). Model bir fonolojik döngü, görsel uzamsal depo, episodik depo ve bir merkezi yürütücüden oluşur.

Fonolojik döngü genellikle konuşma tabanlı sözel bilgilerin kaydedildiği sınırlı kapasitedeki seslendirme döngüsüdür. Görsel uzamsal döngü ise nesnelere ve bunların konumlarının öğrenilmesi gibi görsel ve konumsal bilgilerin kısa süreli depolanması ile sorumlu olup aynı zamanda zihinsel imgeler oluşturmada ve bunların manipülasyonunda da rol oynar. Episodik bellek bilinçli olarak erişilebilen ve yeni bilgiye dayalı olarak entegre temsiller oluşturmak için uzun süreli episodik ve anlamsal bellek arasında yer alan ara yüzdür (Baddeley, 1999). Bu kısım normalde fonolojik ve görsel uzamsal deponun kapasitesini aşan büyük miktarlarda bilgiyi geçici olarak saklayabilir. Episodik depo özellikle çoklu modalitede kodlamalar yaparak çalışma belleği ve uzun süreli belleğin bileşenlerini tek bir yapıda entegre eden temsiller oluşturmasıyla öğrenme açısından önemlidir (Dehn, 2011). Merkezi yürütücü ise diğer alt sistemlerin denetiminden sorumludur. Merkezi yürütücü çalışma belleğindeki bazı bilişsel süreçlerin düzenlenmesi ve koordine edilmesinde anahtar rol oynar, stratejiler ve farklı kaynaklardaki bilgiyi birbiri ile bütünleştirir.

Çalışma belleği, uzun süreli belleğin geçici aktivasyonu, çoklu görevlerin koordinasyonu, görevler arası geçiş ve stratejilerin çağrılması, kullanılacak ya da ayrılacak kapasitesinin organizasyonu gibi kritik rolleri olması ile bireylerin tüm bilişsel süreçlerinin yanı sıra problem çözme, düşünme, öğrenme, yaratıcılık, akıl yürütme vb. işlevlerinde etken durumdadır (Jarrod ve Towse, 2006; St Clair-Thompson ve Botton, 2009). Bu nedenle çalışma belleği matematik, fen, mühendislik, bilgisayar bilimleri, dil bilimi vb. daha birçok alandaki performansı belirleyen bir yapı olarak ortaya çıkmaktadır (Daneman ve Carpenter, 1980; Daneman ve Merikle, 1996; Mancy ve Reid, 2004).

Çalışma belleğinin hem akıl yürütme becerileri hem de genel zeka ile pozitif yönde yüksek korelasyonda olması ile programlama performansında da belirleyici olduğu ve iyi bir programcı olmanın yüksek çalışma belleği kapasitesine, iyi problem çözme becerilerine sahip olup aynı zamanda aktif öğrenen olmak ile olacağı ifade edilmektedir (Shute, 1991).

Her programlama dili tek başına birey için yeni bir dil olduğundan çeşitli sözdizimsel (sintaks) becerileri gerektirir. Kod yazma, kod okuma, her programlama dilinin kendine ait yazım kurallarını bilmek bireylerin sözel bellek süreçlerini işe koşmalarını gerektirir. Diğer yandan, yazılmış bir program kendi içinde küçük kod parçacıklarının bütününden oluşmasıyla, gerek bütün bir programı kavrama, hata ayıklama ve programı düzenleme gibi işlemlerin kod parçacıkları arasında gezinim gerektirmesi gerekse de programın algoritmasına ilişkin akışı ve hiyerarşiyi temsil için zihinsel model oluşturulması ise bireyin görsel-uzamsal belleklerinin rolünü ön plana çıkarmaktadır.

Bu nedenle, bu çalışmada çalışma belleğine ilişkin sözel bellek ve görsel-uzamsal bellek alt boyutları ele alınacaktır.

1.4 Programlamaya İlişkin Özyeterlilik Algısı

Bandura (1994) öz yeterliliği “bireyin bir performans için gerekli eylemleri organize edebilme ve gerçekleştirebilmedeki yeterliliğine ilişkin kendi yargısı” olarak tanımlamaktadır. Bu yargı bireyin sahip olduğu becerilerle değil, sahip olduğu becerilerden bağımsız olarak ne yapabileceğine ilişkin inancı ile ilişkilidir.

Öz yeterlilik inancı bireyin bir görevi başarımdaki etkinlik seçimini, harcanan çabanın seviyesini, zorluklarla başa çıkmadaki direncini ve en önemlisi de performansını etkilemektedir (Bandura, 1977). Bireyler aynı bilişsel beceri seviyesine sahip olsa da öz yeterlilik algısı yüksek olanların üst düzey zorluklarla başa çıkmada daha iyi oldukları ve hedefe ulaşmada daha sebatkar oldukları, öz yeterliliği düşük olan bireylerin ise problemleri gözlerinde daha da büyütme ve bu nedenle de strese girme ve depresif hissederek durumu daha da karmaşılaştırmanın söz konusu olduğu ifade edilmektedir (Davidsson, Larzon, ve Ljunggren, 2010). Ayrıca, öz yeterlilik algısı bireylerin kendini izleme ve değerlendirme, benlik algısı, sonuç beklentisi ve algılanan kontrol gibi diğer inançları ile ilişkili olup, bir hedefi başarmadaki öz düzenleme ve strateji seçiminde de belirleyici olması ile bireyin öğrenme sürecindeki motivasyonu ve başarı çıktılarını etkilemektedir (Zimmerman, 2000). Bu nedenle bireyin herhangi bir bağlamdaki başarısında bilgi, beceri ve daha önceki başarılarından çok, bireyin kendi becerilerine ve çabalarının sonuçlarına ilişkin inancı oldukça güçlü bir etkiye sahiptir (Pajares, 1996). Bu bağlamda, öz yeterlilik bireyin becerisine olan inancı

olarak tanımlanıp, yeterli bilgi ve beceri olsa bile bireyin kendi performansına ilişkin şüpheleri varsa, motivasyonu düşükse ve başaracağına ilişkin algısı olumsuzsa bireyin başarısız olabileceği ifade edilmektedir (Aşkar ve Davenport, 2009).

Programlama üniversitede bilgisayar bilimi ile ilgili alanlarda oldukça önemli bir performans olup, fen ve matematik derslerinin aksine, programlama dersi özellikle giriş seviyesinde öğrenciler tarafından oldukça zor olarak algılanan bir derstir (Aşkar ve Davenport, 2009). Bu nedenle bireylerin genellikle öz yeterlilik algılarının düşük olması nedeniyle, yani programlamayı baştan zor kabul etmeleri nedeniyle, programlama dersinde başarısız olmaları olasıdır.

İlgili alanyazında birçok konu alanındaki akademik başarı, öz yeterlilik ve demografik bilgiler arasındaki ilişkinin araştırılmış olmasına rağmen, öz yeterliliğin programlama performansındaki rolüne ilişkin vurgu yapan çalışmaların sınırlılığına dikkat çekilmektedir (Aşkar ve Davenport, 2009).

1.5 Programlama Ön Deneyimi

Bir konudaki önceki deneyimler ya da ön bilgiler çoğu zaman bireylerin becerilerine ilişkin farklılıklarda dengeleyici etkiye sahip olarak ortaya çıkmaktadır. Yani birey yeterli ön bilgiye sahipse ya da önceden deneyimi varsa bazı bilişsel becerileri düşük dahi olsa o beceriyi kullanmaya gerek duymadan sahip olduğu ön yaşantılar süreci kolaylaştıracaktır. Ya da tam tersi şekilde belirli becerileri yüksek dahi olsa bireyler yeterli ön bilgiye sahip olmadıklarında harcayacakları zihinsel çaba artacağından bilişsel kaynaklarını bu yönde ayırmaları gerekebilir.

Ön bilgi ya da programlama arka planı olarak adlandırılan deneyim bireylerin programlamaya ilişkin mevcut performansını etkileyen en önemli faktörlerden birisi olarak ortaya çıkmaktadır. İlgili alanyazında ön bilgi ya da deneyimin ayrı ayrı ya da tek faktör olarak programlama performansına ilişkin etkisi ele alınmaktadır (Bergin ve Reilly, 2005; deRaadt ve diğ., 2005; Lau ve Yuen, 2011).

Programlamaya ilişkin ön deneyim; ön bilgi, programlama arka planı, program yazmış olma, programlama ile ilgili ders almış olma, bilinen programlama dili sayısı gibi farklı olarak da alınmaktadır.

1.6 Programlama Performansı ve Cinsiyet

Bireylerin bilgisayar tabanlı ortamlarda performanslarının farklılaştığı değişkenlerden birisi de cinsiyettir. Gerek bu çalışma kapsamında ele alınan bilişsel beceriler gerekse programlama cinsiyete ilişkin farklılıkların oldukça yüksek olduğu beceriler olarak bilinmektedir.

Cinsiyet özellikle birçok bilişsel beceri de farklılaştığından, bu farkın da dolaylı olarak ya da doğrudan bu ortamlardaki performansta da fark yarattığı ifade edilebilir. Örneğin kadınların konum bellekleri ve sözel bellekleri erkeklere göre daha yüksek olarak ortaya çıkarken, erkeklerin ise zihinsel döndürme ve uzamsal görselleştirme gibi uzamsal becerileri tutarlı olarak kadınlara göre yüksek olarak ortaya çıkmaktadır (Halpern, 2000; Linn ve Petersen, 1985). Buna bağlı olarak, bu becerilerin işe koşulması gerektiği görevlerde performans da cinsiyete bağlı olarak farklılaşacaktır.

Diğer yandan genel olarak bilgisayar bilimleri ile ilgili alanlar erkek ağırlıklı tercih edildiğinden bireylerin var olan becerilerinin ve başarılarının dışında cinsiyet bakımından yeterlilik hissi farkı olmakta ayrıca, kadın öğreticilerinin ve kadın rol modellerinin olmaması nedeniyle dışarıdan bakıldığında erkekler ve kadınlar bakımından ön yargısal bir ayrım söz konusudur (Byrne ve Lyons, 2001). Gürer ve Camp (2002), programlamada kadınların genel durumuna ilişkin ilgili alanyazın incelemesi yaptıkları çalışma sonunda, kadınların programlamaya ilişkin konularda erkeklere göre oldukça az yer almalarının nedenlerine ilişkin bir derleme yaparak bu faktörleri şu şekilde sınıflamıştır: bilgisayarla ilgili konu ve işlere ilgili tutum, bilgisayar deneyimi, bilgisayar oyunlarına ilişkin farklılıklar, rehberlik ve rol modellerinde farklılıklar, öz güven, programlamaya ilişkin iş ortamları, sosyal etki, aile ve öğreticilerin teşviki, tamamı kadınlara ilişkin ortamlardaki (kız meslek liseleri vb.) programlama teması eksikliği, iş ve aile dengesi (bilgisayar bilimi alanında çalışan kadınların yoğunluktan dolayı iş-ev dengesi zorluğu), lisansüstü alanlara devam. Bu nedenle özellikle programlama performansı gereken görevlerde ve derslerde erkeklerin lehine bulgular söz konusu olup, genel olarak cinsiyet programlama performansında bir bireysel farklılık olarak görülmektedir.

1.7 Çalışmanın Önemi

İyi bir programcı olmak matematik, problem çözme, akıl yürütme, çeşitli bilişsel beceriler gibi birçok beceriyi ve deneyimi birlikte gerektirdiğinden, programlama performansı sadece tek bir beceri ya da birkaç becerinin bileşimi değil, beceriler hiyerarşisi olup programcı bu becerilerinden herhangi birisini herhangi bir aşamada kullanır (Jenkins, 2002). İyi bir programcının sergilediği nitelikler tanımlanabilirse, bu konuya ilişkin verilen dersler ve mesleklerde bireylerin seçimine yönelik uygun değerlendirme ölçütleri belirlenebilir ve potansiyel olarak acemi ya da uzman olan bireylere verilecek programlama eğitiminde daha etkili yöntemler kullanılabilir (Michelson, 2004).

Programlamanın oldukça karmaşık bilişsel süreçler gerektiren görevler olduğu gerçeğinden yola çıkarak bu sürece etki eden bilişsel faktörlerin bilinmesi ve bu faktörlerin alanyazında önemli olarak ortaya çıkan diğer faktörlerle birlikte etkisinin incelenerek programlama performansının açıklanması önemlidir. Bireylerin bilişsel süreçlerine ilişkin ölçümler ve başarı etkenlerinin bilinmesi bireylere verilecek öğretimin düzenlenmesinde de ipucu olarak kullanılabilir (Jegede, 2009; Shute, 1991). Ayrıca bireylerin programlama performansını etkileyen faktörleri bilmenin önceden dezavantajlı olarak gelen bireyleri desteklemede yararı olabilir (Byrne ve Lyons, 2001).

Bu nedenlerle bireylerin genel yeteneklerinin yanı sıra birçok beceri ve zeka ile oldukça ilişkili olduğu bilinen programlama performansının; gerek bireylerin zayıf yönlerinin belirlenerek geliştirilmesi, gerekse bireylerin özelliklerine yönelik programlama öğretiminin sağlanması, ayrıca bilgisayar ve bilişim alanlarındaki meslekler için birey seçiminde ve bireylerin meslek seçiminde kendi özelliklerinin farkına varması açısından programlama sürecine etki eden bireysel özelliklerin bütünsel bir yaklaşımla ele alınarak modellenmesi oldukça önemlidir.

1.8 Çalışmanın Amacı

Bu çalışmanın amacı programlama performansına etki eden bilişsel faktörlere ilişkin bir model geliştirerek bu performans üzerindeki bilişsel tabanlı bireysel farklılıkların etkisini diğer demografik özelliklerin etkisi ile birlikte ortaya koymaktır. Çalışmada bilişsel tabanlı bireysel farklılıklardan uzamsal beceri ve çalışma belleğinin, öz yeterlilik, cinsiyet, üniversite ve ön deneyim değişkenleri ile birlikte ele alınarak, bu değişkenler arasındaki ilişkilerin belirlenmesi ve programlama performansını yordayan değişkenlerle, bunların önem derecelerinin ortaya konulması hedeflenmektedir.

1.9 Araştırma Soruları

1. Bireylerin programlama performansı, öz yeterlilik algıları, uzamsal becerileri , çalışma belleği düzeyleri arasında bir ilişki var mıdır?
2. Bireylerin programlama performansları ön deneyimleri, cinsiyetleri ve üniversitelerine göre farklılaşmakta mıdır?
 - 2.1 Bireylerin programlama performansları ön deneyimlerine göre farklılaşmakta mıdır?
 - 2.2 Bireylerin programlama performansları cinsiyetlerine göre farklılaşmakta mıdır?
 - 2.3 Bireylerin programlama performansları üniversitelerine göre farklılaşmakta mıdır?
3. Bireylerin bilişsel becerileri, programlamaya ilişkin öz yeterlilik algıları, ön deneyimleri, üniversiteleri ve cinsiyetlerine ilişkin bir model programlama performansının ne kadarını yordamaktadır?
4. Programlama performansının yordanmasına yönelik kurulan modelde her bir değişkenin bireysel önemi nedir?
 - 4.1 Uzamsal yönelim becerisinin programlama performansını yordamadaki önemi nedir?

- 4.2 Zihinsel döndürme becerisinin programlama performansını yordamadaki önemi nedir?
- 4.3 Görsel uzamsal belleğin programlama performansını yordamadaki önemi nedir?
- 4.4 Sözel belleğin programlama performansını yordamadaki önemi nedir?
- 4.5 Öz yeterlilik algısının programlama performansını yordamadaki önemi nedir?
- 4.6 Ön deneyimin programlama performansını yordamadaki önemi nedir?
- 4.7 Cinsiyetin programlama performansını yordamadaki önemi nedir?
- 4.8 Üniversite değişkeninin programlama performansını yordamadaki önemi nedir?

1.10 Çalışmanın Sınırlılıkları

Çalışmada programlama performansı olarak katılımcıların programlama dersi sonundaki geçme notları alınmıştır. Geçme notları sadece yazılı sınavlarla verildiği için, programlama bilgisi dışında uygulamaya dönük beceri olan programlama performansını ölçmede sınırlı kalmış olabilir.

Çalışmada örneklem sayısı sınırlı olduğundan, veriler ağaç tabanlı yöntemlerde kullanılan eğitim verisi ile analiz edilen modelin test verisi test edilmesi noktasında sınırlı kalmıştır.

1.11 Tanımlar

Programlama Performansı: Herhangi bir programlama dilini kullanarak verilen bir programlama görevini başarı ile yerine getirme, programı anlama, program hatalarını ayıklayabilme, yeniden bir program yazabilme, programın algoritma şemasını oluşturabilme becerisi olarak alınmıştır.

Bu çalışma kapsamında programlama-I dersi yıl sonu notları olarak alınmıştır.

Çalışma Belleği: Çalışma belleği yapı olarak, karmaşık bilişsel etkinlikler esnasında bilgiyi depolayan ve bütünleştiren bir sistem olup, öğrenme, akıl yürütme, karşılaştırma, problem çözme, kavrama gibi bilişsel performanslar anında bilginin geçici olarak tutularak manipülasyonların yapıldığı bellek kısmıdır. Bilişsel

bir beceri olarak ise, bireyin eş zamanlı olarak bilgiyi işleme ve bilgi üzerinde işlem yapma becerisidir.

Uzamsal Beceri: Uzamsal beceri sembolik ve dilsel olmayan bilginin yani görsel-uzamsal örüntülerin ya da zihinsel imgelerin algılanması, kodlanması, hatırlanması, dönüştürülmesi, ayırt edilmesi ve içsel olarak zihinsel temsili süreçlerini içermekle birlikte bu içsel temsillerin uzamsal problemlerin çözümünde kullanabilme becerisidir. Konum, boyut, uzaklık, yön, şekil ve hareket gibi uzamsal özelliklerin bilişidir.

Programlama Öz Yeterliliği: Bireyin bir programlama görevini yerine getirmede kendi yeterliliklerine ilişkin yargısı, kendi başarısına ilişkin inancıdır.

Artırım (Boosting): Çoklu modeller ya da sınıflayıcılar üretme ve bu modellerden elde edilen tahminleri tek bir tahmin üretmek üzere birleştirmek için ağırlıklar elde etmede kullanılan “tahminleyici veri madenciliği alanına” ilişkin tekniktir.

Regresyon Ağacı: Bağımlı değişkene göre, bağımsız değişkenlere ilişkin matrisi hiyerarşik homojen ağaç şeklinde alt dallara ayırarak, ağaç yapısı içindeki ara düğümlerde bağımlı değişkene ilişkin en iyi ayrımın yapıldığı yerin belirlenmesidir.

Torbalama (Bagging): Ağaç tabanlı yöntemlerde fonksiyon belirlenmesine ilişkin tahminin varyansını azaltma tekniğidir.

Rastgele Ormanlar: Çok sayıda ilişkisiz hale getirilmiş ağaçları bir araya toplayarak daha sonra bunları ortalamasını alan, torbalama tekniğinin oldukça değiştirilmiş bir halidir.

Artırılmış Regresyon Ağacı (Boosted Regression Tree): Her bir terimin bir ağaç olduğu, regresyon modeli olarak da düşünülebilen, birçok ağaç topluluğun lineer kombinasyonunca oluşturulan modeldir. Artırım ve regresyon ağacı tekniklerinin birleşiminden oluşur.

Yordama (Prediction): Yordama veri madenciliğine ilişkin analizlerde tahminleme/tahmin etme kavramına karşılık kullanılmıştır. Bu kavramın İngilizcesi “prediction” olup veri madenciliğine dayalı tekniklerde genellikle tahmin etme olarak kullanılırken, eğitim alanındaki çalışmalarda yordamak ve açıklamak olarak kullanılması yaygındır. Bu çalışmada bütünlüğü sağlamak amacıyla yordama kavramı tercih edilmiştir.

2. İLGİLİ ÇALIŞMALAR

Bu kısımda programlama performansına etki eden faktörleri belirlemeye dönük araştırmalar bilişsel tabanlı bireysel farklılıklara ilişkin çalışmalar ve bilişsel olmayan faktörlere ilişkin çalışmalar olarak ayrılarak incelenmiştir. Bilişsel farklılıklardan çalışma belleği ve uzamsal beceriler iki ayrı başlık olarak incelenirken, bilişsel olmayan faktörler (öz yeterlilik algısı, cinsiyet ve ön deneyim) tek başlık altında toplanmıştır.

Çalışmaların belirlenmesinde çeşitli ulusal ve uluslar arası akademik veritabanları taranarak, dergi makaleleri, konferans ve kongre bildiri kitapçıklarından basılan bildiriler, tezler, kitap ve çevrimiçi dokümanlar olmak üzere farklı türden kaynaklara erişilmiş, bunlardan bu çalışma bağlamına uygun olan ve örnek teşkil eden çalışmalar seçilerek içerik analizi ile incelenmiştir.

Ulaşılan çalışmaların incelenmesi sırasında, bu çalışmaların da referans aldığı ilgili kaynakçalara da ayrıca erişilerek analiz edilmiştir. Çalışmaların seçilmesinde ele alınan bağımlı ve bağımsız değişkenlerin bu çalışma ile benzerlik göstermesi gözetilmiş diğer yandan güncel olan kaynaklar öncelikli olarak incelenmiş, fakat bu bağlamda yapılan çalışmalara temel teşkil eden ve birçok güncel çalışmada da referans verilen daha eski çalışmalara da yer verilmiş, bu nedenle ilgili alanyazın taramasında herhangi bir yıl aralığı gözetilmemiştir.

İlgili çalışmalar kısmında yer verilen araştırmalara erişim için kullanılan veri tabanları ve tarama için kullanılan anahtar kelimeler Çizelge 2.1' de verilmiştir.

Çizelge 2.1. İlgili çalışmalar için taranan kaynaklar ve anahtar kelimeleri

Veri tabanları, arama motorları ve diğer kaynaklar	Anahtar Kelimeler
SciVerse ScienceDirect	“Programming ability” + ”cognitive abilities”
SciVerse Scopus	“Programming ability”+ “cognitive characteristics”
JSTOR	“Programming performance” + spatial ability
ProQuest (thesis and dissertations)	“Programming performance” + “working memory”
SAGE Publications	Programming ability +self-efficacy
Taylor and Francis Online	Programming success + demographic characteristics
Wiley Online	Programming performance +prior knowledge/experience
scholar.google.com	Programming performance+ gender
tez2.yok.gov.tr	Programming performance+ success factors
Ulakbim (ulusal veri tabanları)	Programming ability/performance/success + predictive variables
library.hacettepe.edu.tr	Programlama + bilişsel beceriler
books.google.com	Programlama becerisi / performansı+ uzamsal beceri
PsycNET	Programlama becerisi/performansı+ çalışma belleği
	Programlama başarısı/ performansı + öz yeterlilik
	Programlama başarısı/performansı + demografik değişkenler
	Programlama/ performansı + cinsiyet+ ön deneyim/bilgi
	Programcı + programlama+ bilişsel özellikler

2.1 Çalışma Belleği ve Programlama Performansına İlişkin Çalışmalar

Bergersen ve Gustafsson (2011), çalışma belleği ve deneyimin programlama bilgisi ve programlama becerisi gelişimi üzerindeki etkisini incelemiştir. Çalışmaya sekiz farklı Avrupa ülkesindeki toplam dokuz şirketten 65 profesyonel yazılım geliştirme uzmanı katılmıştır. Çalışmada programlama becerisi araştırmacılar tarafından geliştirilen bir araçtan elde edilen performans ölçümü olarak alınmıştır. Bu araç toplamda 12 tane programlama kodunu yazmaya ya da

düzenlemeye dayalı görevden oluşmakta olup, her bir görevin süresi 10 dk. ile 50 dk. arasında değişmiştir. Performans puanları değerlendirilirken hem doğru çözüm için harcanan süre hem de ürünün niteliği birlikte ele alınmıştır. Çalışma belleği üç farklı çalışma belleği testi ile ve ön programlama bilgisi ise 30 çoktan seçmeli Java programlama diline ilişkin hazırlanmış sorulardan oluşan testle ölçülmüştür. Programlama deneyimi olarak ise bireylerin program yazma yılı ve ortalama ne kadar kod satırı yazdığı bilgisi esas alınmıştır. Programlama becerisini açıklamaya yönelik model kurulurken çalışma belleği ve ön deneyimin programlama bilgisini doğrudan etkileyeceği, programlama bilgisinin ise programlama becerisini etkileyeceği şekilde hipotezler oluşturulmuştur. Modelin test edilmesi ile çalışma belleğinin ve deneyimin programlama bilgisi aracılığıyla programlama becerisine etki ettiği, programlama bilgisinin ise doğrudan programlama becerisini etkilediği ortaya konulmuştur.

Mancy (2007), karmaşık bir beceri olarak programlama becerisini ele aldığı çalışmada, programlama becerisinin kazanımında örtük ve açık öğrenme süreçlerinin rolünü incelemiştir. Örtük ve açık öğrenme süreçlerini çalışma belleğine dayalı bilgiyi işleme modelleri kapsamında depolama ve işleme olarak iki alt boyutta ele almıştır. Geçici depolama kapasitesi sayı dizisi testi ile ölçülürken, işleme becerisi ise yerinden çıkarma testi ile ölçülmüştür. Bireylerin programlama performansı olarak programlama sınavları alınmıştır. Çalışma sonunda bireylerin programlama notlarının çalışma belleği kapasitelerinin depolama süreçleri ile düşük, işleme süreçleri ile ise oldukça yüksek ilişkide olduğunu bulmuştur.

Yousoof, Sapiyan ve Kamaluddin (2007), programlama öğrenme sürecinde çalışma belleğinin yerini bilişsel yük perspektifinden ele almışlardır. Programlama öğreniminde duysal bellek, çalışma belleği ve uzun süreli bellek olmak üzere üç belleğin de etkin olduğunu ifade etmişlerdir. Özellikle programlama öğrenimi birçok bilişsel etkinliğin etkileşimini gerektirdiğinden çoğu zaman bilişsel yük oluştuğunu bunun da programlamadaki başarısızlığın nedenlerinden birisi olduğunu belirtmişlerdir. Nitekim bu bilişsel yükün azaltılmasının sürecin görselleştirilmesi ile mümkün olacağını iler sürerek, buna yönelik kavram haritası metaforunu önermişlerdir.

Mancy ve Reid (2004), çalışma belleği ve bilişsel stilin programlamaya giriş dersi başarısı ile ilişkisini incelemişlerdir. Çalışmaya programlamaya giriş dersini alan 150 öğrenci katılmıştır. Bireylerin çalışma bellekleri sayı uzamı testi ile ölçülmüştür. Alan bağımlılık ise gömülü figürler testi ile ölçülmüştür. Programlama performansı ölçümleri ise ders kapsamında dönem boyunca yapılan uygulama sınavı 1, sınıf içi test, uygulama sınavı 2 ve final sınavı sonuçları olarak ayrı ayrı alınmıştır. Çalışmada programlama performansı olarak alınan dört sınav ölçümü ile çalışma belleği ve alan bağımlılık puanları arasındaki korelasyon katsayıları incelenmiştir. Çalışma sonunda alan bağımlılık puanlarının tüm sınav puanları ile anlamlı ilişkisi olduğu bulunurken, çalışma belleğinin ise sadece uygulama sınavı 2 ile ilişkili olduğu bulunmuştur. Çalışmada başarı ya da başarısızlığa ilişkin herhangi bir neden sonuç ilişkisi incelenmediğinden, programlama performansının etkenleri arasında çalışma belleğinin rolünün az, bilişsel stilin rolünün fazla olduğunu ileri sürmenin mümkün olmadığı, ancak yapılan sınavların içeriği ile bu ilişkinin ortaya çıkmış olabileceği şeklinde bulgular yorumlanmıştır.

Pena ve Tirre (1992), çalışmalarında programlama becerisi kazanımının ilk aşamalarında bilişsel becerilerin rolünü araştırmışlardır. Çalışmada incelenen bilişsel beceriler çalışma belleği, genel akıl yürütme becerileri, programlamaya ilişkin bilgi ve cebir problemi çözme sürecindeki bir takım beceri bileşenleri olarak alınmıştır. Programlama performansı için ise Pascal diline ilişkin bilgisayar tabanlı bir eğitim verilmiştir. Kazanımı ölçmek için eğitimden önce ve sonra programlamaya ilişkin kavrama testleri uygulanmıştır. Programlama performansını yordayan bilişsel becerilere ilişkin bir model kurularak test edilmiştir. Çalışma sonunda programlama performansını yordamada cebir kelime problemi dönüştürme becerileri, akıl yürütme, çalışma belleği ve sözel bilgi gibi daha genel becerileri etkilemiştir. Çalışma belleği, akıl yürütme ve sözel bilgi becerileri ise programlama becerisi kazanımını doğrudan belirleyen bilişsel beceriler olarak bulunmuştur. Yol katsayıları incelendiğinde ise programlama performansında en önemli rol oynayan bilişsel becerinin çalışma belleği olduğu ortaya konulmuştur.

Shute (1991) ön bilgi ve genel bilişsel beceriler, problem çözme becerileri ve öğrenme stilleri olmak üzere farklı bireysel farklılıkların programlama performansı ile ilişkisini araştırmış ve bilişsel becerilerden çalışma belleğinin programlama

performansına ilişkin hem bildirimsel hem de uygulama bilgisini anlamlı şekilde yordadığını ve problem çözme becerilerinin, öğrenme stilinin ve çalışma belleğinin birlikte programlama performansına ilişkin varyansın büyük bölümünü açıkladığını bulmuştur. Nitekim bu sonuçların öğrenme ortamı, öğrenen davranışları ve öğrenme çıktılarına ilişkin ölçümlerle de etkileşim içinde olduğuna dikkat çekmiştir.

Merrienboer ve Paas (1990), programlama becerisini karmaşık bilişsel becerilerden birisi olarak ele alarak, programlama öğretimine ilişkin uygulamaların geliştirilmesinde etkililiğin sağlanması otomatikleşme ve şema kazanımı perspektifinden öğretim tasarım ilkeleri önermişlerdir. Bireylerin yürütücü işlevlerinin etkin olduğu otomatikleşme ve şema kazanımı ile programlama öğrenimine ilişkin öğrenme sonuçlarının artacağını ve ayrıca programlama becerilerinin diğer üst düzey bilişsel süreçlere transferinin sağlanacağını ileri sürmüşlerdir. Otomatikleşmenin programlama davranışını kontrol eden görev odaklı işlemleri yürütürken, şemaların ise yeni problem durumları içinde analogiler sağlayan program planları gibi daha bilişsel yapılar ortaya koyduğunu ifade etmişlerdir. Hem otomatikleşme hem de şema kazanımı için, geleneksel açıklayıcı örnekler yerine transferi ve etkili öğrenme sonuçlarını destekleyen çalışılmış örneklerle programlama öğretiminde yer verilmesini önermişlerdir.

Shneiderman ve Mayer (1979), programcıların programlama bilgisinin yazım bilgisi ve anlamsal bilgi olarak iki alt kategoriden oluştuğunu ifade ederek, programcıların davranışlarına ilişkin bir bilişsel model önermişlerdir. Önerdikleri modelde, programcıya gelen yeni göreve ilişkin bilginin kısa süreli bellekte tutulduğunu ve bunun uzun süreli bellekte var olan önceki ilişkili bilgilerle çalışma belleğinde bütünleştirileceğini öne sürerek, yine problemin çözümüne ilişkin süreçlerin çalışma belleğinde gerçekleştiğini ifade etmişlerdir.

Yukarıda kısaca özetlenen, çalışma belleği ve programlama performansına ilişkin araştırmalar incelendiğinde çalışma belleğinin farklı değişkenlerle birlikte ele alındığı ve programlama performansı üzerindeki tek başına ve diğer değişkenlerle birlikte etkisinin vurgulandığı görülmektedir. Çalışma belleğinin hem programlama öğretim sürecinin planlanması, hem programlamayı öğrenme süreci hem de öğrenme çıktısı (başarı/performans) ile ilişkili olduğu göze çarpmaktadır.

2.2 Uzamsal Beceriler ve Programlama Performansına İlişkin Çalışmalar

Programlama performansı ve uzamsal beceriler arasındaki ilişkiyi ele alan farklı çalışmalar bulunmaktadır.

Jones ve Burnett (2008), uzamsal becerilerden zihinsel döndürme becerisini ele alarak programlama dersine ilişkin performans ile zihinsel döndürme arasındaki ilişkiyi incelemişlerdir. Çalışma sonunda programlama performansı ile uzamsal beceri puanları arasında pozitif ve anlamlı ilişkiler bulunmuştur. Ayrıca çalışmada programlama deneyimi algısı yüksek olanların uzamsal becerilerinin de yüksek olduğu ortaya konulmuştur. Programlama dersinde kendine güvenleri yüksek bireylerin puanlarının anlamlı derecede yüksek olduğu bulunmuş fakat bu bireylerin her ne kadar uzamsal becerileri de yüksek olarak bulunsa da bu faktördeki fark anlamlı bulunmamıştır.

Vainio ve Sajaniemi (2007), programlamanın yazarlık, tasarım, programı kavrama ve hata ayıklama gibi farklı süreçleri içeren bir süreç olduğunu ifade etmiş ve hata ayıklama için programın kavranması ve ayrıntılı izlenmesini özellikle programın çalışır durumunun zihinde canlandırma becerisini gerektirdiğini ifade etmişlerdir. Diğer yandan uzamsal becerilerin hem gerçek dünyadaki gezinim hem de soyut bilgilerin bulunduğu ortamlardaki gezinimde önemli bir beceri olması ile birlikte çok-boyutlu sanal alanlarla ilişkili kod parçacıkları nedeniyle programların anlaşılmasıyla ilişkili olduğu ileri sürülmektedir (Cox, Fisher, ve O'Brien, 2005).

Jones ve Burnett (2007), yaptıkları çalışma ile programı kavrama uygulamasına ilişkin kaynak kodda gezinimde, uzamsal beceriler bakımından farklılıklar bulunmuş, yüksek uzamsal becerili bireylerin fonksiyonlara daha sık geldiğini ve ayrıca dosyalar (sınıflar arası) arasındaki geçişlerin daha çok olduğu ortaya konulmuştur. Bu tarzda gezinim ile şekillenen programlama ile daha iyi bir zihinsel model oluşturulduğu, yani daha iyi kavrama sağlandığı ifade edilmiştir (Jones ve Burnett, 2007).

Simon, Fincher, ve diğ. (2006), bilişsel, davranışsal ve tutum faktörlerinin programlama performansı ile ilişkisini incelenmişlerdir. Bilişsel ölçümler için uzamsal görselleştirme becerisini ölçen kâğıt katlama testi, davranışsal ölçümler için harita çizme testi ve arama stratejisini ifade edebilmeyi ölçen telefon

defterinde arama görevi, tutum ölçümleri için ise öğrenme yaklaşımı ölçeği kullanılmıştır. Çalışmaya 11 farklı kurumdan 177 üniversite öğrencisi katılmıştır. Çalışma programlamaya giriş dersi kapsamında yürütülmüş ve programlama performansı olarak bireylerin derse ilişkin yılsonu notları alınmıştır. Çalışma sonunda bireylerin uzamsal görselleştirme becerileri ile programlama notlar arasında orta düzeyde pozitif korelasyon bulunurken, derin öğrenme yaklaşımı ile programla başarısı arasında pozitif yüksek korelasyon, yüzeysel öğrenme yaklaşımı ve programlama performansı arasında negatif korelasyon bulunmuştur. Ayrıca ayrıntılı gezinimsel haritalar çizen bireylerin ve arama stratejilerini zengin bir şekilde dile getiren bireylerin programlamadan yüksek not olan bireyler olduğu bulunmuştur

Fisher, Cox ve Lin (2006), program koduna ilişkin bilişsel harita kavramı yerine kod alanı (codespace) kavramını kullanmışlar ve bu kavramı programlayıcının kod içinde tanımlanan varlıkların uzamsal özelliklerine ilişkin algısını temsil eden kaynak kodun zihinsel modeli olarak tanımlamışlardır. Yani zihinsel model kaynak kod içinde gezinim esnasında ortaya çıkarılan çeşitli bilgilerin bir araya getirilmesiyle programın soyut temsilidir. Fisher ve diğ. (2006), kaynak kodları doğrusal bir yapıda olduğundan dolayı kod ortamındaki gezinimde zihinsel döndürmenin diğer uzamsal beceriler kadar önemli olmadığını, fakat zihinsel döndürme ile ilgili yapılan çalışmalarda bu becerinin programlama performansı ile kağıt katlama testinin uygulandığı görselleştirme becerisine göre daha güçlü bir ilişkide olduğunu gösterdiğini ifade etmişlerdir.

Fincher ve diğ. (2005), programlama dersindeki performansı etkileyen faktörleri ortaya koymaya dönük olarak yaptıkları çalışmada genellenebilirliği arttırmak amacıyla 11 farklı kurumda programlamaya giriş dersindeki performans ile çeşitli faktörler arasındaki ilişkiyi incelemişler ve uzamsal görselleştirme becerisi ve programlama performansı arasında pozitif fakat küçük bir korelasyon olduğunu bulmuşlardır. Ayrıca, uzamsal becerinin programlama performansındaki rolü ile ilişkili olan bir faktörün de zihinsel modeller olduğunu, zihinsel modellerin ise bir programın belirleyici bir şekilde gösterimi ya da soyutlanabilmesi olduğunu ifade etmişlerdir.

deRaadt ve diğ. (2005) öğrenme yaklaşımları, uzamsal görselleştirme becerisi ve önceki programlama deneyimi faktörlerinin programlama performansına etkisini incelemişler ve çalışma sonunda programlama puanları ve öğrenme yaklaşımları arasında pozitif yüksek düzeyde ve anlamlı bir korelasyon bulunmuştur. Ayrıca önceki programlama deneyimleri de programlama performansı ile yüksek korelasyon içinde bulunurken, uzamsal görselleştirme ile programlama performansı arasında düşük fakat anlamlı korelasyon olduğu bulunmuştur.

Mayer, Dyck ve Vilberg (1986), programlama dillerini öğrenme ve düşünme becerileri arasındaki ilişkiyi incelemişlerdir. Çalışmada düşünme becerileri olarak problem dönüştürme becerileri, işlem kavrama becerileri, genel beceriler (akıl yürütme becerisi, sözel beceri ve görsel-uzamsal beceri) ve aritmetik işlem becerisi alınmıştır. Çalışmaya programlama konusunda hiç deneyimi olmayan ve mühendislik bölümü dışındaki bölümlerde öğrenim gören toplam 111 üniversite öğrencisi katılmıştır. Çalışmada deney grubundaki bireylere temel programlama dersi dışında ön test-son test olarak düşünme becerileri bataryası uygulanırken, kontrol grubu programlama eğitimi almadan sadece düşünme becerileri bataryasını ön test son test olarak almıştır. Çalışma sonunda programlama dersinin düşünme becerilerinden sadece problem dönüştürme ve işlem kavrama becerilerini arttırdığı bulunurken, diğer yandan akıl yürütme becerisi, sözel beceri, uzamsal beceri ve aritmetik işlem becerisinde herhangi bir artış gözlemlenmemiştir. Düşünme becerilerinden hangilerinin programlama performansı ile ilişkili olduğunu ortaya koyabilmek için ise regresyon analizi yapılmış ve analizler sonunda problem dönüştürme becerileri, işlem kavrama becerisi, akıl yürütme becerisi ve uzamsal becerinin programlama performansının yordanmasında anlamlı değişkenler olduğu bulunmuştur.

Irons (1982), programlama performansını; programı oluşturma, kavrama ve hata ayıklama olarak üç alt gruba ayırarak, bu alt performansların tümevarım becerisi, karşılaştırmalı akıl yürütme, ilişkisel bellek, bütünleştirme esnekliği, uzamsal tarama ve sözel zekâ olmak üzere beş farklı bilişsel beceri ile aralarındaki ilişkini incelemiştir. Çalışma sonunda uzamsal taramanın özellikle hata ayıklama alt performansını yordadığı, bunun yanı sıra akıl yürütmenin ise programı

oluştururken biçimlendirme kısmında etkili olduğu, tümevarım becerisi ve sözel zekânın ise her üç alt performansı da anlamlı derecede yordadığı bulunmuştur.

Uzamsal beceriler ve programlama performansının birlikte ele alındığı çalışmalar incelendiğinde uzamsal becerinin programlama sürecinin çeşitli aşamalarında (kaynak kodda gezinim, hata ayıklama vb.) ve/veya programlama süreci sonundaki çıktı (başarı, performans) üzerinde etkili bir faktör olduğu görülmektedir. Çalışmalarda ele alınan uzamsal beceri boyutu da uzamsal görselleştirme, uzamsal gezinim, zihinsel döndürme, uzamsal tarama gibi uzamsal becerinin farklı alt boyutları olarak farklılaşmaktadır. Genel çalışma bulguları ise, programlama sürecinde bireylerin uzamsal becerilerinin programlama performansı ile belirli bir düzeyde ilişki olduğu yönündedir.

2.3 Bilişsel Olmayan Faktörler ve Programlama Performansına İlişkin Çalışmalar (Öz yeterlilik, Cinsiyet ve Ön deneyim)

Öz yeterlilik, cinsiyet ve ön deneyim faktörlerinin programlama performansı ile ilişkine yönelik yapılan çalışmalarda farklı bulgular söz konusudur.

Lau ve Yuen (2011), programlama performansını etkileyen faktörleri ortaya koymak amacıyla yaptıkları çalışmada cinsiyet, öğrenme stili, zihinsel modeller, ön akademik beceri ve öğretim ortamı olmak üzere bir dizi etken faktörün etkisini birlikte incelemişlerdir. Çalışmaya toplam 131 lise öğrencisi katılmıştır. Çalışmada bu faktörlerin programlama performansına ilişkin etkisi en küçük kısmi kareler yöntemi kullanılarak yapısal model ile test edilmiştir. Çalışmada programlama performansına ilişkin varyansın %43,6'sı açıklanmıştır. Çalışma sonunda tüm faktörlerin dolaylı ya da doğrudan etkileri anlamlı bulunmuş, bireysel farklılıklar olan cinsiyet, öğrenme stili ve zihinsel modellerin yanı sıra ön akademik beceri ve öğretim ortamının öğrenme çıktıları üzerindeki etkisi ortaya konulmuştur.

Davidsson ve diğ. (2010), programlama dersi almanın bireylerin programlama ilişkin öz yeterlilik algısına etkisini incelemişlerdir. Öz yeterlilik algısının yanı sıra bireylerin derse ait belirtilen beş hedefi ne kadar iyi karşıladıklarına ilişkin algılarındaki değişim de incelenmiştir. Çalışmaya programlama dersini ilk defa alan 53 ikinci sınıf mühendislik öğrencisi ve daha önceden dersi almış olan 24 üçüncü sınıf öğrencisi katılmıştır. Çalışma sonunda bireylerin programlama ilişkin

öz yeterliliklerinde deęişim bulamazken “hata ayıklama”, “programı analiz etme ve yeniden tasarlama” ve “programın alıřmasına iliřkin temel ilkeleri tanımlayabilme” olmak üzere derse iliřkin üç hedefin karřılanmasına yönelik bireysel algıda artış gözlemlenmiřtir.

Ařkar ve Davenport (2009) yaptıkları alıřmada programlamaya iliřkin öz-yeterlilik algısının cinsiyete, bölüme, ön bilgisayar bilgisine, bilgisayar kullanım sıklığına ve ailenin bilgisayar kullanımına göre farklılık gösterip göstermediğini incelemiřlerdir. alıřmaya toplam 326 üniversite öğrencisi katılmıřtır. alıřma sonunda erkeklerin programlamaya iliřkin öz yeterlilik algısı kadınlara göre daha yüksek bulunurken, bilgisayar mühendislięi bölümü öğrencilerinin programlamaya iliřkin öz yeterlilik algısı da dięer bölümlere göre daha yüksek olarak bulunmuřtur. Ayrıca bilgisayar kullanımına iliřkin deneyim yılının, bilgisayar kullanım sıklığının ve aile üyelerinin bilgisayar kullanımının programlama öz-yeterliliğini anlamlı řekilde yordadığını bulmuřlardır.

Jegede (2009), programlaya iliřkin öz yeterlilik algısı ile bilgisayara iliřkin arka plan arasındaki iliřkiyi incelemiřtir. Bilgisayar arka planı olarak ise programlama deneyimi, programlama dersi geme notları, kaç yıldır program yazıldığı ve programlamaya iliřkin alınan ders sayısı alınmıřtır. alıřmaya altı farklı mühendislik bölümünden toplam 192 üniversite öğrencisi katılmıřtır. alıřma sonunda sadece programlama derslerinden alınan geme notları ortalaması ve ders sayısının programlamaya iliřkin öz-yeterlilięi ile anlamlı iliřkide olduęu bulunmuřtur. Dięer yandan programlama deneyimi ve programlama deneyimine iliřkin yıl sayısının programlama öz yeterlilięi ile iliřkisi anlamlı olarak ıkmamıřtır..

Ramalingam ve dię. (2004) ise önceki deneyimler, öz yeterlilik ve zihinsel modellerin programlama performansı üzerindeki etkisini inceledikleri alıřmada bireylerin önceki deneyimlerinin ve zihinsel modellerinin öz yeterlilikleri ile doğrudan iliřkili olduęunu ve iyi oluřturulmuř zihinsel modellerle yüksek öz yeterlilik algısının birlikte programlama performansını anlamlı řekilde yordadığını ortaya koymuřlardır.

Cegielski ve Hall (2006) iyi bir programcının özelliklerini arařtırdıkları alıřmada bu özellikleri deęer inanları, biliřsel beceriler ve kiřilik özellikleri olmak üzere üç

temel kategoride toplayarak bir model oluşturmuşlardır. Bu modelde bireylerin değer inançlarının kişilik özellikler ve bilişsel becerilerini etkileyeceği, bilişsel beceriler ve kişilik özelliklerinin ise programlama performansının yordayıcısı olacağı varsayılmıştır. Çalışmada kişilik özellikleri olarak kendine saygı, öz yeterlilik algısı, kontrol odağı ve nevrozizm olarak dört temel faktör ele alınmıştır. Çalışma nesne-tabanlı programlama dersine kaydolmuş 139 üniversite öğrencisi ile yürütülmüştür. Kurulan modelin test edilmesi sonucu hem değer inancının hem bilişsel becerilerin hem de bireyin kişilik özelliklerini oluşturan kendine saygı, öz yeterlilik algısı, kontrol odağı ve nevrozizmin programlama performansının anlamlı bir yordayıcısı olduğu bulunmuştur.

Byrne ve Lyons (2001), öğrencilerin birinci yıl programlama performansı ile cinsiyet, bilgisayar ön deneyimi, öğrenme stili ve akademik performans arasındaki ilişkiyi incelemişlerdir. Çalışma sonunda erkekler ve kadınlar arasında anlamlı fark bulunmamış fakat fen ve matematik puanları ve programlama sınavı sonuçları arasında anlamlı korelasyon bulunmuştur. Diğer yandan öğrenme stilleri ve programlama arasındaki ilişkide anlamlı olarak ortaya çıkmamıştır.

Bergin ve Reilly (2005) nesne-tabanlı programlama dersinde performansı belirleyen faktörleri ortaya koymaya dönük akademik deneyim (fizik, kimya, matematik, biyoloji), bilgisayar ön deneyimi, bireyin kendi programlama performansına ilişkin algısı, kendini derste rahat hissetme seviyesi ve bilişsel becerilerin etkisini incelemişlerdir. Çalışmaya programlamaya giriş dersini alan 96 üniversite öğrencisi katılmıştır. Verileri araştırmacılar tarafından geliştirilen bir anket ve ilişkisel becerileri ölçmeye dönük bir bilişsel testle toplanmıştır. Çalışma sonunda kendini derste rahat hissetme seviyesi, matematik ve fen bilimi puanları programlama performansı ile oldukça yüksek korelasyonda bulunurken bilişsel becerilerle olan korelasyon ise düşük bulunmuştur. Analizler sonucunda cinsiyet, matematik puanı, derste kendini rahat hissetme seviyesi ve bireyin kendi performansına ilişkin algısı programlama performansının %79'unu açıklamıştır.

Grant (2003), yapısal programlama ve nesne yönelimi programlama öğretiminin bireylerin eleştirel düşünme becerileri üzerindeki etkisini ve bu iki programlama dersindeki başarıda öğrenme stili ve cinsiyete göre bir farklılık olup olmadığını incelemişlerdir. Çalışmada yapısal programlama dersi C programlama dili

üzerinden verilmiş olup bu grupta toplam 17 kişi yer alırken, nesne yönelimli programlama dili C++ dili üzerinden verilmiş olup bu grupta ise toplam 24 kişi yer almıştır. Bireylerin dersten önce ön bilgileri, eleştirel düşünme becerileri ve öğrenme stilleri belirlenmiştir. Dönem sonunda bireylerin programlama dersindeki puanları alınmış ve ayrıca eleştirel düşünme becerileri tekrar ölçülmüştür. Çalışma sonunda yapısal programlama ve nesne yönelimli programlamanın eleştirel düşünme becerileri üzerindeki etki anlamlı olarak bulunamamıştır. Ayrıca bireylerin programlama performansları üzerinde öğrenme stili ve cinsiyetin etkisi de anlamlı olarak bulunamamıştır.

Erdoğan (2005), programlama başarısına etki eden faktörleri belirlemek üzere cinsiyet, lise türü, genel akademik başarı, genel yetenek, matematik başarı ve bilgisayara karşı tutum değişkenlerini ele almıştır. Çalışma Bilgisayar ve Öğretim Teknolojileri Eğitimi Bölümü'ne devam eden 60 öğrenci yürütülmüştür. Çalışmada ilişkisel istatistiklerle ele alınan bağımsız değişkenlerle programlama başarısı arasındaki ilişki incelenmiştir. Çalışma sonunda sadece matematik başarı ve genel akademik ortalama ile programlama başarısı arasında anlamlı bir ilişki olduğu bulunmuştur. Diğer yandan cinsiyet, mezun olunan lise türü, genel yetenek (sayısal, sözel ve soyut) ve tutum ile programlama başarısı arasında ilişki bulunmamıştır.

Pillay ve Jugoo (2005), acemi bireylerin ilk programlama deneyimlerinde genellikle zorluk yaşadığından yola çıkarak programlama performansını olumsuz etkileyen faktörleri ortaya koymayı amaçlamışlardır. İlk defa programlama dersi alan iki farklı üniversiteden toplam 97 öğrenci ile çalışılmıştır. Çalışmada programlama performansı ile ilişkili olduğu düşünülen cinsiyet, ön deneyim, öğrenme stili ve problem çözme becerilerinin etkisi incelenmiştir. Çalışma sonunda programlama performansının ön deneyim ve cinsiyetten bağımsız olduğu bulunurken, problem çözme becerisi programlama performansı ile pozitif ve anlamlı ilişkiye sahip olarak bulunmuştur. Diğer yandan öğrenme stiline etkisi üniversitelere göre farklılaştığından öğrenme stiline ilişkin net bir bulgu elde edilememiştir.

Lau ve Yuen (2009), öğrenme stilleri ve cinsiyetin bireylerin farklı seviyelerdeki programlama performansı üzerindeki etkisini önceki akademik başarının etkisi ile birlikte incelemişlerdir. Çalışmada programlama performansı bildirimsel

programlama bilgisi, işlemsel programlama bilgisi, durumsal programlama bilgisi ve stratejik programlama bilgisi (en üst düzey) olarak 4 farklı boyutta ele alınmıştır. Programlama performansı ölçülürken bu dört bilgi tipi temel alınarak, bildirimsel programlama bilgisine dayalı üç, işlemsel bilgiye dayalı dört tane çoktan seçmeli soru, durumsal bilgiye dayalı dört boşluk doldurma sorusu ve stratejik bilgiye dayalı iki tane probleme dayalı program yazma sorusu içeren bir test kullanılmıştır. Öğrenme stilleri ise öğrenme stili ölçeği ile belirlenmiştir. Çalışmaya dokuz farklı okuldan toplam 217 ortaöğretim öğrencisi katılmıştır. Tüm ölçümler çevrimiçi ortamda yapılmıştır. Bireylerin geçmiş akademik başarıları ise kontrol değişkeni olarak alınmıştır. Verilerin analizinde çok değişkenli varyans analizi kullanılmıştır. Analizler sonucunda önceki akademik başarının ve öğrenme stilinin programlama performansı üzerinde anlamlı etkisi bulunurken, cinsiyetin temel etkisi de etkileşim etkisi de anlamlı olarak bulunmamıştır. Hem soyut hem de somut ardışık öğrenen bireyler tüm programlama performanslarında rastgele öğrenme stiline sahip bireylerden daha başarılı olarak bulunmuştur. Ayrıca geçmişteki akademik başarı tüm programlama performansı ölçümlerinde ve tüm öğrenme stillerinde anlamlı performansın anlamlı bir yordayıcısı olarak bulunmuştur.

Wilson ve Shrock (2001), üniversitedeki programlamaya giriş derslerinde hangi faktörlerin başarıyı etkilediğini incelemek üzere yaptıkları çalışmada, programlama performansı ile ilişkili olabilecek değişkenleri modellerken, matematik arka planı, başarı/başarısızlığa atfetme, öz yeterlilik algısı, teşvik, rahatlık seviyesi, çalışma stili tercihi, programlama ön deneyimi, programlama dışındaki bilgisayarla ilişkili ön deneyim ve cinsiyet olmak üzere toplam dokuz değişkeni ele almışlardır. Toplam 105 programlamaya giriş öğrencisi çalışmaya katılmıştır. Çalışma sonunda rahatlık seviyesi, matematik ve başarı/başarısızlığa atfetme programlama performansını yordayan en önemli üç değişken olarak ortaya çıkarken, cinsiyet, ön deneyim, çalışma stili, öz yeterliliğin tüm model içinde anlamlı bir etkisi bulunmamıştır.

Yurdugül ve Aşkar (2013) programlama bilgisini kavramsal bilgi, sözdizimsel bilgi ve stratejik bilgi olmak üzere üç boyutta ele alarak, programlama bilgisine ilişkin bilişsel gelişim üzerinde problem çözme becerileri ve cinsiyetin etkisini gelişimsel bir yöntemle incelemiştir. Çalışmaya programlama dersi alan, toplam 86

Bilgisayar ve Öğretim Teknolojileri Öğretimi Bölümü öğrencisi katılmıştır. Verilerin analizinde örtük büyüme modeli kullanılmıştır. Araştırma sonuçlarına göre hem stratejik programlama bilgisi kazanımında hem de kavramsal bilgi kazanımında erkeklerin lehine gelişim bulunmuştur. Ayrıca problem çözme becerisi yüksek bireylerin tüm bilgi türlerinde programlama performanslarında daha yüksek erişimi sağladığı bulunmuştur.

Wiedenbeck (2005), programlamaya giriş dersinin genel olarak zor olarak algılandığından yola çıkarak, buna neden olan faktörleri ortaya koymaya dönük bir model kurarak, bu modeli test etmiştir. Modelde programlama dersindeki başarıya etki eden faktörler programlamaya ilişkin ön deneyim, öz yeterlilik algısı ve bilginin organizasyonu olarak alınmıştır. Modelde programlamaya ilişkin ön deneyimin öz yeterlilik algısını ve bilginin organizasyonunu etkileyeceği, öz yeterlilik algısı ve bilginin organizasyonunun ise birlikte programlama performansını belirleyeceği şeklinde hipotezler oluşturulmuştur. Çalışmaya bilgisayar bilimi alanında olmayan bölümlerden 120 üniversite öğrencisi katılmıştır. Çalışmada programlama performansı olarak geçme notlarının yanı sıra hata ayıklama performansı alınmıştır. Çalışma sonunda, modelin test edilmesi ile ön deneyimin öz yeterlilik algısını anlamlı şekilde yordarken bilginin organizasyonu ile anlamlı bir ilişki olmadığı bulunmuştur. Hem bilginin organizasyonu hem de öz yeterlilik algısı programlama performansının anlamlı bir yordayıcısı olarak bulunmuştur.

Milic (2009), programlamaya ilişkin bilgiyi düşük seviyede (bir program kodunun okunması) ve yüksek seviyede bilgi (program kodunun değiştirilerek yazılması) olmak üzere iki farklı seviyede ele alarak, bilgi seviyesi ile bireyin zekası, tutumu ve cinsiyeti arasındaki ilişkiyi incelemiştir. Çalışmada programlama performansı olarak yüksek ve düşük seviyede programlama bilgisi ölçen iki ayrı bilgi testi kullanılmış olup bu testler araştırmacılar tarafından dersin içeriğine uygun olarak geliştirilmiştir. Çalışmaya toplam 79 lise öğrencisi katılmıştır. Araştırma sonunda çalışmada özellikle yüksek düzeydeki programlama bilgisinin düşük düzeydeki programlama performansına bağlı olduğu, ayrıca bireylerin zekâlarının ve bilgisayara karşı tutumlarının yine yüksek düzeydeki programlama bilgisinin anlamlı yordayıcısı olduğu bulunurken, cinsiyetin programlama bilgi düzeyinde herhangi bir etkisi bulunmamıştır.

Yukarıda kısaca özetlenen arařtırmalar incelendiğinde cinsiyet, ön deneyim ve öz yeterlilik algısının programlama performansı ile ilişkisinin birden çok deęişkenle birlikte ele alınarak sonuçların ortaya konulduğu görölmektedir. Öz yeterlilik algısının ele alındığı çalışmalarda genel olarak programlama performansı ile ilişkisi bulunmuş ve programlama performansına etkisine ilişkin önemi vurgulanmıştır. Ön deneyim ve cinsiyet ise daha önceki kısımda incelenen bilişsel beceriler ile ilgili olan arařtırmalar da dâhil olmak üzere hemen hemen tüm çalışmalarda ele alınarak, dięer deęişkenlerle birlikte olan etkileri, doğrudan ilişkileri ya da kontrol edildiği durumlardaki etkileri incelenen deęişkenler olarak ortaya çıkmaktadır. Bu iki deęişkenin programlama performansı üzerindeki etkileri ve başarı ile olan ilişkileri noktasında ise farklı bulgular söz konusudur. Çalışmaların bir kısmında cinsiyet programlama performansında etkili bir deęişken olarak bulunurken bir kısmında ise cinsiyete göre anlamlı farklılıklar bulunmamaktadır. Dięer yandan programlama ön deneyiminin de özellikle ele alınış biçiminde ve etkisine ilişkin farklı bulgular ortaya konulmuştur. Programlama ön deneyimi çalışmalarda önceki programlama bilgisi, program yazma yılı, programlamaya ilişkin alınan ders sayısı, bilinen programlama dilleri sayısı, yazılan program sayısı gibi ele alınış biçimiyle farklılaşmaktadır. Benzer şekilde programlama ön deneyiminin programlama performansı üzerindeki etkisine ilişkin de bulgular farklılaşmaktadır.

İncelenen arařtırmaların büyük bir kısmında üniversite öğrencileri ile çalışıldığı ve programlamaya giriş dersinin seçildiği dikkat çekmektedir.

3. YÖNTEM

Çalışmanın bu kısmında araştırma modeli, çalışma grubu, veri toplama süreci, veri toplama araçları, ölçme araçlarının puanlanması ve verilerin analizi başlıklarına yer verilecektir.

3.1 Araştırma Modeli

Bu çalışma betimsel ilişkisel yöntemler ve veri madenciliğine dayalı tahminleyici parametrik olmayan yöntemlerden “Arttırılmış Regresyon Ağaçları” ve “Rastgele Ormanlar” yöntemleri temelinde modellenmiştir. Programlama performansının bilişsel ve bilişsel olmayan değişkenlerle olan ilişkisi incelenmiş, ardından programlama performansının yordanmasına ilişkin kurulan model öncelikle “Arttırılmış Regresyon Ağaçları” ile ardından modelin geçerliliğini test etmek ve açıklama oranlarını karşılaştırarak en uygun modeli seçmek için “Rastgele Ormanlar” yöntemi ile test edilmiştir.

3.2 Çalışma Grubu

Çalışma gurubunu Hacettepe Üniversitesi, Ankara Üniversitesi ve Gazi Üniversitesi olmak üzere toplam üç farklı üniversitedeki Bilgisayar ve Öğretim Teknolojileri Öğretmenliği (BÖTE) bölümü ikinci sınıf öğrencilerinden “Programlama Dilleri-I” dersine kaydolan toplam 129 lisans öğrencisi oluşturmaktadır. Programlama Dilleri-I dersi BÖTE bölümlerinde programlamaya ilişkin alınan ilk ders olup, üçüncü dönemde verilmektedir.

Öğrencilerin tümü 18-25 yaş aralığında ve lisans öğrenimlerindeki ilk programlama dersini almış bireylerden oluşmuştur. Çalışmaya katılımında gönüllülük esas tutulmuştur. Katılımcıların üniversite, cinsiyet ve programlamaya ilişkin ön deneyime sahip olup olmama durumlarına ilişkin demografik bilgiler Çizelge 3.1’de verilmiştir.

Çizelge 3.1. Çalışma grubuna ait demografik bilgiler

		N	%
Cinsiyet	Kadın	69	53,5
	Erkek	60	46,5
Üniversite	Hacettepe Ün.	63	48,8
	Ankara Ün.	37	28,7
	Gazi Ün.	29	22,5
Ön Deneyim	Var	71	55
	Yok	58	45

3.3 Veri Toplama Süreci

Veriler üç farklı üniversitede ayrı ayrı gerçekleştirilmiştir. Programlama Dilleri-I dersi BÖTE bölümlerinin tümünde 3. yarıyılıda verildiği için veriler tüm üniversitelerde aynı eğitim öğretim döneminde toplanmıştır. Veri toplama tüm katılımcılarda aynı araştırmacı tarafından tek bir bilgisayarla bireysel olarak gerçekleştirilmiştir.

Verilerin toplanmasına ilk olarak Hacettepe Üniversitesi'nde başlanmıştır. BÖTE 2. sınıf öğrencilerine genel duyuru yapılarak çalışma hakkında bilgi verilmiş ve çalışmada gönüllüğünün esas olduğu bildirilmiştir. Bu doğrultuda çalışmaya katılmak isteyen öğrencilerle bir randevu listesi oluşturulmuş, uygulamaya ilişkin mekan ve süre ile ilgili bilgi verilmiştir.

Uygulama öğrencilerle birebir ve bireysel gerçekleştirilmiştir. Her bir öğrenci ile bire bir yürütülen uygulama oturumu ortalama 25 dk. ve 30 dk. arasında değişmiştir. Uygulama oturumunda ilk olarak bireylere çalışma hakkında ve uygulama sürecine ilişkin kısa bir bilgi verilmiştir, eğer varsa sorular alınmıştır. Öğrencinin uygulamaya hazır olması halinde, öğrenci bilgisayar başına alınarak teste ilişkin yazılım açılmıştır. İlk olarak uzamsal yönelim testi ile başlanmış ve teste ilişkin yönergelerin yer aldığı ekranlar uygulamacı tarafından ekstra açıklamalar yapılarak ilerlenmiştir. Yönergeler tamamlandıktan sonra testin nasıl

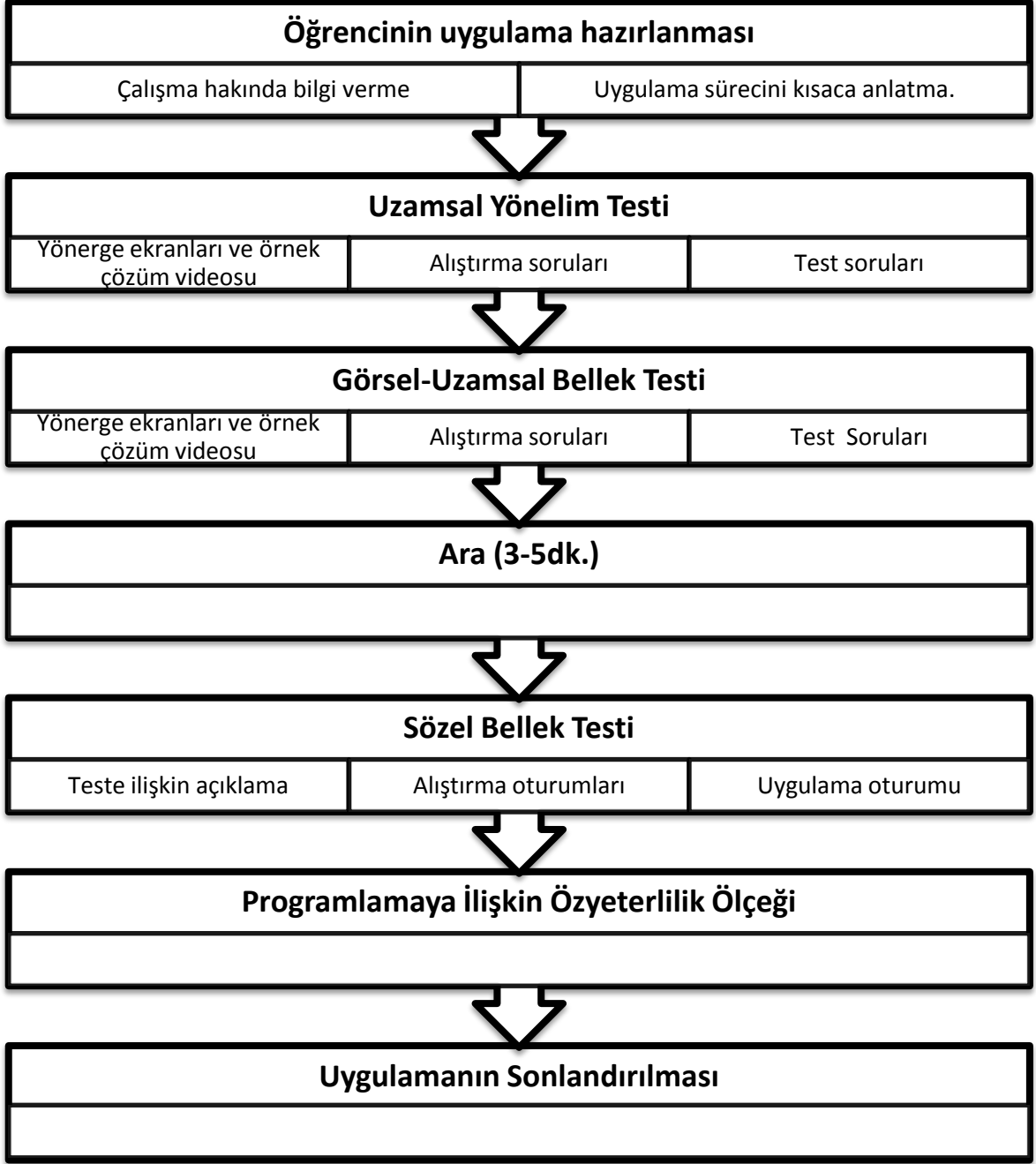
yapılacağı anlaşılmadıysa tekrar edilmiş, anlaşıldıysa alıştırmaya sorularına geçilmiştir. Alıştırma soruları tamamlandıktan sonra ise test soruları başlamıştır. Testteki sorular için herhangi bir zaman sınırlaması olmadığı, fakat tepki sürelerinin kayıt edildiği belirtilmiştir. Uzamsal yönelim testi tamamlandıktan sonra hemen ardından görsel-uzamsal çalışma belleği testine geçilmiştir. Görsel uzamsal bellek testi de benzer şekilde yönergeler ekranı araştırmacı tarafından anlatılarak ilerlenmiş, uygulamadan örnek bir soru çözümüne ilişkin video izlettirilmiş ve alıştırmaya sorularına gelinerek alıştırmalar kısmında katılımcıya bırakılmış, ardından katılımcının testi anladığını onaylaması doğrultusunda test kısmına geçilmiştir. Görsel-uzamsal bellek testinden sonra dinlenmek üzere, kısa bir ara verilmiştir.

Son olarak sözel çalışma belleği testi için katılımcıya hazır olup olmadığı sorularak, teste geçilmiştir. Sözel çalışma belleği testi için yine araştırmacı tarafından gerekli açıklamalar yapılarak, alıştırmaya oturumları başlatılmış ve ardından asıl test uygulamasına geçilmiştir. Sözel çalışma belleği testi de tamamlandıktan sonra katılımcıya bilgisayar tabanlı testlerin bittiği haber verilmiş ve programlamaya ilişkin öz-yeterlilik ölçeği doldurmaları istenmiştir. Katılımcının ölçeğini doldurmasının ardından teşekkür edilerek uygulama oturumu tamamlanmıştır.

Hacettepe üniversitesinin ardından veri toplama sürecine Gazi Üniversitesi'nde devam edilmiştir. Gazi Üniversitesi'ne gidilerek programlama dersinin öğretim üyesi ile birlikte BÖTE 2. sınıf öğrencilerine çalışmaya ilişkin duyuru yapılmış ve katılmaya gönüllü olan öğrencilerle bir randevu çizelgesi oluşturulmuştur. Veri toplama süreci yine katılımcıya araştırmacının bilgisayar tabanlı testleri uygulaması şeklinde bireysel olarak gerçekleştirilmiştir.

Son olarak Ankara Üniversitesi'nde benzer şekilde BÖTE 2. sınıf öğrencilerine duyuru yapılarak, gönüllü öğrencilerle bir randevu çizelgesi oluşturulmuş, araştırmacının ve öğrencilerin uygun olduğu saatler belirlenerek uygulamalar gerçekleştirilmiştir.

Uygulama sürecinin aşamalarına ilişkin akışı temsil eden bir grafik Şekil 3.1'de verilmiştir.



Şekil 3.1. Uygulama sürecine ilişkin akış diyagramı (ortalama 25 dk. - 30 dk.)

3.4 Veri Toplama Araçları

Çalışmada demografik bilgi formu, uzamsal yönelim testi, görsel-uzamsal bellek testi, sözel bellek testi, programlamaya ilişkin öz yeterlilik algısı ölçeği ve programlama performansı notları kullanılmıştır

3.4.1 Uzamsal Yönelim Testi

Çalışmada katılımcıların uzamsal yönelim becerilerini ölçmek için Kozhevnikov ve Hegarty (2001) tarafından geliştirilen, Mazman ve Altun (2013) tarafından revize edilerek Türk örnekleminde üniversite öğrencileri ile norm çalışması yapılmış olan bilgisayar tabanlı “Uzamsal Yönelim Testi” kullanılmıştır. Testin norm çalışması 52 (%51,5)’si kadın 49 (48,5)’u erkek olmak üzere, 18-25 yaş aralığındaki toplam 101 üniversite öğrencisi ile yapılmıştır. Testin test tekrar test güvenilirlik katsayısı doğruluk puanları için ($r = .778$, $p < .001$), tepki süreleri için ise ($r = .634$, $p < .001$) olarak bulunmuştur. Katılımcıların doğru puan ortalamaları 14 üzerinden 12,56 ($\bar{X} = 12,56$; $ss = 1,87$) olarak bulunurken, tepki süresine ilişkin ortalamaları ise 17,975 saniye ($\bar{X} = 17975$ ms.; $ss = 9918$ ms.) olarak bulunmuştur.

Test E-prime 2.0 yazılımı ile geliştirilmiş olup toplamda 14 soru yer almaktadır.

Bireyler bilgisayar ekranında her seferinde bir nesnelere dizisi görmektedir. Bu dizide toplam sekiz nesne (sandalye, araba, kedi, koltuk, trafik lambası, köpek, dur işareti ve otobüs) olup, nesnelere bir halka oluşturacak şekilde dizilmiştir. Bu nesnelere merkezinde ise bir karakter kafası yer almaktadır. Bu karakterin yüzü her seferinde dizinin tam orta sağında yer alan kediye dönük olarak ekrana gelmektedir. Tüm nesnelere ve karakter kafası siyah beyazdır.

Bireylerden kendilerini merkezdeki karakter olarak hayal etmeleri ve yüzlerini belirtilen bir nesneye saat yönünde dönmeleri ve belirtilen nesne tam karşıdayken bir diğere üçüncü nesnenin yerini gösteren kutucuğuna cevap anahtarından işaretlenmesi istenmektedir.

Soruya ilişkin yönergenin ilk kısmı ekranın üstünde ve hemen altında nesnelere dizisi resmi ile birlikte gelmektedir. Yönergenin ilk kısmında “Merkezdeki figür olduğunuzu hayal edin”, “Şimdi saat yönünde yüzünüzü’ya dönün” ifadeleri yer almaktadır. Yönergenin bu kısmı ve nesnelere dizisi ekranda 6 sn. kalmaktadır.

Bu esnada bireylerin kendilerini merkezdeki figür olarak hayal ederek, zihinlerinde yüzlerini belirtilen nesneye doğru dönmeleri beklenmektedir. 6 sn. sonra sorunun devamı olan “şimdi tam karşınızdayken’yı gösteren kutucuğu işaretleyin” ifadesi ve bireylerin cevaplarını işaretleyecekleri cevap anahtarı gelmektedir. Bireylerden, soruda istenilen nesnenin konumunu gösteren kutucuğu işaretlenmesi beklenmektedir. Cevap anahtarı, bir çember etrafında 8 temel yönün (0^0 , 45^0 , 90^0 , 135^0 , 180^0 , 225^0 , 270^0 , 315^0) bulunduğu noktada boş kutucukların yer aldığı ve merkezden gelen okun işaret ettiği 0 noktasında ise bireyin yüzünü dönmesinin istenildiği nesne ile merkezde bireyin kendisini hayal etmesi istendiği figürün bulunduğu bir şekilden oluşmaktadır.

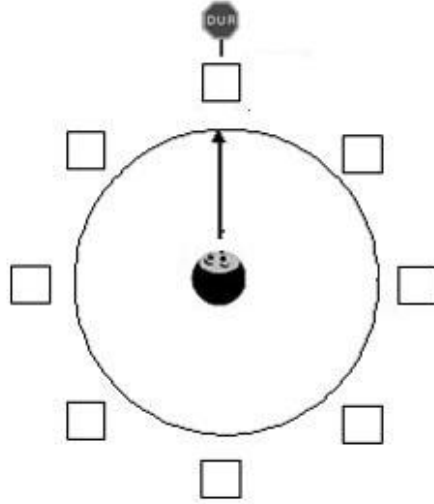
Test yönerge ekranları (EK-1), örnek soru çözüm video ekranı, üç alıştırmaya sorusundan oluşmaktadır.

Teste ilişkin bir sorudan örnek görüntü Şekil 3.2.’de verilmiştir.

Merkezdeki şeklin siz olduğunuzu hayal edin.
Şimdi saat yönünde yüzünüzü **dur işareti**ne dönün.



Şimdi dur işareti tam karşınızdayken **otobüsü** gösteren kutucuğu işaretleyin



Şekil 3.2. Uzamsal yönelim testi örnek ekran görüntüsü

Test puanlarının hesaplanmasında hem her bir soru için doğruluk puanı (1-0) hem de tepki süreleri birlikte ele alınmaktadır.

3.4.2 Görsel-Uzamsal Bellek- Sayı Döndürme Testi

Çalışmada kullanılan “Görsel - Uzamsal Bellek-Sayı Döndürme Testi” (Blasko, Holliday-Darr, Mace, ve Blasko-Drabik, 2004) tarafından Görselleştirmeyi

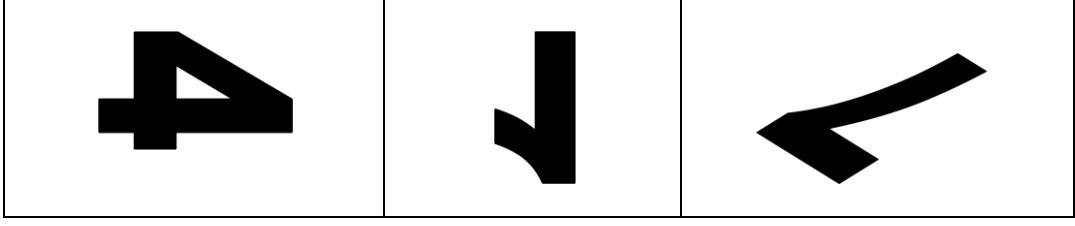
Değerlendirme ve Eğitim Projesi (VIZ) kapsamında uzamsal performansı değerlendirme, inceleme ve geliştirme amaçlı olarak geliştirilen web tabanlı uygulamadaki (<http://viz.bd.psu.edu>) uzamsal bellek - harf döndürme testindeki soruların revize edilerek modifikasyonu ile araştırmacılar tarafından yeniden geliştirilmiştir.

Testin ilk olarak Türk örneklemindeki norm çalışması yapılmıştır. Norm çalışması için uygulamaya 52 (%51,5)'si kadın 49 (48,5)'u erkek olmak üzere, 18-25 yaş aralığındaki toplam 101 üniversite öğrencisi katılmıştır. Testin güvenilirliği için test tekrar test güvenilirlik katsayısı zihinsel döndürme puanları ve uzamsal bellek puanları olmak üzere testin iki alt boyutu için hem doğruluk hem de tepki süreleri bakımından incelenmiştir. Zihinsel döndürme doğruluk puanı için test tekrar test güvenilirlik katsayısı ($r = .740$, $p < .001$) olarak, tepki süresi için ise ($r = .575$, $p < .001$) olarak bulunmuştur. Konumsal bellek doğruluk puanı için test tekrar test güvenilirlik katsayısı ($r = .481$, $p < .001$) olarak, tepki süresi için ise ($r = .788$, $p < .001$) olarak bulunmuştur. Testte zihinsel döndürme puanları 27 üzerinden ortalama ($\bar{X} = 20,65$; $ss = 3,8$) bulunurken, tepki süresi ise ortalama 9,6 sn. ($\bar{X} = 96855,4$ ms.; $ss = 31227,6$) olarak bulunmuştur. Konumsal bellek puanları ise 9 üzerinden 5,49 ($\bar{X} = 5,49$; $ss = 1,91$) olarak, ortalama tepki süresi ise 7 sn. olarak ($\bar{X} = 7042,77$ ms.; $ss = 2505,58$) bulunmuştur.

Test E-Prime 2.0 yazılımı ile geliştirilmiş olup, toplamda 3 setten oluşmaktadır. Soruların zorluk seviyesi uzamsal bellek açısından her sette artmaktadır. Testte bazı harfleri zihinde döndürme ve döndürülen harflerin yönlerinin konumlarını akılda tutma olarak iki farklı alt görev bulunmaktadır.

Orijinal testten farklı olarak harfler yerine (F, J, P, L, R), tepe noktası belirgin olan sayılar (2, 7, 1, 4) kullanılmış, ayrıca test dört zorluk seviyesi yerine üç zorluk seviyesinden oluşan setler şeklinde tasarlanmıştır.

Testte her bir ekranda her seferinde bir sayının düz ya da ayna görüntüsü verilmektedir. Ayrıca sayıların tepe noktası sekiz farklı yönelimden (45, 90, 135, 180, 225, 270, 315, 360 derece) birinde döndürülmüş şekilde görünmektedir. Örneğin her defasında ekranda Şekil 3.3.'teki gibi bir sayı görüntüsü, tek olarak gelmektedir.

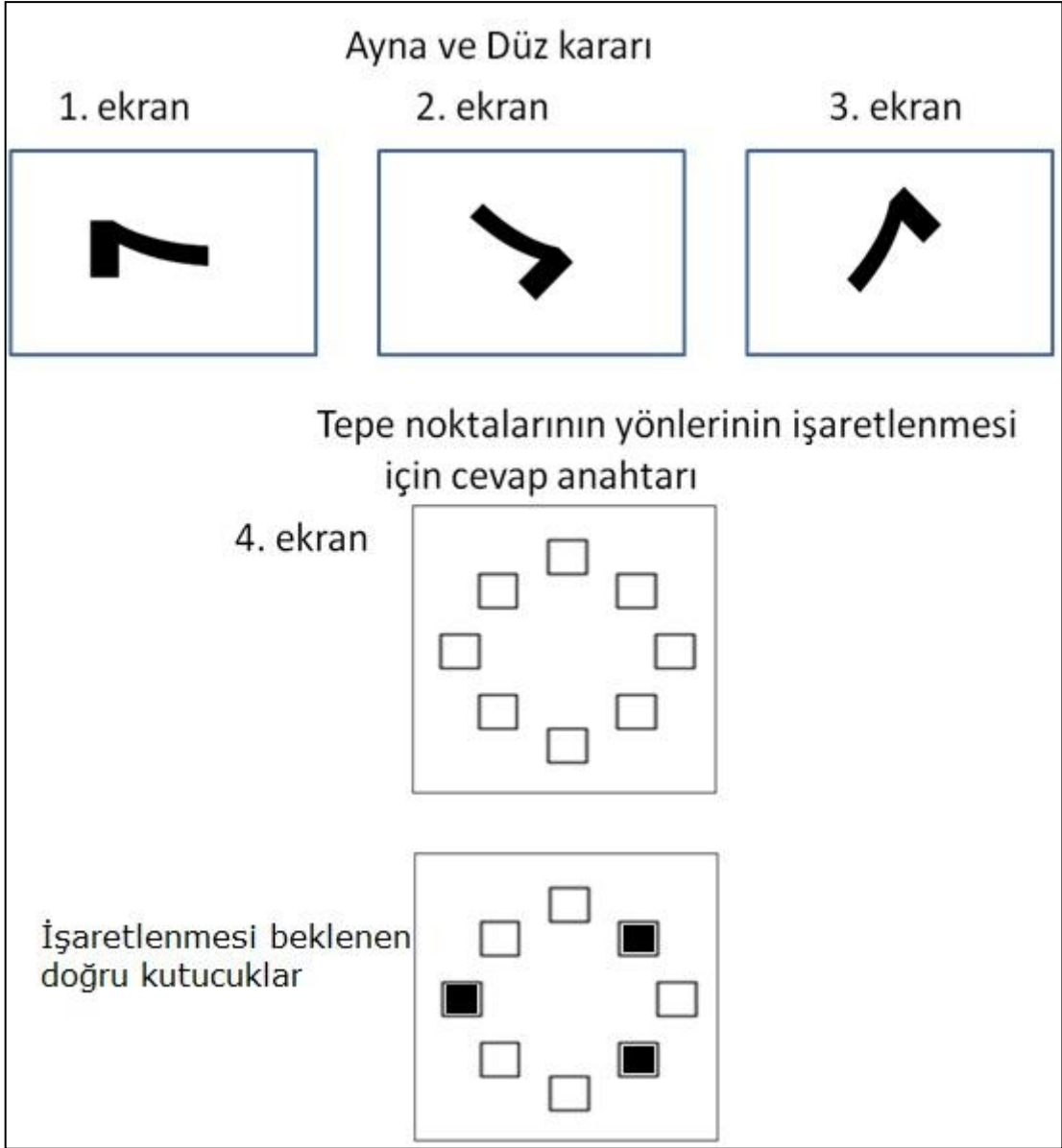


Şekil 3.3. Görsel-Uzamsal bellek testi örnek sayı görüntüleri

Katılımcıdan beklenen ise ilk olarak görüntünün düz görüntü mü ayna görüntüsü mü olduğuna karar vermesidir. Eğer görüntü sayının düz görüntüsü ise klavyeden “D”, ayna görüntüsü ise klavyeden “A” tuşuna basması istenmektedir. Bu kısım bireylerin zihinsel döndürme yaptığı aşamadır. Her bir ekranda hem ayna/düz kararlarına ilişkin tepki süreleri hem de doğruluk puanları kaydedilmektedir.

Bireylerden sayıların ayna mı düz mü görüntü olduğuna karar verirken aynı zamanda gördükleri sayıların tepe noktalarının hangi yönü gösterdiğini unutmamaları da istenmektedir. Çünkü ilgili setteki ayna/düz kararları bittikten sonra, ekrana gelen cevap anahtarında geçen sayıların tepe noktalarının hangi yönü gösterdiğine ilişkin kutucukları sırası önemli olmaksızın işaretlemeleri gerekmektedir. Test ikişerli set ile başlamakta ve ikişerli sette üst üste iki sayı geldikten sonra cevap anahtarı çıkarken, üçerli sette üç sayıdan sonra dörderli sette ise dört sayıdan sonra cevap anahtarı ekrana gelmektedir. Bu şekilde her bir soru setinde hatırlanması gereken yön sayısı bir artmaktadır. Testin bu kısmı ise uzamsal bellek becerisi gerektiren kısımdır. Bu sorularda da yine katılımcının hem tepki süresi hem de doğruluk puanları kaydedilmektedir. Uzamsal bellek sorusuna verilen yanıtın doğru olması için tüm yönlerin doğru olarak işaretlenmesi gerekmektedir. Örneğin üçerli sette her bir soruda üç sayı geçtikten sonra işaretlenen yönlerden ikisi doğru biri yanlış ise soru yanlış olarak kabul edilmektedir.

Üçerli sete ilişkin bir soruda yer alan toplamda dört ekran görüntüsü Şekil 3.4'te verilmiştir.



Şekil 3.4 Görsel uzamsal bellek testi örnek soru ekranları

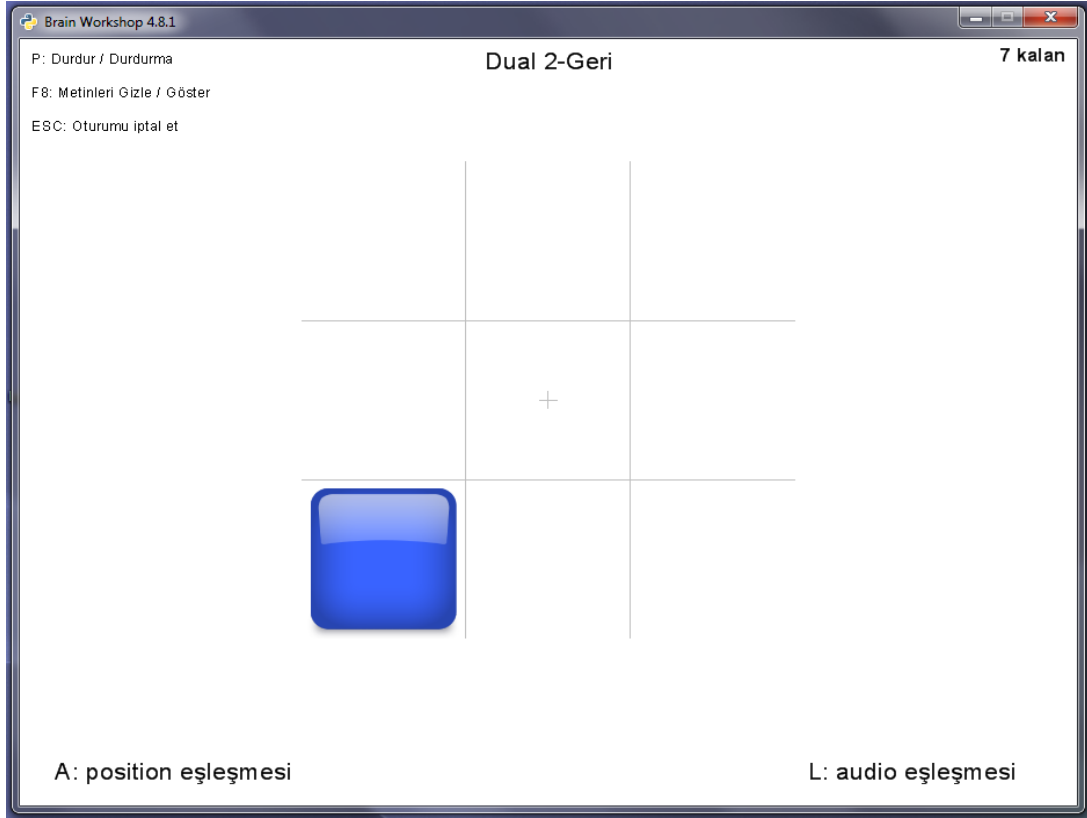
Test, yönerge ekranları (Ek-2), örnek çözüm videosu, iki alıştırmaya sorusu ve test sorularından oluşmaktadır. Toplamda zorluk seviyesi bakımından üç set, her sette de toplamda üç soru vardır. Bu şekilde zihinsel döndürme yapılan rakam sayısı $27 (2*3 + 3*3 + 4*3)$, sayıların tepe noktalarının yönlerinin hatırlanarak işaretlenmesi gereken soru sayısı ise 9'dur.

3.4.3 Sözel Çalışma Belleği Testi

Bireylerin sözel çalışma belleklerinin ölçümü "Brain Workshop" adlı n-geri görev yazılımı ile yapılmıştır. Yazılım Python programlama dili ile (Hoskinson, 2012) tarafından geliştirilmiştir. Yazılım açık kaynak kodlu olduğundan, ara yüzü ve ses

dosyaları tarafından (Çevik, 2012) Türkçeye çevrilerek derlenmiş ve kullanıma hazır hale getirilmiştir. Türkçe standardizasyon çalışmasına 115 üniversite öğrencisi katılmış, katılımcılar testten 100 üzerinden minimum 21, maksimum 90 almıştır. Grubun puan ortalaması $\bar{X}=55,87$, standart sapması ise $ss=15,3$ olarak raporlanmıştır.

Yazılımın varsayılan başlangıcı ikili 2-geri modudur (Dual 2-Back). Burada ikili yazılımın sunum türünü –modalitesini- (ses, konum, renk, şekil...vb.) ve 2-geri de geri gidilmesi gereken n değerini ifade etmektedir. Modaliteden bağımsız olarak her oturum için geçerli ekran tasarımı ortada yer alan (3 x 3) lük 9 bölgeye ayrılmış ızgaradır (Şekil 3.5).



Şekil 3.5. Sözel çalışma belleği testi örnek ekran görüntüsü

Göreve ilişkin modalite uygulayıcı tarafından amaca göre ayarlanabilmektedir. Örneğin sadece konum, sadece ses, konum ve ses, konum, ses ve renk gibi özelliklerin değiştirilmesiyle, bireylere bilgisayar ortamında farklı görevler sunulabilmektedir. Bu çalışmada n-geri görevi sadece sözel belleği ölçmek için kullanıldığından işitsel modalite esas alınmış ve iki-geri modunda ölçüm yapılmıştır. Alıştırma uygulamaları için ise önce işitsel (10+4), daha sonra konum

(10+4) ve daha sonra işitsel-konum (15+4) modalitelerde uygulama yapılarak, asıl uygulama için toplamda 30+4 deneme şeklinde sadece işitsel modalite kullanılarak uygulama yapılmıştır.

İşitsel modalitede bireye bir kulaklık verilmektedir. Katılımcıya bir dizi harf sesli olarak 2,5 sn. aralıklarla kulaklıktan söylenmektedir. Katılımcı duyduğu bir uyarıcıyı iki geride de duyduysa klavyeden "L" tuşuna basmaktadır. Örneğin bireye söylenen harfler sırasıyla "T K N M N H M" olması durumunda ikinci M'yi duyduğunda anda "L" tuşuna basarak tepki vermesi gerekmektedir. Çünkü M harfleri arasında iki uyarıcı (N, H) bulunmaktadır. Alıştırmada ekranın sağ üst köşesinde geriye kalan deneme sayısı gösterilirken, sol üst köşede ise uygulamayı durdurma, uygulamadan çıkma ve ekrandaki metinleri gizleme/gösterme için gerekli tuşlar gösterilmektedir. Alt köşelerde n adım geride duyulan ya da görülen uyarıcılar için basılacak tuşlar yer almaktadır.

Uygulama sonunda katılımcılara ilişkin puanların ve uygulamaya ilişkin bilgilerin yer aldığı veriler programın ürettiği bir metin dosyasında tutulmaktadır. Puanlama 100 üzerinden derecelendirilmektedir.

3.4.4 Programlamaya İlişkin Öz Yeterlilik Ölçeği

Çalışmada bireylerin programlamaya ilişkin öz yeterlilik algısını ölçmek üzere Ramalingam ve Wiedenbeck (1998) tarafından geliştirilmiş ve Altun ve Mazman (2012) tarafından Türkçeye uyarlanmış olan "Programlamaya İlişkin Öz Yeterlilik Algısı Ölçeği" (EK 3) kullanılmıştır. Ölçek orijinalde 32 madde ve dört faktörden oluşmaktadır. Uyarlama çalışması için 152 üniversite öğrencisine uygulanarak faktör analizi ile ölçeğin faktör yapısı yeniden incelenmiştir. Uyarlama çalışması sonunda ölçeğin Türkçe formunun 9 madde ve iki faktörden oluştuğu bulunmuştur.

Ölçek 7'li Likert tipindedir. "Karmaşık programlama görevleri" ve "basit programlama görevleri" olmak üzere iki faktörden oluşmaktadır. 1= kendime hiç güvenmiyorum, 2= genellikle güvenmiyorum, 3= biraz güveniyorum, 4= %50/50, 5= oldukça güveniyorum 6=genellikle güveniyorum, 7=tamamen güveniyorum belirtecek şekilde puanlanmıştır. Ölçekten alınacak minimum puan 7, maksimum puan ise 63'tür. Ölçeğin tümünün iç tutarlılık katsayısı .928, alt faktörlerden "karmaşık programlama görevlerinin" iç tutarlılık katsayısı .943, "basit programlama görevlerinin" iç tutarlılık katsayısı ise .907'dir.

3.4.5 Programlama Performansı

Programlama performansı için bireylerin Programlama – I dersinin sonunda aldıkları yılsonu notları değerlendirilmiştir. Her üç üniversitedeki dersin öğretim üyelerinden öğrencilerin yılsonu geçme notu 100'lük sistemde istenmiştir. Yılsonu geçme notu vize ara sınavı ve yılsonu final sınavını içermektedir.

Yılsonu sınav notlarının dışında öğretim üyelerinden yılsonu sınav soruları da ayrıca istenerek sınavların bağlamı incelenmiştir. Her üç üniversitedeki sınavların kağıt kalem tabanlı olarak uygulandığı, bilgisayar ortamındaki uygulamaya dayalı değerlendirmenin olmadığı görülmüştür. Dönem boyunca eğitimi verilen içeriğin temel alındığı programlama dilinin farklı olmasından dolayı final sınavlarının içerdiği sorulardaki programlama dilinin de farklılaştığı görülmüştür. Üniversitelerin birinde sınav soruları genel programlama mantığı soruları çerçevesinde şekillenirken, herhangi bir programlama diline bağlı kalınmaksızın hazırlanmıştır. Bir diğer üniversitede ise sorular C# dili üzerinden örneklenecek hazırlanmıştır. Üçüncü üniversitede ise sınav PHP programlama dili kullanılarak cevaplama gerektiren sorular içermektedir. Üç üniversitedeki final soruları içeriği programlama dili bakımından farklılaşsa da dersin programlama giriş niteliğinde olmasıyla soruların temel düzeyde ve giriş seviyesinde zorlukta olduğu söylenilebilir.

3.5 Ölçme Araçlarının Puanlanması

Programlamaya ilişkin öz-yeterlilik için ölçekten elde edilen toplam puan alınmıştır (maksimum=63, minimum=0). Sözel bellek için ise yazılımın kendi ürettiği 100 üzerinden elde edilen başarı puanı alınmıştır.

Kullanılan testlerden Uzamsal Yönelim Testi, Zihinsel Döndürme Testi ve Görsel-Uzamsal Bellek Testinden hem tepki süresi hem de doğruluk puanları olarak iki ayrı puan elde edilmektedir.

Bilişsel davranışlara ilişkin ölçümlerde tepki süresi ve tepkinin doğruluğu çıktı olarak gözlemlenen değişkenlerdir. Nitekim performans puanı olarak ya sadece doğruluk ya da sadece tepki süresi ya da bu ikisi ayrı ayrı performans puanı olarak alınmaktadır. Bu durumun ise bulguların yorumlanmasında karmaşıklığa yol açtığı ifade edilmektedir (Bruyer ve Brysbaert, 2011). Tepki süresi ve doğruluk arasında karşılıklı dengeli bir ilişki söz konusu olup, bireyler hızlı tepkiler vererek tepki süresini azaltırken, diğer yanda hata oranlarını arttırabilirler (Wagenmakers, Maas,

ve Grasman, 2007). Ayrıca hem doğruluk hem de tepki süresi görev zorluğu ve bireyin becerisi hakkında önemli bilgi sunduğundan, doğruluk ve tepki süresi puanlarının uygun şekilde ağırlıklandırılıp entegre edilerek tek bir bağımlı değişken üretilmesinin daha anlamlı olacağı ifade edilmektedir (Bruyer ve Brysbaert, 2011; Wagenmakers ve diğ., 2007).

Doğru puanlar ve tepki süresini birleştirerek tek bir bağımlı değişken endeksi üretmeye dönük farklı formüller geliştirilmiştir. Townsend ve Ashby (1978) tarafından ortalama doğru cevapların süresi ve doğru cevapların yüzdesi metrikleri ile üretilen tersine “etkililik skoru” ortaya konulmuş Ratcliff (2002) tarafından hata oranı ve doğru ve yanlış cevaplar için tepki süresi dağılımını kullanan yayılım modeli önerilmiş, Wagenmakers ve diğ. (2007) ise Ratcliff modeline dayalı olarak ortalama tepki süresi, tepki süresinin varyansı ve tepkilerin doğruluğunu girdi olarak alan EZ-yayılım modelini ortaya koymuşlardır. Luft, Gomes, Priori ve Takase (2013) ise toplam doğru sayısından yanlış sayısının çıkarılarak, toplam ortalama süreye bölünmesi ile hesapladıkları genel etkililik olarak adlandırdıkları bir indeks kullanmışlardır.

Bu çalışmada sadece doğru cevap ya da sadece yanlış cevaplara verilen tepki sürelerini ele almak yerine tüm tepki süresi üzerinden hesaplanan Luft ve diğ. (2013) tarafından önerilen genel etkililik puanları hesaplanmıştır. Genel etkililik puanlarına ilişkin formül (1) aşağıda gösterilmektedir.

$$\text{Etkililik Skoru} = \frac{\text{Doğru cevap sayısı} - \text{Yanlış Cevap Sayısı}}{\text{Ortalama Tepki Süresi}} \quad (1)$$

3.6 Verilerin Analizi

Verilerin analizinde t testi, tek yönlü varyans analizi ve veri madenciliğine dayalı ağaç tabanlı yöntemlerden “Artırılmış Regresyon Ağacı” yöntemi kullanılmıştır. Ayrıca Artırılmış Regresyon Ağacı yöntemine ilişkin bulguların geçerliliğini test etmek için yine veri madenciliğine dayalı ağaç tabanlı yöntemlerden “Rastgele Ormanlar” yöntemi kullanılmıştır.

Geleneksel yaklaşımlara göre bireylerin bilişsel sistemlerde verdiği tepkiler genellikle bir dizi bileşen etkisinin ortak ürünüdür. Her ne kadar bu bileşenlerin içsel dinamiklerinin karmaşık olduğu bilinse de, örneğin en genel olarak bileşenler arasındaki etkileşimin doğrusal olmasının varsayılması gibi, bileşenler arasındaki etkileşimler çoğu zaman kısıtlanmaktadır (Carello ve Moreno, 2005). Nitekim son zamanlarda yapılan çalışmalarda da düşük ve yüksek performans ve bilişsel beceriler arasındaki ilişkinin her zaman lineer olamayacağı bu nedenle özellikle karmaşık ilişkileri tanımlayabilen doğrusal olmayan yöntemlere gereksinim olduğu ifade edilmektedir (Luft ve diğ., 2013).

Bu çalışmada da bir dizi bilişsel değişkenle bilişsel olmayan performansın modellenmesi söz konusu olduğundan, veriler arasındaki doğrusal ilişkinin anlamlı olmadığı gözlemlenmiştir. Bu doğrultuda programlamaya ilişkin performansı açıklayan değişkenleri ve bunların önemini modellenmesinde diğer ağaç tabanlı regresyon yöntemlerinin sınırlılıklarının birçoğunu ortadan kaldırmaya dönük olarak geliştirilmiş güncel doğrusal olmayan tekniklerden artırılmış regresyon ağacı kullanılmıştır. Artırılmış Regresyon ağaçlarından elde edilen bulguların geçerliliğini test etmek için aynı model ayrıca rastgele orman tekniği ile analiz edilmiştir. Bu iki yöntemin işleyişi ve temel özellikleri aşağıdaki kısımda kısaca özetlenmiştir.

3.6.1 Artırılmış Regresyon Ağaçları (Boosted Regression Trees)

Artırılmış regresyon ağacı; regresyon ağaçları (yinelemeli ikili bölümlerle yordayıcılarına ilişkin tepki veren modeller) ve artırım (performansı yordama oranını artırmak üzere basit modelleri birleştiren uyarlanabilir bir yöntem) tekniklerinin güçlü yönlerini birleştiren bir yöntemdir (Elith, Leathwick, ve Hastie, 2008). Artırılmış ağaç algoritmaları, regresyon ağacı yapısına artırma yönteminin uygulanması ile gerçekleşir. Temel amaç her bir ardışık ağacın kendinden önceki ağaçtan kalanları tahmin etmek üzere kurulduğu sıralı ve ileri dönük basit ağaçlar üretmektir (Elith ve diğ., 2008; StatSoft, 2013). Bu tür artırılmış ağırlıklandırılmış yayılım, yordayıcı değişkenler ve bağımlı değişkenin arasındaki ilişkinin doğrusal olmaması gibi karmaşık durumlarda dahi, yordanan değişkenle gözlemlenen değişkenlere ilişkin mükemmel uyum üretir (StatSoft, 2013). Artırılmış regresyon ağaçları, regresyon ağaçları ve artırım olmak üzere iki temele dayanmaktadır. Bir

sonraki kısımda regresyon ağacı ve artırım tekniğine ilişkin temel bilgiler sunulacaktır.

Regresyon ağaçları:

Regresyon ağaçları bir ya da birden çok açıklayıcı değişkenle tek bir bağımlı değişkendeki (kategorik ya da sürekli) varyasyonu açıklarlar (De'ath ve Fabricius, 2000). Ağaç yapılı modeller yordayıcıya ilişkin en homojen bölgeleri tanımlayabilmek için bir dizi kurallar kullanarak yordayıcı alanı dikdörtgenlere böler ve sonrasında hataların normal dağıldığını varsayarak, her bir bölgeye sabit yerleştirmekle birlikte, en uygun sınıfı sabit olarak uyumunu sağlayarak ağaçları sınıflarlar (Elith ve diğ., 2008). Bir ağaç modeli geliştirmek için iki ölçüte ihtiyaç vardır; bir düğümdeki en iyi bölünmeyi belirleme ve budama için en iyi ağaç boyutunu seçme (Mendez, 2008).

Young (2007) bir regresyon ağacının yapılandırılmasındaki temel basamakları sıralayarak aşağıdaki gibi özetlemiştir:

- Verilerin bölünmesi
- Her bir bölünmeden sonra bir modelin veriye uydurulması
- Modele ilişkin artık veriler sifıra yaklaştığında durulması
- Eğer ağaçlar aşırı uyum vermişse, ağaçların budanması

Artırım Tekniği (Boosting):

Artırım modelin doğruluğunu tek bir ve yüksek doğrulukta bir tahmin kuralı bulmak yerine ortalama ve daha basit bir temel kurma fikrine dayanarak geliştirmeye dönük bir yöntemdir (Elith ve diğ., 2008). Artırım tekniği tek bir regresyon ağacının zayıf yönlerini azaltmada etkilidir. Artırım tekniği ile modelin doğruluğu tek bir doğru kural yerine birçok tahmin kuralı arasında arama yaparak, modelin doğruluğunu artırır (Schapire, 2003).

Artırılmış Regresyon Ağacı Analiz Süreci

Standart regresyonların aksine tek bir yordayıcı model üretmek yerine, artırılmış regresyon ağaçlarının birden çok basit modele uyumu sağlanır ve daha sonra bu modeller tahmin için birleştirilir, böylece yordayıcı performans artar (Buston ve Elith, 2011).

Artırılmış regresyon ağacı sürecinde üç temel unsur önemlidir (Elith ve diğ., 2008):

- Süreç tamamen olasılıksaldır, rastgele ya da olasılıksal bileşenler içerir. Olasılıksallık her yeni ağaca uyum göstermesi için verinin sadece rastgele bir alt grubunu kullanarak yordama gücünü artırırken, nihai modelin varyansını düşürür.
- Sıralı model uygunluğu sürecinde ağaçlar daha önce uyum gösteren yapılarının üzerine kurulur ve tahmin için en kesin gözlemler üzerine odaklanılır. Eğer mükemmel uyum sadece bir ağaçta ise, bir artırılmış model, büyük olasılıkla kendi daraltılmış versiyonlarının benzerlerinin toplamı ile iyi uyum göstermiştir.
- Analiz sürecinde iki önemli parametre için değerler ortaya konulmalıdır; 1) her bir ağacın büyüyen modele olan katkısını temsil eden öğrenme oranı (küçülme parametresi olarak da bilinir) ve 2) etkileşimlerin uyum gösterip göstermediğini kontrol eden ağaç karmaşıklığıdır.

Artırılmış regresyon ağacı analizlerinde farklı algoritmalarla yararlanılmakla birlikte, bu çalışmada kullanılan STATISTICA 10 paket programının kullandığı Olasılıksal Gradyan Artırım (Stochastic Gradient Boosting) algoritması da bunlardan birisidir.

Artırılmış regresyon ağacı analizlerinde Olasılıksal Gradyan Artırım algoritmasını kontrol etmede beş temel parametre kullanılır (Carty, 2011);

- 1) Öğrenme oranı / küçülme parametresi (Learning rate-lr): Ardışık basit regresyon ağaçlarının tahmin eşitliğine eklendiği belirlenmiş özgün ağırlıktır; öğrenme katsayısında belirlenmiş küçülme oranı nihai artırılmış regresyon ağacı modelindeki her bir ağaç için uygulanır (Elith ve diğ., 2008).
- 2) Toplanabilir terimlerin sayısı (number of additive terms-nat): Ardışık artırım basamaklarında hesaplanacak olan basit regresyon ağaçlarının sayısını belirtir. Elith ve diğ. (2008) öğrenme katsayısının küçük "nat"ın ise büyük olmasının gözlem sayısı ve işlem süresine bağlı olarak daha tercih edilir olduğunu ifade etmişlerdir.
- 3) Maksimum düğüm sayısı (maksimum number of node-mnm): Artırım sırasında yer alan her bir bireysel ağaç için sunulan maksimum düğüm sayısıdır. Her bir üst düğüm bölündüğünde ağaçtaki toplam düğüm sayısı

incelenir ve eğer bu sayı maksimum düğüm sayısını aşarsa bölünmenin durdurulması ile bir durdurma parametresi olarak kullanılır. Elith ve diğ. (2008) BRT modelinin nat, lr ve mnn değerlerinin tümünün optimizasyonu ile düzenleneceğini ifade etmişlerdir.

- 4) Alt örneklem oranı (subsample proportion): Ardışık artırım basamakları için rastgele bir öğrenme örnekleme seçimi için kullanılır. Bu oran kullanıcı tarafından en optimal tahmin oranını bulmada ayarlanabilir.
- 5) Rastgele test verisi oranı (random test data proportion): Standart olarak 0.2 (%20)'dir. Bunun anlamı gözlemlerin %80'i rastgele olarak eğitim seti için, %20'si ise test seti için seçilmiştir. Bu oran kullanıcıların veri setinin büyüklüğüne göre değiştirilebilir.

Artırılmış regresyon ağaçları önemli yordayıcı değişkenleri tanımlar ve işlem ya da etkileşimlerin biçimine ilişkin herhangi bir varsayımda bulunmaksızın karmaşık işlem ve etkileşimleri modeller (Elith ve diğ., 2008). Yordayıcı değişkenlerin görelî önemi Friedman (2001) tarafından formüle edilmiş ve yordayıcı değişkenlerin görelî önem etkisini tahmin ederek analiz sonucundaki modelin yorumlanmasını sağlar. Değişkenlerin görelî önemlerinin hesaplanması için, ağaçların her birisinin üretilmesi esnasında her bir bölünme için tahmin edici istatistikler (regresyon kareleri toplamı) tüm yordayıcı değişkenler için hesaplanır. İlgili düğümdeki en iyi bölünmeyi gerçekleştiren en iyi yordayıcı değişken bölünme için seçilir. Seçilen değişken sabit için hata kareler riskinde gelişimi tüm bölgede en iyi tahmin eden değişkendir (Hastie, Tibshirani, ve Friedman, 2009). Tüm bölünmelerdeki ve artırımdaki tüm ağaçlardaki tahminleyici istatistiklerin ortalaması her değişken için hesaplanır. Nihai tahminleyici önem değerleri ise bu ortalamaların normalleştirilmesiyle hesaplanır, böylece en yükseğe bir değerini verecek şekilde ve diğer tüm tahminleyici önemler tahminleyici istatistiklerin değerlerinin ortalamasına ilişkin görelî büyüklüğe göre açıklanır (Carty, 2011; Hastie ve diğ., 2009).

3.6.2 Rastgele Ormanlar (Random Forests)

Rastgele ormanlar, her bir ağacın bağımsız bir şekilde örneklenen rastgele vektör değerine bağlı olduğu ve ormandaki tüm ağaçlar için bu dağılımın aynı olduğu yordayıcı ağaç kombinasyonlarıdır (Breiman, 2001). Rastgele ormanların temel

hedefi torbalamaya (bagging) ilişkin varyansın indirgemesini sağlamak için bireysel ağaçlar arasındaki korelasyonu yerine koyma ve rastgele değişken seçme yöntemi ile azaltmaktır (Hastie ve diğ., 2009; Jun, 2013). Ormandaki ağaçların sayısı arttıkça ormanın genelleme hatası belirli bir limite yaklaşır ve bu da her bir bireysel ağacın gücüne ve aralarındaki ilişkiye bağlıdır (Breiman, 2001).

Breiman ve Cutler (2004) rastgele ormanların temel özelliklerini aşağıdaki gibi sıralamışlardır:

- Doğruluk bakımından diğer algoritmalarından daha üstündürler.
- Oldukça geniş veri setlerinde etkili bir şekilde çalışır.
- Binlerce girdi değişkeni silme işlemi olmaksızın analize alınabilir.
- Sınıflamada ya da tahmin etmede hangi değişkenlerin önemli olduğunu ortaya koyar.
- Ormanı yapılandırma sürecinde genelleme hatasına ilişkin içsel yansız bir tahmin üretir.
- Kayıp verileri tahmin etmede oldukça etkilidir ve veri setindeki kayıp veri çok fazla olsa dahi doğruluk oranını etkilemez.
- Üretilen ormanlar ileride başka veri setlerinde kullanılmak üzere kaydedilebilir.
- Dengelenmemiş veri setlerindeki sınıf örnekleme için hatayı dengeleme yöntemlerine sahiptir.

Rastgele ormanlar bootstrap ve rastgele değişken seçimine dayalı bir yöntemdir. Rastgele ormanlarında geçerlilik için *out of bag* tekniği kullanılırken, değişken seçimi için ise değişken önemi parametresi kullanılmaktadır. Rastgele orman yöntemine ilişkin bu kavramlar aşağıda kısaca özetlenmiştir.

Bootstrap (Yerine koyma):

Bootstrap istatistiksel olarak doğruluğu değerlendirmede kullanılan bir tekniktir. Temel mantık eğitim verisinden rastgele olarak yerine koyma yöntemi ile veri setleri çekilmesi olup, her bir örneklem büyüklüğü orijinal eğitim setindeki ile aynıdır. Bu işlem x bootstrap veri setini oluşturmak üzere x kez tekrarlanır. Daha sonra model her bir bootstrap veri setine yeniden uydurulur ve her bir x yinelemesi için uyum davranışı incelenir (Hastie ve diğ., 2009).

Rastgele deęişken seçimi:

Rastgele deęişken seçimi ile ağaçlar ilişkisiz hale getirilir ve böylece tahmin hatası en aza indirgenir (Clarke, Fokoue, ve Zhang, 2009). Klasik bir ağaç yapılandırılmada yinelemeli bölünme esnasında tüm deęişkenler incelenerek en iyi bölünme noktası bulunmaya çalışılırken, rastgele ormanların rastgele deęişken seçimi özellięi ile tüm deęişkenlerin sayısının karekökü kadar deęişken seçilerek bunlar arasından en iyi bölünme noktası bulunur. Bu da oldukça ilişkili olan deęişkenlere eş roller vererek, ormandaki farklı ağaçlar arasındaki benzerlięi azaltır.

Rastgele orman teknięinde analiz sürecinde “out of bag” ve deęişken önemi olmak üzere iki önemli parametre söz konusudur.

Out of bag (oob) hata tahmini:

Rastgele karar ormanlarında test setindeki hata analiz esnasında içsel olarak tahmin edildięinden, yansız hata tahmini için herhangi bir çapraz geçerlilięe ya da ayrı bir teste gerek yoktur (Breiman ve Cutler, 2004). Rastgele ormanda bunun için OOB hata tahmini kullanılır. Dięer doğrusal olmayan tahminleyicilerin aksine rastgele ormanlar tek bir dizi şeklinde uyum sağlarken ederken aynı zamanda çapraz geçerlilik sağlanır. OOB hatası sabit hale geldiğinde test süreci sonlandırılır (Hastie ve dię., 2009).

OOB hata tahmininin gerçekteşme süreci şu şekilde tanımlanmaktadır (Breiman ve Cutler, 2004; Liaw ve Wiener, 2002):

Her bir ağaç orijinal veri setinden farklı bir yerine koyma örnekleme oluşturulur. Gözlemlerin yaklaşık 1/3’ü k.’nci ağacın oluşturulmasında kullanılmaz ve yerine koyma örnekleminin dışında bırakılır. Dışarıda bırakılan bu veri ile (OOB) her bir ağaç için hata tahmini test edilir. Her bir karar ağacı için OOB hata tahmini yapıldıktan sonra, bu hata oranlarının bir ortalaması alınarak model için tüm hata oranı tahmini hesaplanır.

Deęişken Önemi:

Deęişken önemi rastgele ormanlara ilişkin modeldeki deęişkenlerin deęerlendirilmesi ve seçiminde kullanılan bir parametredir. Rastgele orman algoritması bir deęişkenin önemini, o deęişkene ilişkin veriler deęişirken

diğerlerinin sabit kaldığı durumda hata tahminin ne kadar arttığına bakarak hesaplar. Bu hesaplama şu şekilde gerçekleşir (Breiman ve Cutler, 2004):

Ormanda büyüyen her bir ağaç için oob gözlemleri yerine koyulur ve doğru sınıflama için kullanılan oylar sayılır. Daha sonra oob gözlemleri içindeki m değişkenine ait değer rastgele değiştirilir ve bu gözlemler ağaca yerleştirilir. Oob verisindeki değeri değiştirilen m değişkenindeki doğru sınıflama için kullanılan oylar değeri değiştirilmeyen oob verisindeki doğru sınıflama oylarından çıkarılır. Ormandaki tüm ağaçlara ait bu değerlerin ortalaması m değişkeni için ham önem puanını üretir.

Tüm bu hesaplamalar rastgele ormanın yapılanması esnasında ağaç ağaç yürütülür (Liaw ve Wiener, 2002).

Arttırılmış Regresyon Ağaçları ve Rastgele Ormanlar Yöntemlerinin Karşılaştırılması

Arttırılmış regresyon ağaçları ve rastgele orman yöntemlerinin her ikisi de ağaç tabanlı regresyon yöntemlerinden olmakla birlikte kullandıkları algoritmalar ve işe koşulan teknikler bakımından farklılıklar göstermektedir. Jun (2013), her iki yöntem arasındaki farklılıkları kategorilere ayırmış, buna ilişkin karşılaştırma Çizelge 3.2'de sunulmuştur;

Çizelge 3.2. Rastgele Ormanlar ve Artırılmış Regresyon Ağacı teknikleri karşılaştırma tablosu

Rastgele Ormanlar	Artırılmış Regresyon Ağacı
<ul style="list-style-type: none">• Yerine koyma (Bootstrap) ve rastgele değişken seçimi tekniklerini kullanır.	<ul style="list-style-type: none">• Artırım (Boosting) ve regresyon ağaçları tekniğini kullanır.
<ul style="list-style-type: none">• Ormanda büyüyen ağaçlar için kullanılan değişken sayısı sabitlenirken ağaç büyüklüğü aşırı uyuma karşı kontrol edilir.	<ul style="list-style-type: none">• Rastgele ormandaki aşırı uyum kontrolünden ayrıca her bir iterasyondaki ağaç büyüklüğü ayrıca kontrol edilir.
<ul style="list-style-type: none">• Rastgele ağaçlarda eğer her bir ağacın büyümesi esnasında kullanılan değişken sayısı çok fazla ise bu değişkenler arasında karmaşık ilişkiye neden olacağından aşırı uyuma sebep olabilir.	<ul style="list-style-type: none">• Artırılmış regresyon ağaçlarında ise iterasyon sayısının çok fazla olması aşırı uyuma sebep olabilir. Artırılmış regresyon ağaçlarında aşırı uyumu önlemek sadece iterasyon sayısını ayarlamak ile mümkün olduğundan daha tercih edilir bir yöntemdir.
<ul style="list-style-type: none">• Tahmin doğruluğu açısından rastgele ormanlar, rastgele değişken seçimine dayandığından, eğer her bir ağacın büyümesi için kullanılacak değişken sayısını belirlerken aşırı uyumu önlemek için küçük bir değer girilirse, ilişkili değişkenler ihmal edilmiş olacaktır.	<ul style="list-style-type: none">• Artırılmış regresyon ağaçlarında ise tahmine ilişkin doğruluğu azaltabilecek olası faktör, seçilen ağaç boyutunun değişkenler arasındaki etkileşimi doğru bir şekilde yansıtmayacak şekilde belirlenmesidir. Artırılmış regresyon ağaçları algoritmasında herhangi bir rastgele bileşen yoktur. Artırım tekniği ile popülasyondaki işlev kaybını minimize eden artırılmış bir modeli fit eder.

4. BULGULAR

Bu bölümde araştırma bulgularına yer verilecektir.

İlk kısımda çalışmada katılımcılardan elde edilen ölçümler olan uzamsal yönelim becerisi, zihinsel döndürme becerisi, uzamsal bellek, sözel bellek, programlamaya ilişkin öz yeterlilik algısı ve programlama performansı olarak alınan programlama geçme notlarına ilişkin ortalama, standart sapma, standart hata, basıklık ve çarpıklık değerlerine ilişkin betimsel istatistikler verilecektir. Ayrıca bu değişkenler arasında doğrusal ilişkinin anlamlılığını incelemek için korelasyonları incelenerek korelasyon matrisi sunulacaktır.

İkinci kısımda ise programlama performansının yordanmasına ilişkin kurulan modelde, katkıları incelenen demografik değişkenler olan cinsiyet, ön deneyim ve üniversite faktörlerinin programlama performansı ile ilişkisini inceleyebilmek için fark analizleri yapılmış ve bunların sonuçları raporlanmıştır.

Üçüncü kısımda ise programlama performansının yordanmasına ilişkin kurulan modelin ağaç tabanlı yöntemlerle analizine ilişkin bulgular verilmiştir. Bu kısımda ilk olarak modelin artırılmış regresyon ağacı tekniği ile analizi raporlanmış, ardından bu yöntemle elde edilen bulguların geçerliliğinin sağlanması için rastgele orman tekniği ile analiz sonuçlarına yer verilerek, bulgular karşılaştırılmalı olarak sunulmuştur.

4.1 Bireylerin Programlama Performansı, Öz Yeterlilik Algıları, Uzamsal Becerileri, Çalışma Belleği Düzeyleri ve Aralarındaki İlişkiye İlişkin Bulgular

Katılımcılara uygulanan ölçme araçları olan uzamsal yönelim testi, zihinsel döndürme testi, görsel-uzamsal bellek testi, sözel bellek testi, öz yeterlilik algısı ölçeğinden elde edilen sonuçlar ve programlama performansı olarak alınan programlama geçme notlarına ilişkin betimsel istatistikler Çizelge 4.1'de verilmektedir.

Çizelge 4.1. Programlama performansı, uzamsal beceri, çalışma belleği ve öz yeterlilik algısına ilişkin betimsel istatistikler

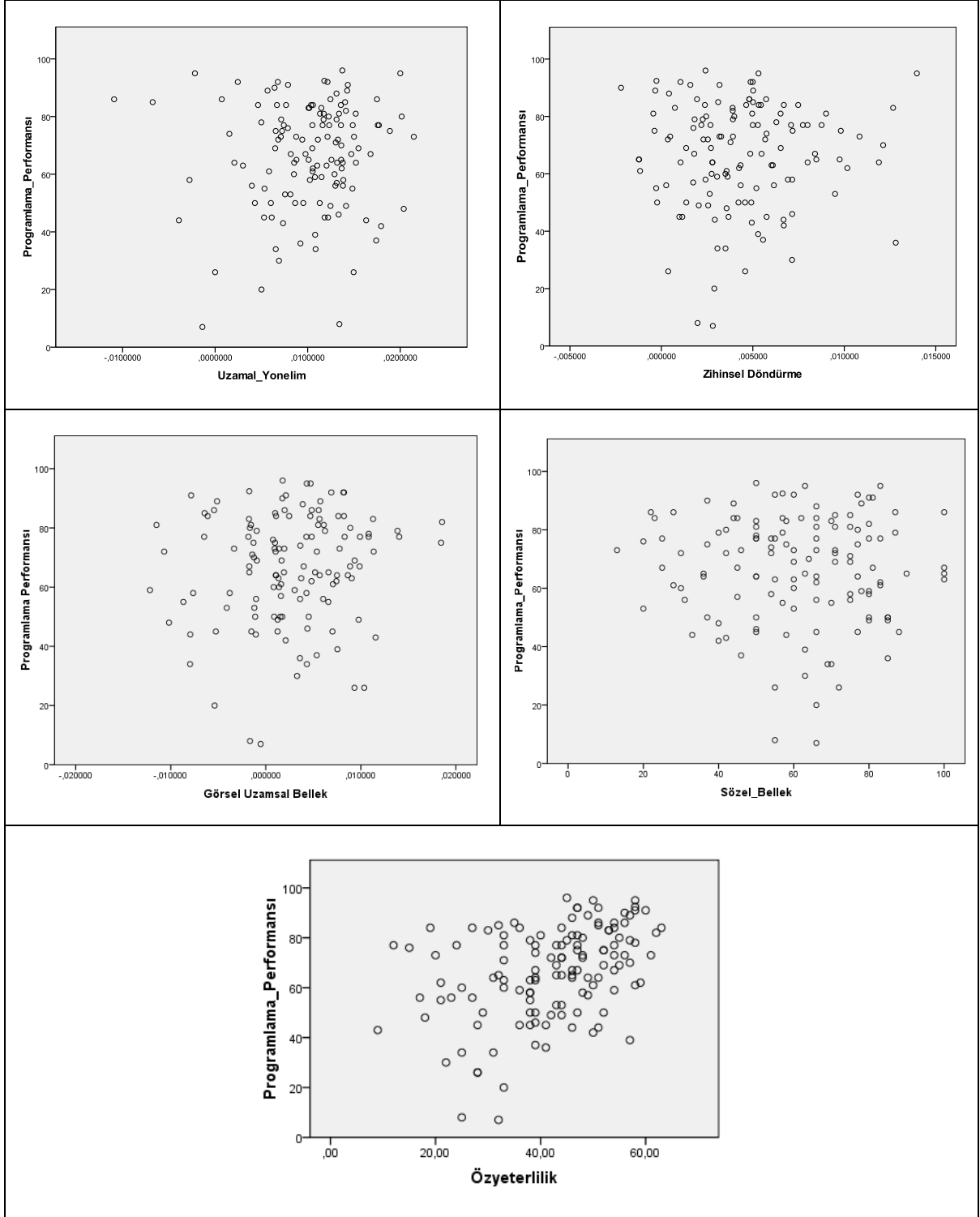
	\bar{X}	SS	SH	Basıklık	Çarpıklık
Programlama Performansı	66,36	18,4	1,62	,506	-,790
Öz-yeterlilik	42,12	11,8	1,04	-,270	-,562
Uzamsal Yönelim	0,1016	0,0053	0,00047	1,9	-,955
Zihinsel Döndürme	0,0043	0,0032	0,00027	,509	,633
Uzamsal Bellek	0,0026	0,0058	0,00051	,263	-,142
Sözel Bellek	60,64	19,1	1,68	-,431	-,246

Değişkenler arasındaki ilişkiyi gösteren korelasyonlar hesaplanmış ve Çizelge 4.2'de gösterilmiştir.

Çizelge 4.2. Değişkenler arasındaki korelasyon matrisi

	Programlama Performansı	Öz-yeterlilik	Uzamsal Yönelim	Zihinsel Döndürme	Uzamsal Bellek	Sözel Bellek
Programlama Performansı	1					
Öz-yeterlilik	,406**	1				
Uzamsal Yönelim	,063	,000	1			
Zihinsel Dönd.	,027	,062	,130	1		
Uzamsal Bellek	,117	,091	,168	,061	1	
Sözel Bellek	-,037	,059	,089	,148	,225*	1

Çizelge 4.2'den görüldüğü gibi programlama performansı ve öz yeterlilik algısı ile uzamsal bellek ve sözel bellek arasında oldukça düşük ($r < 0.5$, $p < 0.05$) korelasyon bulunurken, diğer değişkenler arasındaki korelasyonlar anlamlı olarak bulunmamıştır. Programlama performansı ile her bir bağımsız değişken arasındaki ilişkiyi gösteren saçılım diyagramları Şekil 4.1'de verilmiştir



Şekil 4.1 Bağımlı değişken ve sürekli bağımsız değişkenlere ilişkin saçılım grafikleri

Hem korelasyon matrisi hem de Şekil 4.1'deki saçılım diyagramları programlama performansının programlama öz yeterliliği dışındaki sürekli bağımsız değişkenlerle arasındaki ilişkinin doğrusal yapıda olmadığını göstermektedir.

4.2 Programlama Performansının Bireylerin Ön Deneyimleri, Cinsiyet ve Üniversitelerine Göre İncelenmesine İlişkin Bulgular

Programlama performansının yordanmasına ilişkin kurulan modeldeki kategorik değişkenler olan üniversite, cinsiyet ve ön deneyim faktörleri ile programlama performansının ilişkisini ortaya koyabilmek için ise fark analizleri yapılmıştır. Cinsiyet ve ön deneyim değişkenleri için bağımsız örneklem t testi yapılırken, üniversite değişkeni için tek yönlü varyans analizi yapılmıştır.

Çizelge 4.3. Cinsiyet ve ön deneyime göre programlama performansına ilişkin t testi sonuçları

		N	\bar{X}	SS	sd	t	p
Cinsiyet	Kadın	69	67,14	17,379	127	,516	,607
	Erkek	60	65,46	19,764			
Ön deneyim	Var	71	70,99	16,414	127	-3,268	,001
	Yok	58	60,69	19,389			

* p < .05

Çizelge 4.3'den görüldüğü gibi kadın ve erkeklerin programlama performansı arasında anlamlı fark bulunmazken, ön deneyime sahip olup olmama durumuna göre anlamlı fark bulunmuştur. Buna göre ön deneyimi olan grubun programlama performansı ön deneyimi olmayan gruba göre anlamlı derecede daha yüksek olarak ortaya çıkmıştır.

Programlama performanslarının üniversitelere göre farklılık gösterip göstermediğine bakmak üzere tek yönlü varyans analizi yapılmış, bu analize ilişkin bulgular Çizelge 4.4'te sunulmuştur.

Çizelge 4.4 Üniversitelere göre programlama performansına ilişkin tek yönlü varyans analizi sonuçları

Üniversite	N	\bar{X}	ss	Varyans kaynağı	Kareler Ort.	sd	Ort. Kare	F	p	Scheffe Testi
Hacettepe	63	76,2	10,5	Gruplar Arası	14645,23	2	7322,6	31,78	.000	Hacettepe>Gazi p=.000 Hacettepe>Ankara p=.000
Gazi	29	50	22	Gruplar İçi	29031,04	126	230,4			Ankara >Gazi p=.003
Ankara	37	66,4	15,3	Toplam	43676,27	128				

Çizelge 4.4'ten görüldüğü gibi öğrencilerin programlama performansları üniversitelere göre anlamlı olarak farklılık göstermektedir ($p<.05$). Hangi üniversiteler arasında olduğunu ortaya koymak için post hoc analizlerden Scheffe testi yapılmıştır. Scheffe testi sonuçlarına göre, programlama performansı puanları açısından Hacettepe Üniversitesi'nin hem Ankara Üniversitesi'nden hem de Gazi Üniversitesi'nden anlamlı derecede yüksek olduğu, Ankara Üniversitesi'nin de Gazi Üniversitesinden anlamlı derecede yüksek olduğu bulunmuştur.

4.3 Programlama Performansının Yordanmasına İlişkin Kurulan Modelin Ağaç Tabanlı Yöntemlerle Test Edilmesine İlişkin Bulgular

Bu kısımda programlama performansının yordanmasına ilişkin kurulan modelin ağaç tabanlı yöntemlerle test edilmesine ilişkin analiz sonuçları yer almaktadır.

İlk olarak artırılmış regresyon ağacı tekniği analizine ilişkin bulgulara, ardından rastgele orman tekniği ile ilgili analiz bulgularına yer verilecektir.

4.3.1 Artırılmış Regresyon Ağacı Sonuçlarına İlişkin Bulgular

Uzamsal yönelim becerisi, zihinsel döndürme, görsel-uzamsal bellek, sözel bellek olmak üzere dört bilişsel beceri ile programlamaya ilişkin öz yeterlilik algısı, öğrenim görülen üniversite, ön deneyim ve cinsiyet faktörlerinin programlama performansını yordama gücünü belirleyebilmek için artırılmış regresyon analizi tekniği uygulanmıştır.

Analiz için değişken belirleme kısmında açıklanacak olan bağımlı değişken için programlama performansı seçilirken, yordayıcı bağımsız değişkenler ise uzamsal

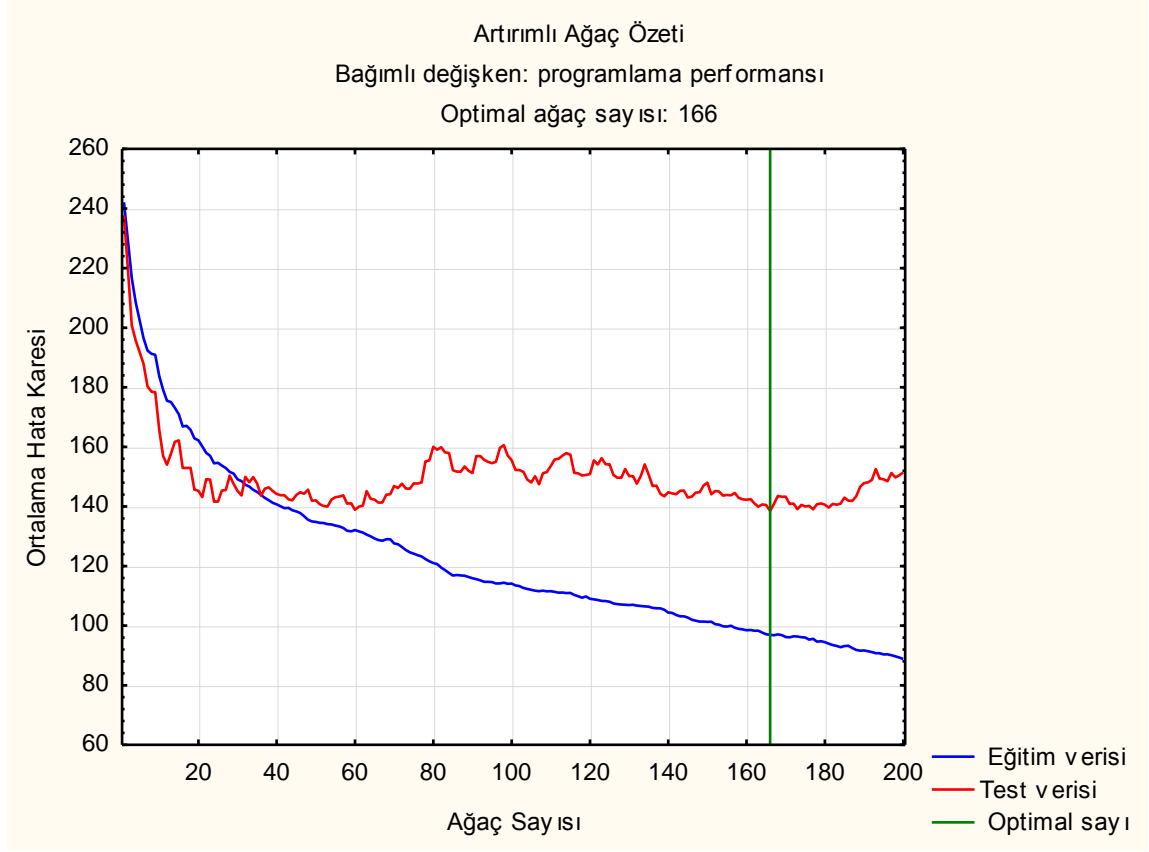
yönelim, zihinsel döndürme, görsel-uzamsal bellek, sözel bellek ve öz yeterlilik algısı sürekli değişkenler; üniversite, ön deneyim ve cinsiyet ise kategorik değişkenler olarak seçilmiştir.

Her bir ağaç için maksimum düğüm sayısı varsayılan şekilde 3 olarak ayarlanmış, toplamsal terimlerin sayısı (ardışık ağaç sayısı) varsayılan şekilde 200 olarak bırakılmış ve öğrenme oranı (küçülme parametresi) da yine varsayılan biçimde .1000 olarak seçilmiştir.

Rastgele test veri oranı (random test data proportion) 0.1 (%10) olarak belirlenmiştir. Bu oran toplam gözlem sayısının %10'nun geçerlilik için test verisi olarak, %90'ının ise modelleme için eğitim verisi olarak kullanılacağını ifade etmektedir. Ardışık artırım aşamalarındaki rastgele öğrenme örnekleminin seçimini belirten alt örneklem oranının en uygun değerini sağlamak için ise “.20 ile .60” arasındaki değerler denenerek analize alınmıştır. Analizler sonucu en uygun alt örneklem değeri .45 olarak ayarlanmıştır.

Durdurma parametreleri için ise varsayılan değerler kullanılmıştır. Bunlar; “maksimum düğüm sayısı” 3, “alt düğümde olabilecek minimum gözlem sayısı” 1, tüm terminal düğümlerde belirtilen minimum sayıda gözlemden daha fazla kalmayacak şekilde bölünmeyi kontrol eden “durmak için minimum sayı” 1 ve her bir ağaçtaki olabilecek maksimum seviye (kök düğümden en üst düğüme kadar olan ağacın derinliği) 5 olarak belirlenmiştir.

Analiz sonucunda optimal ağaç sayısına 166 ağaçta ulaşıldığı bulunmuştur. Artırılmış regresyon ağaçları analizindeki temel etki modeline ilişkin optimizasyon grafiği Şekil 4.2'de verilmiştir.



Şekil 4.2 Artırılmış Regresyon Ağacı analizi optimizasyon grafiği

Şekil 4.2'deki grafik aynı zamanda ağaçların sürece aşamalı bir şekilde eklenmesiyle modele ilişkin sapmanın minimuma indirgenmesini temsil eder.

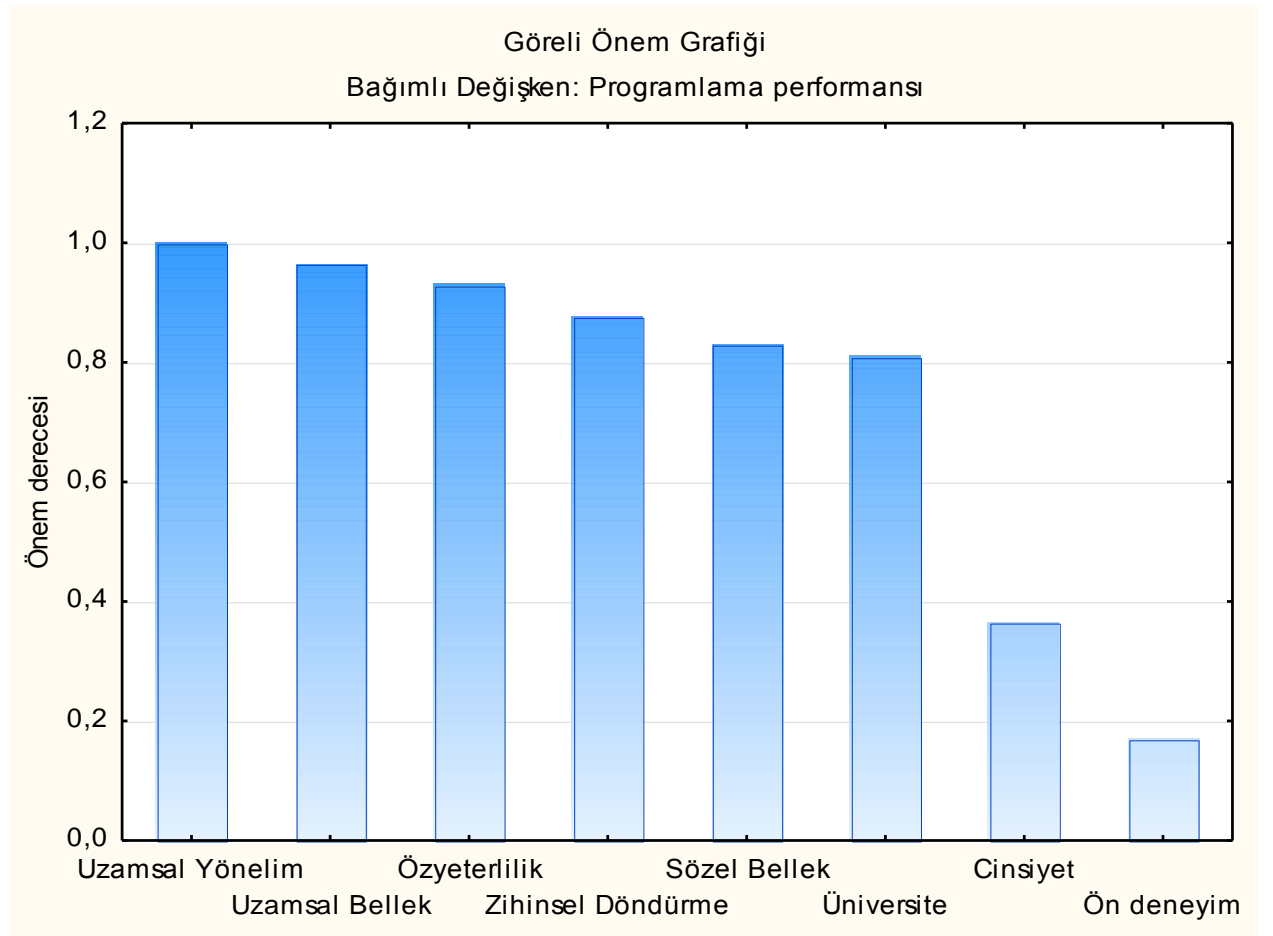
Analiz sonuçlarına göre 8 değişken (zihinsel döndürme, uzamsal yönelim, görsel-uzamsal bellek, sözel bellek, programlaya ilişkin öz yeterlilik algısı, ön deneyim, üniversite ve cinsiyet) birlikte programlama performansının %65,'ini yordamaktadır.

Her bir değişkenin artırılmış modeldeki yordama gücünün ayrı ayrı belirlenmesinde görece önem ağırlıkları ve değişken sırası parametreleri incelenmiştir. Bağımlı değişkenin yordanmasına katkı sağlayan değişkenlerin önem ağırlığı ve değişken sırası değerleri Çizelge 4.5' te sunulmuş, bu sıralamaya ilişkin görece önem grafiği ise Şekil 4.3'de verilmiştir.

Çizelge 4.5. Artırılmış Regresyon Ağacı analizi açıklayıcı değişkenlere ilişkin değişken sıraları ve önem ağırlıkları

	Değişken Sırası	Önem ağırlığı
Uzamsal Yönelim	100	1
Uzamsal Bellek	97	0,965*
Özyeterlilik	93	0,929*
Zihinsel Döndürme	88	0,876*
Sözel Bellek	83	0,830*
Üniversite	81	0,809*
Cinsiyet	37	0,365
Ön deneyim	17	0,169

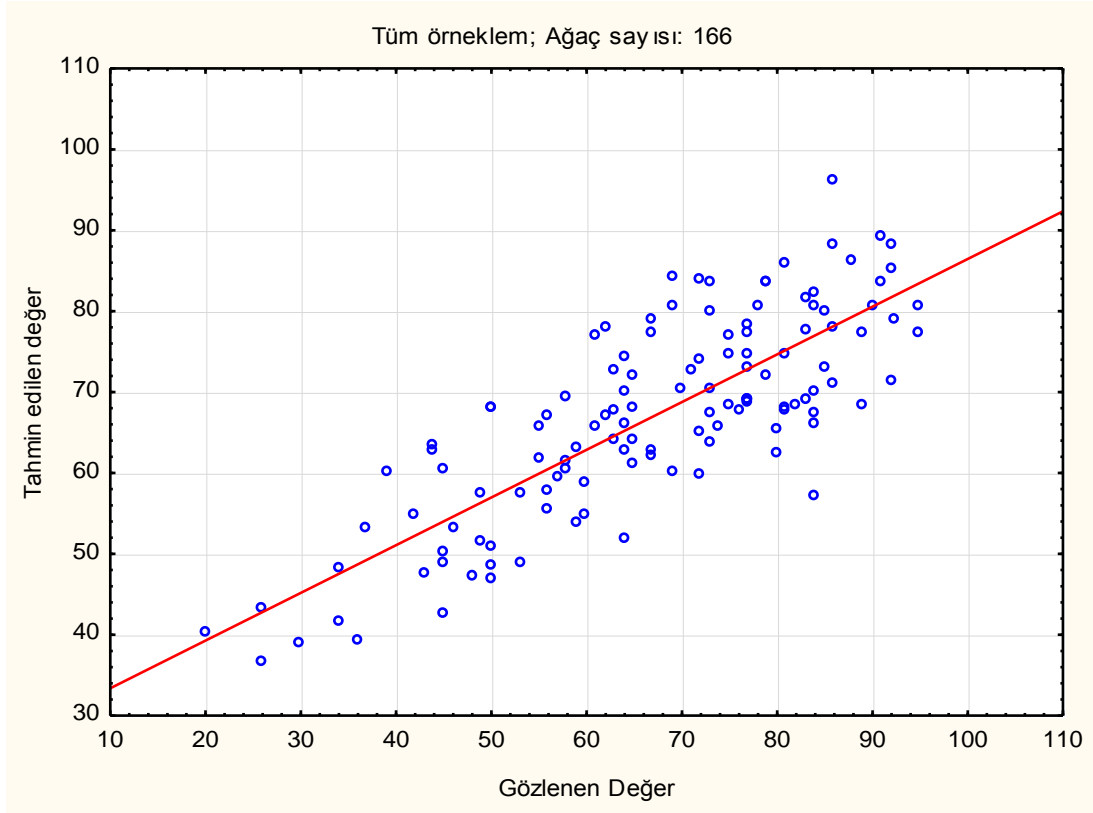
* = Önem derecesi > 0.4 olan değişkenler



Şekil 4.3. Artırılmış Regresyon Ağacı analizi açıklayıcı değişkenlere ilişkin görelî önem katsayıları grafiđi

Çizelge 4.5'ten görüldüğü gibi programlama performansının yordanmasına ilişkin analize alınan değişkenler arasından görece önem ağırlığı en yüksek değişken uzamsal yönelim becerisi olurken (önem derecesi= 1), uzamsal bellek (önem derecesi= 0.965) ikinci sırada, öz yeterlilik algısı (önem derecesi= 0.929) üçüncü sırada, zihinsel döndürme becerisi (önem derecesi= 0,876) dördüncü sırada yer almakta olup, bunu sözel bellek (önem derecesi= 0,830) izlemiş ve altıncı sırada ise üniversite (önem derecesi= 0,809) değişkeni yer almıştır. Cinsiyet (önem derecesi= 0,365) ve ön deneyim (önem derecesi= 0,165) değişkenleri ise programlama performansını yordamada önem sırası bakımından son iki sırada yer almıştır. Şekil 4.3' ten de görüldüğü üzere programlama performansının yordanmasında dört bilişsel beceri, öz yeterlilik algısı ve üniversite hemen hemen eşit öneme sahip olarak ortaya çıkarken, cinsiyet ve ön deneyim faktörlerinin önemi oldukça düşmektedir.

Bağımlı değişken programlama performansına ilişkin gözlemlenen ve beklenen değerler arasındaki ilişkiyi gösteren saçılım grafiği Şekil 4.4'te verilmiştir.



Şekil 4.4 Artırılmış Regresyon Ağacı analizi saçılım grafiği

4.3.2 Rastgele Orman (Random Forest) Analizine İlişkin Bulgular

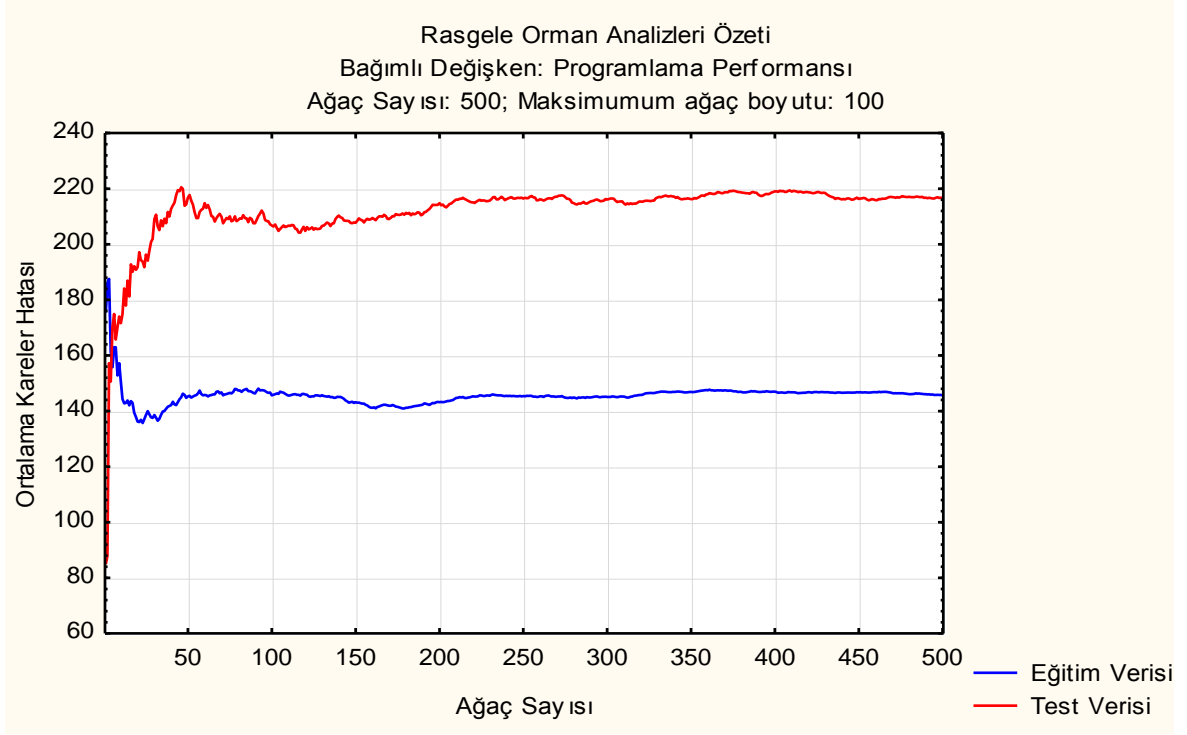
Uzamsal yönelim becerisi, zihinsel döndürme, uzamsal konumsal bellek, sözel bellek olmak üzere dört bilişsel beceri ile programlamaya ilişkin öz yeterlilik algısı, öğrenim görülen üniversite, ön deneyim ve cinsiyet faktörlerinin programlama performansını yordama gücüne ilişkin kurulan modeli doğrulamak için rastgele ormanlar tekniğinin regresyon analizi özelliği kullanılmıştır.

Değişkenler belirlenirken bağımlı değişken olarak programlama performansı seçilirken, yordayıcı değişkenler olarak uzamsal yönelim, zihinsel döndürme, uzamsal bellek, sözel bellek ve öz yeterlilik algısı sürekli bağımsız değişkenler; üniversite, ön deneyim ve cinsiyet ise kategorik bağımsız değişkenler olarak seçilmiştir.

Rastgele ağaç analizleri için toplam ağaç sayısı 500 olarak girilmiştir. Geçerlilik için kullanılacak rastgele test veri oranı (random test data proportion) 0.1 (%10) olarak, en uygun alt örneklem değeri ise 0.56 olarak belirlenmiştir.

Durdurma parametreleri için ise varsayılan değerler kullanılmıştır. Bunlar; “maksimum düğüm sayısı” 100, “alt düğümde olabilecek minimum gözlem sayısı” 5, tüm terminal düğümlerde belirtilen minimum sayıda gözlemden daha fazla kalmayacak şekilde bölünmeyi kontrol eden “durmak için minimum sayı” 5 ve her bir ağaçtaki olabilecek maksimum seviye (kök düğümden en üst düğüme kadar olan ağacın derinliği) 10 olarak belirlenmiştir.

Analize giren toplam ağaç sayısı 500, maksimum ağaç boyutu ise 100'dür. Analiz sürecinde, test ve eğitim verisine ait ardışık sınıflama oranına ilişkin gelişim sürecini yansıtan ortalama hatalar karesi grafiği Şekil 4.5' te gösterilmektedir.



Şekil 4.5 Ortalama hatalar karesi tahmin grafiği

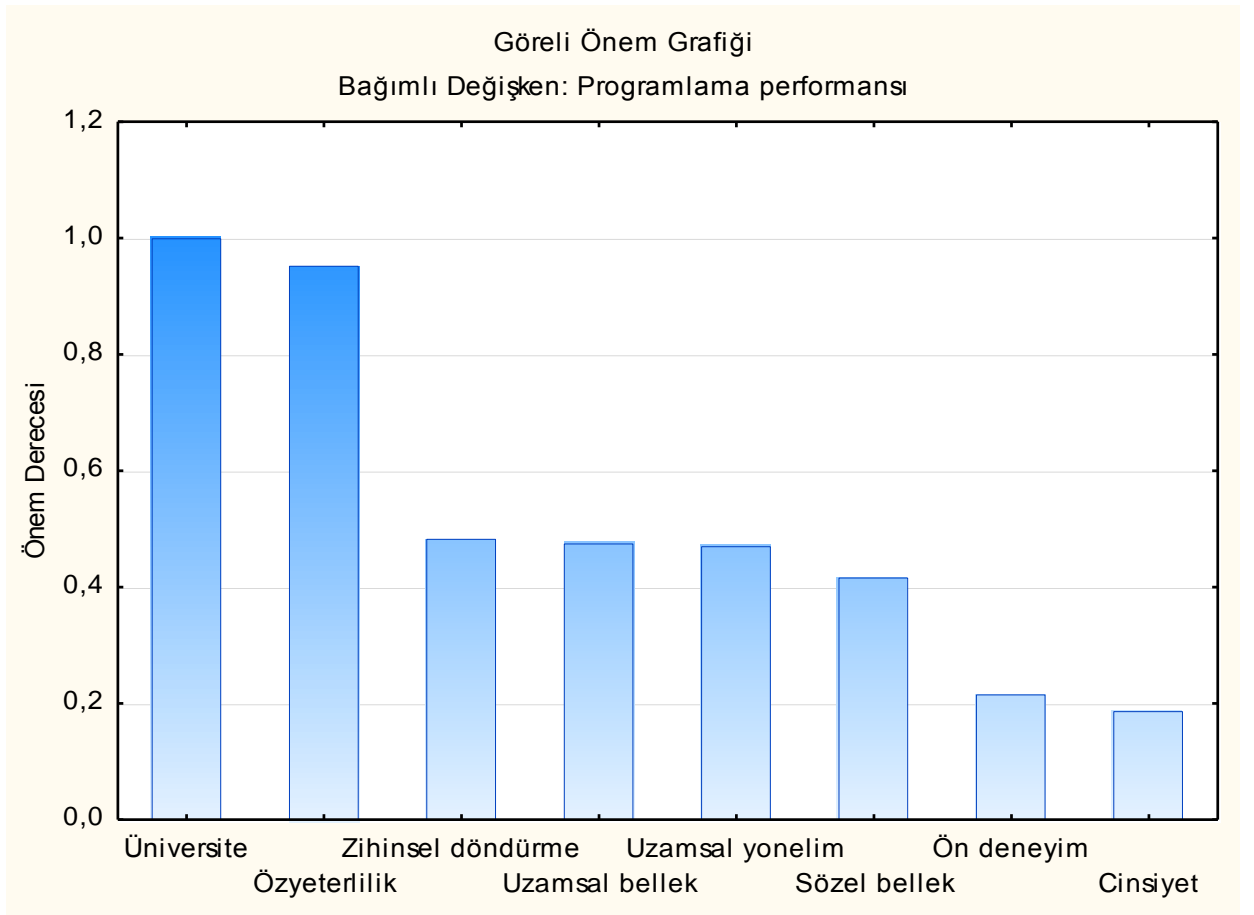
Şekil 4.5'teki grafik rastgele orman algoritmasının aşırı uyumu önleme sürecindeki temel işleyişi yansıtmaktadır. Bu modele ne kadar çok basit ağaç eklenirse eğitim setine ilişkin hatalı sınıflama da o oranda düşecektir (StatSoft, 2013).

Rastgele orman analizi sonuçlarına göre 8 değişken (zihinsel döndürme, uzamsal yönelim, görsel-uzamsal bellek, sözel bellek, programlaya ilişkin öz yeterlilik algısı, ön deneyim, üniversite ve cinsiyet) birlikte programlama performansının %60'ını yordamaktadır. Her bir değişkenin artırımlı modeldeki yordama gücünün ayrı ayrı belirlenmesinde görelî önem ağırlıkları ve değişken sırası parametreleri incelenmiştir. Değişkenlerin görelî önem ağırlıkları ve değişken sıraları Çizelge 4.6'da, bu değişkenlerin modele olan katkılarına ilişkin önem sıralarını temsil eden grafik ise Şekil 4.6'da sunulmuştur.

Çizelge 4.6 Rastgele Orman analizi açıklayıcı değişkenlere ilişkin değişken sıraları ve önem ağırlıkları

	Değişken Sırası	Önem ağırlığı
Üniversite	100	1*
Özyeterlilik	95	0,952*
Zihinsel döndürme	48	0,483*
Uzamsal bellek	47	0,474*
Uzamsal yönelim	47	0,470*
Sözel bellek	42	0,416*
Ön deneyim	22	0,215
Cinsiyet	19	0,186

* = Önem derecesi > 0.4 olan değişkenler

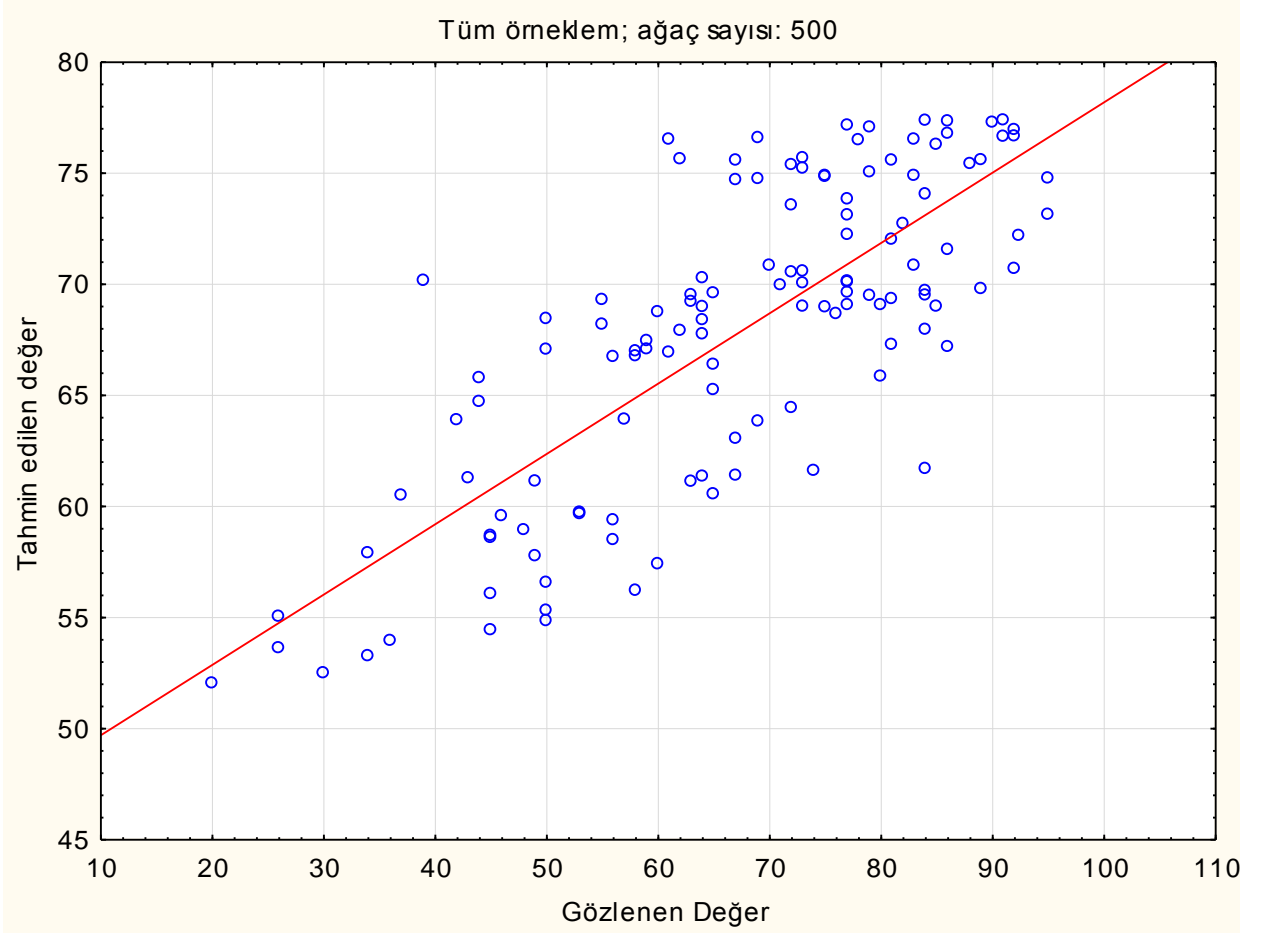


Şekil 4.6 Rastgele Orman analizi açıklayıcı değişkenlere ilişkin görelî önem katsayıları grafiđi

Çizelge 4.6'dan görüldüğü gibi programlama performansının yordanmasına ilişkin rastgele ormanlar tekniği ile kurulan modelde önem ağırlığı 0.4'ün üzerinde olarak modele anlamlı katkıda bulunan değişkenler üniversite, öz yeterlilik algısı, zihinsel döndürme, uzamsal bellek, uzamsal yönelim ve sözel bellek olarak bulunmuştur. Bu bulgu artırılmış regresyon ağaçları analizi sonucu elde edilen bulgu ile birebir örtüşmektedir. Ancak, değişkenlerin önem sırası artırılmış regresyon analizi tekniğine göre farklılaşmaktadır. Çizelgeye göre, görelî önem ağırlığı en yüksek değişken üniversite değişkeni olurken (önem derecesi= 1), öz-yeterlilik (önem derecesi= 0.952) ikinci sırada, zihinsel döndürme (önem derecesi= 0,483) üçüncü sırada, uzamsal bellek (önem derecesi= 0,474) dördüncü sırada yer almış, bunu uzamsal yönelim becerisi (önem derecesi= 0,470) izlemekte olup, altıncı sırada ise sözel bellek (önem derecesi= 0,416) değişkeni yer almıştır. Ön deneyim (önem derecesi= 0,215) ve cinsiyet (önem derecesi= 0,186) değişkenleri ise programlama performansını yordamada önem sırası bakımından son iki sırada yer almaktadır.

Şekil 4.6'dan da görüldüğü üzere programlama performansının yordanmasında üniversite ve öz-yeterlilik algısı değişkenleri birbirine yakın önem ağırlığı ile ilk iki sırada yer alırken, diğer dört bilişsel beceri de yine birbirine yakın önem ağırlıkları ile ikinci sırada yer almaktadır. Ön deneyim ve cinsiyet faktörlerinin yordamaya ilişkin katkılarının önemi ise oldukça düşmekte olup önem ağırlıkları 0.4'ün altında olması nedeniyle anlamlı olarak bulunmamıştır.

Bağımlı değişken programlama performansına ilişkin gözlemlenen ve beklenen değerler arasındaki ilişkiyi gösteren saçılım grafiği Şekil 4.7' de verilmiştir.



Şekil 4.7 Rastgele Orman analizi saçılım grafiği

Rastgele orman analizi sonuçları artırılmış regresyon ağacı analizi sonuçları ile paralellik göstermiştir. Bağımlı değişkeni yordamadaki önem ağırlıklarının oranları oldukça yakın bulunurken, modele katkı sağlayan değişkenler aynı, sıralamaları farklı olarak bulunmuştur. Her iki modelde de bilişsel beceriler, öz yeterlilik algısı ve üniversite önem ağırlığı bakımından modele katkı sağlayan anlamlı değişkenler olarak bulunurken, cinsiyet ve ön deneyimin önem ağırlığı oldukça düşük olarak bulunmuştur.

5. TARTIŞMA VE SONUÇ

Bu çalışma ile bireylerin programlama performansına etki eden faktörler bilişsel tabanlı bireysel farklılıklar temelinde incelenmiş, programlama performansı bu bilişsel farklılıklarla birlikte bilişsel olmayan değişkenlerle modellenerek açıklanmıştır. Bilişsel becerilerden uzamsal beceriler (uzamsal yönelim ve zihinsel döndürme) ve çalışma belleği (uzamsal bellek ve sözel bellek) temel alınarak, programlamaya ilişkin öz yeterlilik algısı, cinsiyet, ön deneyim ve üniversite değişkenleri ile birlikte programlama performansının yordanmasına ilişkin model kurulmuştur. Kurulan model veri madenciliğine dayalı yöntemlerden artırılmış regresyon ağaçları ve rastgele ormanlar tekniği ile analiz edilmiş, her iki teknik için programlama performansının yordanma oranı ve değişkenlerin önem sıraları incelenmiştir.

Programlama performansı ile bilişsel beceriler arasındaki korelasyonların incelenmesi sonucu, doğrusal ilişkinin anlamlı olmadığı bulunmuştur. Bu bulgu bireylerin bilişsel becerileri ve herhangi bir konudaki başarısı arasındaki etkileşimin dinamik olabileceğini, yani bazı becerileri yüksek bireylerin, bazı becerilerinin aynı zamanda düşük olabileceğini göstermektedir. Nitekim ilgili alanyazında Carello ve Moreno (2005) da bireylerin bilişsel sistemlerde verdiği tepkilerin genellikle bir dizi bileşen etkisinin ortak ürünü olması ile, bu bileşenler arasındaki etkileşimin karmaşık olacağını ve doğrusal bir varsayımın etkileşimi kısıtlayacağını ifade etmiştir. Luft ve diğ. (2013) de düşük ve yüksek performans ve bilişsel beceriler arasındaki ilişkinin her zaman doğrusal olamayacağını, bu nedenle özellikle karmaşık ilişkileri tanımlayabilen doğrusal olmayan yöntemlere gereksinim olduğu ifade etmiş, çalışmalarında ele aldıkları değişkenler için doğrusallık varsayımı olmayan yöntemlerden yapay sinir ağlarını kullanmışlardır.

Bilişsel tabanlı bireysel farklılıklar çalışılırken, bilişsel faktörler arasındaki etkileşimin içsel dinamiklerinin karmaşık olduğu göz ardı edilmekte ve doğrusallık ve normallik gibi birçok istatistiksel varsayımlara bağlı analizler yapılmaktadır. Hem bireysel farklılıklardan bahsederken hem normal dağılımdan söz etmek, hem her bireyin becerilerine ilişkin bir profili olduğunu ileri sürerek bireylerin bu özelliklerini modellerken hem de bu özellikleri arasında doğrusal ilişkinin her zaman anlamlı olacağını varsaymak, her bireyin tüm bilişsel özelliklerinin yüksek

ya da düşük olması gibi bir varsayımı da ortaya koymaktadır. Bu nedenle bilişsel tabanlı bireysel farklılıkları çalışırken bileşenler arasındaki etkileşimin kısıtlanmadan, sıkı varsayımlara dayanmayan yöntemlerle de analiz edilmesi önemlidir.

Çalışma kapsamında kullanılan non-parametrik ağaç tabanlı regresyon tekniklerinden artırılmış regresyon ağacı analizi, programlama performansının %65'ini yordamıştır. Bu sonuca göre programlama performansının yordanmasında en önemli değişken uzamsal yönelim becerisi olarak bulunurken, uzamsal bellek, zihinsel döndürme, öz yeterlilik algısı ve sözel bellek ise benzer önem ağırlıkları ile programlama performansının yordanmasına en çok katkıda bulunan değişkenler olarak bulunmuştur. Ön deneyim ve cinsiyetin programlama performansı üzerindeki etkisi anlamlı olarak bulunmamıştır. Bu bulgu bireylerin bilişsel becerilerinin programlama performansı üzerinde oldukça önemli etkisi olduğunu göstermektedir. Artırılmış regresyon ağaçları tekniği ile test edilen modelin çapraz geçerliliğini test etmek için model aynı zamanda rastgele ormanlar tekniği ile de test edilmiştir. Rastgele ormanlar analizi sonucuna göre programlama performansının %60'ı yordanırken, programlama performansının modellenmesine anlamlı katkıda bulunan değişkenler aynı kalmış, bunların önem sıraları değişmiştir. Her teknik ile üretilen sonuç karşılaştırıldığında yordama oranı her iki teknik için de yakın olup, anlamlı değişkenler değişmese de, artırılmış regresyon ağaçları tekniğinde yordanan oranın kısmen daha yüksek olması bu tekniğin çalışma kapsamındaki bulguların yorumlanmasında tercih edilebilir yöntem olduğunu göstermektedir. İki tekniğin ürettiği modelde bağımlı değişkenin tahmin edilmesine anlamlı katkıda bulunan değişkenlerin aynı olması ve tahmin edilme oranının yakın olması kullanılan yöntemin çapraz geçerliliğini gösterdiği şeklinde tartışılabilir. Diğer yandan, her iki modeldeki değişkenlerin önem sıralarının farklılaşması ise bu tekniklerin analiz sürecinde farklı algoritmaları işe koşmalarıyla açıklanabilir.

Artırılmış regresyon ağaçları tekniği olasılıksal gradyan artırım algoritmasını kullanırken, rastgele ormanlar ise Breiman tarafından ortaya konulan rastgele ormanlar algoritmasını işe koşmaktadır. Ayrıca artırılmış regresyon ağaçları temel olarak artırım tekniği üzerine, rastgele ormanlar ise yerine koyma tekniği üzerine

kurulmuş tekniktir. Bu nedenlerle test edilen modelin bulguların yorumlanmasında kullanılan teknikler arasından seçim yaparken, araştırmacılar, araştırma soruları temelinde, üretilen sonuçlardaki farklılıkların nedenlerini gözeterek seçim yapma yoluna gidebilir.

Bireylerin bilişsel becerilerinin programlama performanslarında belirleyici değişken olması, programlama performansında bilişsel bireysel farklılıkların göz önüne alınması gerektiğini ortaya koymaktadır. Bu bulgu ilgili alanyazındaki programlamanın bireysel bir etkinlik olduğu ve bu nedenle programlama performansının bilişsel yaklaşımla incelenmesi, bireylerin bilişsel becerilerinin ve süreçlerinin programlama sürecinde dikkate alınması gerektiğini öne süren çalışmalarla örtüşmektedir (Ambrosio ve diğ., 2011; Mead ve diğ., 2006; Weinberg, 1971; White ve Sivitanides, 2002). Benzer şekilde Sterling ve Brinthaup (2003) bir programcının başarı ölçütlerini bireysel ve grup bağlamında ortaya koydukları çalışmada, önem sırası bakımından özellikle bilişsel becerilerin bireysel bağlamdaki başarı ölçütlerinden en önemlilerinden birisi olduğunu bulmuşlardır. Bu nedenle hem akademik bağlamda programcıların eğitiminde hem de endüstriyel bağlamda programcı seçiminde her ne kadar teknik becerilere önem verilse de, bireylerin bilişsel becerilerinin ve problem çözme becerilerinin üzerine odaklanması gerektiği vurgulanmaktadır (Sterling ve Brinthaup, 2003).

Çalışmada ele alınan bilişsel becerilerden uzamsal becerilerin programlama performansının açıklanmasında önemli bir faktör olduğu bulunmuştur. Hem uzamsal yönelim becerisi hem de zihinsel döndürme becerisi programlama performansının yordanmasına oldukça yüksek katkıda bulunmuştur. Bu bulgu ilgili alanyazındaki benzer çalışmaları destekler niteliktedir. Jones ve Burnett (2007, 2008) program parçacıklarını kavramak için koddaki gezinimin gerçek hayattaki gezinimle benzer olduğunu ileri sürmüş ve yaptıkları çalışma sonunda programlama performansı ile uzamsal becerilerin yakın ilişkide olduğunu ortaya koymuşlardır. Benzer şekilde Fincher ve diğ. (2005) uzamsal beceri ile programlama performansı arasında anlamlı ve pozitif bir korelasyon olduğunu ortaya koyarken, Webb (1984) uzamsal becerilerin bilişsel beceriler içinde programlama performansının en önemli yordayıcılarından birisi olduğunu bulmuştur. Bu bulgular ışığında uzamsal becerisi yüksek ve düşük bireylerin

programlama performanslarının da farklılaşacağı ileri sürülebilir. Bu şekilde programlamaya ilişkin verilecek eğitimlerin içeriğinin de uzamsal becerinin göz önüne alınarak hazırlanması doğrultusunda etkililik artırılabilir. Ayrıca programlamaya ilişkin eğitim verilmeden önce, bireylerin uzamsal beceri düzeyleri tanılanarak, düşük bireyler için uzamsal beceriyi geliştirici uygulamaların, verilecek eğitimin niteliğini arttıracığı ifade edilebilir.

Programlama performansının modellenmesine ilişkin ele alınan diğer bilişsel faktör olan çalışma belleğinin de hem görsel-uzamsal bellek hem de sözel bellek boyutları programlama performansının açıklanmasına anlamlı katkıda bulunmuştur. Bu bulgu çalışma belleğinin programlama sürecinde önemli bir faktör olduğunu göstermektedir. İlgili alanyazında çalışma belleğinin programlama sürecindeki yerine ilişkin paralel çalışmalar bulunmaktadır. Shute (1991) programlama becerisi kazanmada, hem uygulamaya dönük hem de bildirimsel bilgiye dönük beceri kazanımında çalışma belleğinin yordayıcı bir değişken olduğunu bulurken, Bergersen ve Gustafsson (2011) da programlama bilgisi ve becerisi kazanmada çalışma belleği kapasitesinin önemli bir bireysel farklılık olduğunu ortaya koymuştur. Mancy (2007) ise programlamaya giriş dersindeki akademik not ortalamasının bireylerin çalışma belleği kapasiteleri ile oldukça yüksek düzeyde ilişkili olduğunu bulmuştur. Çalışma belleğine ilişkin bir diğer bulgu ise, alt boyutlar olarak alınan sözel bellek ve görsel uzamsal bellek arasındaki doğrusal ilişkinin pozitif yönde anlamlı olarak bulunmasıdır. Bu bulgu Baddeley ve Hitch (1974) tarafından ortaya konulan çalışma belleği modeli yapısını destekler niteliktedir. Modele göre çalışma belleği sözel bilgilerin depolandığı fonolojik depo ve görsel ve mekansal bilgilerin depolandığı görsel-uzamsal depo olmak üzere iki temel depolama sisteminden oluşur (Baddeley ve Hitch, 1974; Baddeley, 1999). Bu çalışma kapsamında ölçümlenen bellek puanlarının, çalışma belleğine ait iki alt boyutu temsil etmesiyle, çalışma belleği yapısını desteklemektedir.

Programlama, ilgili programlama dilinin yazım kurallarını bilme, kod okuma, hata ayıklama gibi birçok sözel beceri gerektirmesiyle hem sözel çalışma belleğinin, diğer yandan kod parçacıkları arasında gezinim, döngüler arası geçişler ve ilişkileri kavrama, algoritmanın akışını takip edebilme gibi uzamsal süreçler içermesiyle

görsel-uzamsal çalışma belleğinin rolü çalışmada önemli olarak ortaya çıkmıştır. Benzer şekilde, Irons (1982) da sözel becerilerin programlama performansının anlamlı bir yordayıcısı olarak bulunduğu çalışmada, her ne kadar programlamaya ilişkin görevlerde sayısal-matematiksel becerilerin önemli olduğu vurgulansa da programlama dillerine ilişkin dilbilgisi/yazım yanlışı gibi performanslarda sözel becerilerin de önemli olduğunu ifade etmiştir. Bu çalışmanın bulguları ve ilgili alanyazındaki bulgular, bireylerin çalışma belleği kapasitelerinin programlama eğitimi sürecinde göz önüne alınması gerektiğini göstermektedir. Özellikle çalışma belleği kapasiteleri düşük bireyler için verilen programlama eğitiminin içeriğinin bellek stratejilerine uygun şekilde hazırlanması yararlı olabilir.

Öz-yeterlilik algısı modelde programlama performansının yordanmasına ilişkin her iki analiz yönteminde de önemli bir değişken olarak ortaya çıkmıştır. Bu bulgu ilgili alanyazında bireylerin öz yeterlilik algılarının başarılarını doğrudan etkilediğine ilişkin yapılan çalışmalarla paraleldir (Cegielski ve Hall, 2006; Ramalingam ve diğ., 2004; Wiedenbeck, 2005). Öz yeterlilik algısının performansı doğrudan etkilemesi, bireyin kendine olan inancın yüksek olması durumunda öğrenmeye ilişkin motivasyonun da yüksek olması ve kaygı düzeyinin düşük olması ile açıklanabilir. Yine öz yeterlilik algısı düşük bireylerin programlama görevlerinin başlangıcında kendilerine olan inançları yetersiz olduğundan, görevi zor olarak algılayarak başarısızlık oranının yüksek olduğu şeklinde yorumlanabilir.

Çalışmada üniversite faktörü programlama performansının yordanmasına katkıda bulunan önemli bir değişken olarak ortaya çıkmıştır. Bu bulgu, her üniversitede verilen programlaya giriş dersinin içeriği her ne kadar benzer olsa da derse giren öğretim üyesi, dersin değerlendirilmesi, sınavların puanlanması gibi çeşitli faktörlerin farklılaşmasıyla açıklanabilir. Mancy (2007) çalışma belleği ve programlama performansı arasındaki ilişkiyi incelediği çalışmada, çalışma belleğinin ölçümünde kullandığı testlerin puanlanma biçimini değiştirdiğinde çalışma belleği ve programlama performansı arasındaki korelasyon katsayısının da değiştiğini bulmuştur. Bu nedenle farklı üniversitelerde farklı sınav soruları kullanılarak, puanlama biçimi de farklılaşabileceğinden, üniversite faktörü programlama performansında belirleyici bir rol oynamış olabilir.

Çalışmada ön deneyim programlama performansının yordanmasında önemli bir değişken olarak ortaya çıkmamıştır. İlgili alanyazında programlama performansının ön deneyimle ilişkisini inceleyen çalışmalarda ise farklı bulgular söz konusudur. Bazı çalışmalarda ön deneyim programlamaya ilişkin başarının anlamlı bir yordayıcısı olarak ortaya çıkarken (Morrison ve Newman, 2001; Ramalingam ve diğ., 2004; Wiedenbeck, 2005), bazı çalışmalarda ise ön deneyime sahip olup olmamanın programlama performansı ile ilişkisi bulunmamıştır (Jegede, 2009). Ön deneyime ilişkin farklı bulgular olmasının nedeni, ön deneyim olarak alınan verilerin her çalışmada farklılaşması olabilir. Bazı çalışmalarda geçmişte alınan programlamaya ilişkin ders sayısı, bazı çalışmalarda ise kaç yıldır program yazıldığı, daha önce programlama yapılıp yapılmadığı gibi farklı bilgilerle ele alınmaktadır. Jegede (2009) ön deneyimi dört farklı boyutta ele almış ve bunlardan programlama deneyimi ve program yazma yılı sayısının öz yeterlilikle ilişkili olmadığını bulurken, alınan programlama ders sayısı ve programlama derslerine ilişkin başarı notlarının özyeterlilik ile anlamlı ilişkili olduğunu bulmuştur.

Morrison ve Newman (2001) ise ön deneyim ve programlama performansı arasındaki ilişkinin farklılaşmasına ilişkin farklı bir bakış açısı sunarak, ön deneyimin programlama performansı üzerindeki etkisinin daha önce programlama dersinin nerede alındığı (ortaokul, lise ya da üniversite) ve hangi programlama diline ilişkin ön bilginin olduğunun sonraki programlama performansını farklı şekilde etkilediğini ortaya koymuştur. Bu çalışmada da öğrencilerin farklı lise türlerinden gelmesi, bazı bireylerin iki yıllık yüksek okullardan devam etmesi ile ön deneyim olarak alınan değişken her birey için aynı olmayabilir. Hagan ve Markham (2000) ise ön deneyimin programlamaya giriş derslerinin başlangıcında bir avantaj olabilirken, zaman ilerledikçe bu avantajın önceki programlama deneyimine ilişkin dilin öğretilen dille aynı olup olmamasına ya da daha önceki deneyimden elde edilen ön yargılara göre zamanla farklılaşabileceğini ifade etmiştir.

Ele alınan değişkenlere ilişkin kurulan modelde son olarak cinsiyetin programlama performansının yordanmasına ilişkin önemli bir değişken olmadığı ortaya çıkmıştır. Bu da bireylerin programlama performanslarının kadın ya da erkek olma durumlarına göre farklılaşmadığını ortaya koymaktadır. Bu bulgu ilgili alanyazındaki birçok çalışma ile paralellik göstermektedir. Byrne ve Lyons (2001),

Milic (2009), Pillay ve Jugoo (2005), Wilson ve Shrock (2001), programlama performansına etki eden faktörleri belirlemeye dönük çalışmalarında cinsiyetin anlamlı bir etkisi olmadığını ortaya koymuşlardır. Diğer yandan programlama performansında cinsiyetin önemli bir rolü olduğuna dair karşıt bulgular da söz konudur. Lau ve Yuen (2011) programlama performansında erkekler lehine bulgular ortaya koyarken, Yurdugül ve Aşkar (2013) programlama bilgisinin gelişimine ilişkin yaptıkları çalışmada erkeklerdeki erişimin daha fazla olduğunu, Aşkar ve Davenport (2009) ise programlama performansını doğrudan etkileyen faktörlerden birisi olan programlamaya ilişkin öz yeterlilik algısının yine erkeklerde anlamlı derece yüksek olduğunu bulmuşlardır. Beyler, DeKeuster, Walter, Colar ve Holcomb (2005) ise cinsiyete ilişkin farklılıkların programlamaya giriş derslerinin başlangıcında söz konusu olduğunu fakat, ders dönemi sonuna doğru bu farklılığın azaldığını ortaya koymuşlardır.

Bu çalışmada programlama performansı olarak alınan veriler bireylerin dönem sonu notlarıdır. Bundan dolayı, bireylerin dönem başında var olabilecek olası cinsiyete ilişkin farklılıklar, dönem sonunda kaybolmuş olabilir. Diğer yandan programlama performansındaki cinsiyet farklılıklarının temel nedenlerinden birisinin erkeklerin daha fazla ön deneyime sahip olduğu ile ilgili olduğu ifade edilmektedir (Murphy, McCauley, ve Westbrook, 2006). Bu çalışmada ön deneyimi olan ve olmayan grup açısından kadın ve erkeklerin sayısının hemen hemen eşit olması programlama performansı açısından cinsiyete ilişkin farklılıkların olmamasının nedenleri arasında tartışılabilir.

Bu çalışma bireylerin programlama performansını bireysel farklılıklara ilişkin çeşitli değişkenlerle açıklamaya dönük olarak yapılmış olup, programlama sürecinde bireyin bilişsel becerilerinin önemini vurgulamaktadır. Çalışmada programcı olarak incelenen kişiler Programlama Dilleri-I dersini alan öğrencileri ve programlama performansı ise bu dersten alınan akademik yıl sonu notlarını kapsamaktadır. Robins, Rountree ve Rountree (2003), programlamayı öğrenme ve öğretme süreçlerine ilişkin yapılan çalışmaları inceleyerek bu çalışmalardan psikolojik/egitsel bakış açısı kapsamında yer alanların ise 1) acemi/uzman karşılaştırmaları, 2) programlama bilgisi ve programlama stratejisi çalışmaları, 3) programı kavrama/anlama ve yeniden program üretme karşılaştırma çalışmaları, 4) biçimsel programlama dilleri ve nesneye yönelik programlama dilleri

karşılaştırma çalışmaları olmak üzere dört farklı genel sınıf altında toplamıştır. Acemi ve uzman karşılaştırmalarını bu çalışmada ele alınan değişkenler çerçevesinde düşünüldüğünde, acemi ve uzman bireylerin bilişsel becerilerinin programlama performansını açıklamaya ilişkin önem sıraları da farklılaşabilir. Örneğin çalışma belleği yüksek bireyler programlama konusunda acemi olsalar dahi programlama bilgisini öğrenme ve bu bilgileri işe koştaki becerileri geliştirmede çeşitli bellek stratejilerinden yararlanarak daha etkili bir öğrenme sağlayabilirler. Karşı taraftan bakıldığında ise uzman bireyler birçok noktada otomatikleştiklerinden ekstra bilişsel çaba ya da bellek stratejisine ihtiyaç duymadan programlama görevlerini tamamlarlar. Bu noktada çalışma belleğinin önemi acemi bireyler için uzman bireylere göre daha önemli olabilir. Robins ve diğ. (2003) de programlamada acemi olan bireyleri her ne kadar bilişsel beceriler çerçevesinden olmasa da genel olarak, etkin acemiler ve yeteneksiz acemiler olmak üzere iki sınıfa ayırarak, etkin acemilerin programlamayı çok fazla çaba ve ekstra destek gerektirmeden kavrayarak öğrenen, yeteneksiz acemilerin ise çok kolay öğrenemeyen, ekstra dikkat ve çaba ile daha uzun sürede öğrenen bireyler olduğunu vurgulamıştır. Bu çalışmanın genel çerçevesi düşünüldüğünde ise bireyler her ne kadar ön deneyimi olan ve olmayan bireyler olarak gruplandıysa da ön deneyim acemi ya da uzman olmak için yeterli bir gösterge değildir. Bu nedenle acemi ve uzman bireylerin bilişsel becerilerinin programlama performansını açıklamaya ilişkin önem sıralarının farklılaşıp farklılaşmayacağına dair çıkarımda bulunmak bu çalışmanın bulgularına dayalı olarak mümkün değildir.

Programlama bilgisi ve programlama stratejisi karşılaştırma çalışmaları bu çalışma kapsamında düşünüldüğünde de, yine öğrenilen programlama diline ilişkin değişken isimleri, döngü, fonksiyon, koşul durumları gibi kavramsal bilgiler bireyin programlama bilgisi kapsamında olduğundan bu bilgilerin öğrenilmesi ve kavranması bireylerin sözel belleğini ön plana çıkarmaktadır. Diğer yandan programlama diline ilişkin kavramsal bilgilerin işe koşularak karmaşık programlar yazılması, fonksiyon, döngü ve koşula bağlı durumlar arasındaki ilişkiyi kavrama, programın algoritmasını oluşturma ve bütün bir programı zihinde temsil etme, kod parçacıkları arasında gezinim gibi işlemler işe bireyin programlama stratejisini gerektirdiğinden bu işlemler için bireylerin görsel-uzamsal çalışma bellekleri ve uzamsal becerileri ön plana çıkabilir. Bu şekilde programlama performansı

programlama bilgisi (bildirimsel bilgi) ve programlama stratejisi olarak incelendiğinde performansı açıklamaya dönük ele alınan bilişsel becerilerin önem sıraları da değişebilir. Nitekim bu çalışma kapsamında programlama performansı bireylerin yıl sonu akademik not ortalaması olarak alındığından ve programlama performansının ölçümü yazılı sınav ile yapıldığından, bu çalışmanın verileri ile böyle bir ayrıma gitmek mümkün değildir. Benzer şekilde yeniden program üretme ve var olan programı kavrama ya da nesne tabanlı programlama dilleri ve biçimsel programlama dillerine ilişkin karşılaştırma çalışmalarında bilişsel becerilerin ve bu çalışma kapsamındaki diğer değişkenlerin önem sıralarını inceleyerek tartışmak, bu çalışma kapsamındaki veriler ışığında mümkün değildir.

Programlama performansı ölçümleri, bu çalışmada da olduğu gibi, genel olarak programlama dersini ya da programlamaya ilişkin eğitimi aldıktan hemen sonra yapılmaktadır. Fakat yıl sonu final sınavlarının ya da dersten sonra yapılan diğer beceri ölçümlerinin her zaman bireyin performansını yansıtmayabileceği, çünkü bireylerin kurallar, gerçekler, formüller ve şemalar gibi sahip oldukları bilişsel yapılara, tekrar ya da alıştırmaya gibi etkinliklerle erişemedikleri sürece unutmaya geçtiği ve bunun da programlama yeterliliğine ilişkin sürdürülebilirliği ortadan kaldırdığı ifade edilmektedir (Bennedsen ve Caspersen, 2009). Programlama bireyin sahip olduğu bir beceri olduğu için sadece kısa bir süreli kullanımı değil, gerektiği zaman işe koşarak problem çözme, ürün ortaya koyma ve diğer durumlara transfer edebilmeyi kapsar. Bu nedenle beceri olarak ele alınan değişkenin verilen eğitimin hemen ardından değil eğitim etkinliğinin üzerinden belirli bir zaman geçtikten sonra, boylamsal ölçümlerle elde edilmesi sağlıklı olacaktır.

Programlamanın her ne kadar genel olarak mühendislik bilimlerinin alanı olduğu gibi bir genel yargı söz konusu olsa da, programlama performansının genel zeka, matematik becerisi, yaratıcılık, problem çözme, akıl yürütme, analitik düşünme vb. becerilerle ilişkili olması ve günümüzdeki en önemli teknoloji okuryazarlıklarından birisi olarak görülmesiyle her alandan ve her yaşta birey için önemli bir beceridir. Nitekim tüm dünyada olduğu gibi Türkiye’de de yazılım derslerinin sadece ilgili bölümlerde lisans düzeyinde verildiği, ancak yazılım geliştirmeyi sevdirmek, erken yaşta programlama mantığını kazandırmak ve ilerideki meslek hayatında farklı

alanlar seçilse dahi programlama ile ilişkili olan diğer becerilerin gelişimine katkı sağlamak üzere farklı araçlar ve ortamlar geliştirildiği (Karabak ve Güneş, 2013) ve müfredatın da yeniden düzenlendiği görülmektedir. Ülkemizde de Talim ve Terbiye Kurulunun kararı ile “İlköğretim Kurumları (İlkokul ve Ortaokul) Haftalık Ders Çizelgesinin” ortaokul kısmında değişiklik yapılarak; Bilişim Teknolojileri ve Yazılım Dersi; ortaokulların 5 ve 6. sınıflarında haftada 2 saat olmak üzere zorunlu dersler, 7 ve 8. sınıflarda da yine haftada 2 saat olmak üzere seçmeli dersler kapsamına alınmıştır (MEB, 2012). Bu ders kapsamında kazandırılması hedeflenen bilgi, beceri ve değerlere ilişkin yeterlilikler ise dört farklı başlıkta toplanmıştır; 1) Bilişim Okur-Yazarlığı, 2) Bilişim Teknolojilerini Kullanarak İletişim Kurma, Bilgi Paylaşma ve Kendini İfade Etme, 3) Araştırma, Bilgiyi Yapılandırma ve İşbirlikli Çalışma, 4) Problem Çözme, Programlama ve Özgün Ürün Geliştirme.

Bunlardan “Problem Çözme, Programlama ve Özgün Ürün Geliştirme” yeterliliğinin altında ise yazarlık ve programlama dillerini tanıma, en az bir yazarlık/programlama dilini etkili bir biçimde kullanma, kullanıcı etkileşimli programlar hazırlayabilme gibi kazanımlar yer almaktadır. Bu kazanımlar bireylere erken yaşta programlama mantığını kazandırarak, problemlere analitik yaklaşma, ortak fikir ve ürün geliştirme, yaratıcılık, problemlere farklı çözümler üretme gibi birçok beceriyi de birlikte geliştirmeyi hedeflemektedir. Bu dersin öğretiminden sorumlu kişiler ise Bilişim Teknolojileri (BT) öğretmenleridir. Bu noktada bu dersi verecek olan bireylerin de programlamaya ilişkin belirli düzeyde programlama becerisine sahip olması gerekmektedir. Bilişim teknolojileri öğretmenleri bu çalışmanın örneklem grubunu da oluşturan Eğitim Fakültesi BÖTE mezunları ya da Teknik Eğitim Fakültesi altındaki bilgisayar öğretmenliği mezunlarından oluşmaktadır. BT öğretmenleri lisans eğitimlerinde farklı dönemlerde programlama dersleri ya da programlama kullanımı gerektiren dersler almaktadırlar. Farklı üniversitelerde verilen programlama derslerinin içeriğinde tam olarak bir standartlaşma söz konusu değildir. Örneğin bu çalışma kapsamında ele alınan örnekleme oluşturan üç farklı üniversitede Programlama Dilleri-I dersinin içeriği her ne kadar benzerlik gösterse de, üniversitelerden birinde doğrudan algoritma geliştirme mantığı ile herhangi bir programlama diline girmeden değerlendirme yapılmış, bir diğer üniversitede sorular Visual Basic programlama dili üzerinden örneklendirilirken, diğer üniversitede ise C# dili üzerinden sorular üzerinden

değerlendirme yapılmıştır. Yine farklı üniversitelerde programlamaya ilişkin sunulan seçmeli ve zorunlu dersler de farklılaşmaktadır. Bu da öğretmen adaylarının mezun olduklarında sahip olacakları programlama yeterliliklerine ilişkin farklılaşmaya sebep olabilir. İlköğretim ve ortaöğretim kurumlarında öğrencilere yazılım becerisi kazandırmak üzere görev yapacak olan BT öğretmenlerinin eğitiminde programlama öğretimine bir standardizasyon sağlamak adına üniversitelerdeki BÖTE bölümü ve diğer bilgisayar öğretmenliği bölümü ders içerikleri öğretilen programlama dilleri, seçmeli programlama dersleri ve ders ağırlıkları gibi noktalarda ortak bir müfredat hazırlama yoluna gidilebilir.

6. ÖNERİLER

6.1 Araştırmaya Dönük Öneriler

- İlerdeki araştırmalarda bu model daha büyük örneklem verileriyle test edilerek, geçerliliği artırılabilir.
- İlerdeki araştırmalarda bu çalışmada ele alınan değişkenler dışında dikkat, genel zeka (g faktörü), algı gibi farklı bilişsel faktörlerle birlikte problem çözme becerileri, matematik başarıları, akademik ortalama gibi farklı değişkenlerde modele alınarak, programlama performansının açıklanmasına olan katkıları incelenebilir.
- Bu çalışma sadece üç üniversitenin verisi ile sınırlı olduğundan ve çalışmada üniversite faktörü verilen eğitimin içeriği ve değerlendirilmesi bakımından farklılaşmaya neden olduğundan, ilerideki araştırmalarda örnekleme ilişkin üniversiteler çeşitlendirilerek modelin genellenebilirliği artırılabilir.
- Bilişsel faktörlerin ele alındığı çalışmalarda, bu becerilerin aralarındaki etkileşimin doğrusal olmadığı bulgusundan yola çıkarak, bu çalışmada kullanılan doğrusallık varsayımı gözetmeyen yöntemler dışındaki yöntemler kullanılarak sonuçlar karşılaştırılabilir.
- Bu çalışmada programlama performansı olarak katılımcıların dersteki geçme notları alınmıştır. İlerdeki araştırmalarda programlama performansının araştırmacıların kontrolünde değerlendirilebilmesi için, çalışma bağlamına özgü programlama görevleri ile değerlendirilmesi programlamaya ilişkin daha geçerli bir değerlendirme ve ölçüm sunabilir.

- Programlama dilleri kendi içinde nesne tabanlı diller, görsel programlama dilleri, skript tabanlı programlı dilleri, işaretleme (mark up) dilleri vb. farklı alt gruplara ayrılmaktadır. Programlama performansını açıklayan değişkenleri belirlerken, farklı programlama dilleri için farklı beceriler gerekebileceğinden, programlama dillerine ilişkin alt sınıfların her biri ile ilişkili etkili beceriler için karşılaştırmalı çalışmalar yapılabilir. Bu şekilde programlama derslerinin içeriği hazırlanırken her tip programlama için bireylerin gereksinimlerine ve özelliklerine uygun ders tasarımları ve uygulamalar geliştirilebilir.
- Hem programlama performansının hem de bilişsel becerilerin gelişimsel bir yapıda olduğu bulgusundan yola çıkarak, bireylerin programlamaya ilişkin dersler alması ile bilişsel becerilerindeki gelişim ya da tam tersi şekilde programlama performansı ile ilişkili olduğu belirlenen bilişsel becerilerin geliştirilmesi sonucu programlama performansındaki gelişim ve bunlar arasındaki etkileşim boyutsal çalışmalarla incelenebilir.
- İlerideki çalışmalarda bireylerin programlama performansını bilişsel becerileri ile açıklamaya dönük çalışmalarda, bireyler acemi ve uzman bireyler olarak kategorilendirilerek her bir grubun programlama performansını yordayan bilişsel becerilerin önem sıraları karşılaştırılıp, buna dönük olarak acemi bireyler açısından önemli olan bilişsel beceriler belirlenerek programlama eğitiminin yanı sıra bu becerilerin geliştirilmesine dönük uygulamalar geliştirilebilir. Benzer şekilde bilişsel beceriler bakımından bireyler yüksek ve düşük olarak gruplandırılarak her bir gruptaki acemi ve uzman bireyler için programlama performansını yordayan diğer değişkenlerin etkileri incelenebilir.
- Bu çalışmada üniversite öğrencileri ile çalışılmıştır. Nitekim bilişsel beceriler gelişimsel bir yapıda olduğundan bireylerin oynadığı oyunlar, çalıştıkları meslek, günlük etkinlikleri, bilgisayar kullanımları vb. etkenler zaman içerisinde sahip olduğu bilişsel becerileri etkilemektedir. Bu nedenle ilerideki çalışmalarda ilköğretim gibi daha küçük yaş gruplarında programlama performansı ile ilişkili bilişsel becerilerin incelenmesi gerek bilişsel beceri geliştirmeye dönük etkinlikler düzenleme açısından gerekse programlama eğitimi açısından yararlı olabilir.

6.2 Uygulamaya Dönük Öneriler

- Programlama performansı üzerinde bilişsel becerilerin etkisi olduğundan ve bu becerilerin ilgili alan yazıda geliştirilebilir olduğuna dair bulgular yer aldığından, programlamaya ilişkin eğitim sürecinde bu becerileri düşük olan bireylerin becerilerinin geliştirilmesine dönük uygulamalarla eğitim verilmesi, programlama eğitimin etkililiği açısından yararlı olabilir.
- Programlama eğitimi verilemeden önce bireylerin çalışma belleği ve uzamsal becerilerine ilişkin tanılayıcı testler uygulanarak, beceri seviyeleri belirlenebilir ve programlama eğitiminin içeriği bu becerilere uygun şekilde hazırlanabilir.
- Öz yeterlilik algısı, yani bireyin kendi başarısına olan inancı programlama performansını etkileyen oldukça önemli bir faktör olduğundan bireylere özellikle programlamaya giriş derslerinden önce dersin içeriğine yönelik motive edici ve öz yeterlilik algısını artırıcı etkinlikler düzenlenebilir. Yine dersin başlangıç kısımlarındaki içeriğin özellikle basit ve anlaşılır hazırlanarak baştan yıldırıcı olmayacak şekilde tasarlanması, öz yeterlilik algılarının şekillenmesine olumlu etki edebilir.
- Üniversite faktörü programlama performansında belirleyici bir role sahip olduğundan, ileride ilköğretim ve ortaöğretim kurumlarında programlamaya ilişkin eğitimi verecek olan Bilişim Teknolojilerinin lisans eğitimlerindeki programlama dersleri sayısı, içeriği ve değerlendirme biçimi bakımından standart hale getirilebilir.

KAYNAKÇA

- Ackerman, P. L., Beier, M. E., & Bowen, K. R. (2002). What we really know about our abilities and our knowledge. *Personality and Individual Differences*, 33, 587-605.
- Ahmed, I., & Blustein, J. (2005, 23-25 February). *Navigation in information space: How does Spatial ability play a part?* Paper presented at the Web Based Communities 2005 (IADIS), Algarve, Portugal.
- Akıncı, A., & Tüzün, H. (2012). *Okuryazarlığa yeni bir bakış: Programlama okuryazarlığı*. Paper presented at the 6th International Computer Education and Instructional Technologies Symposium, Gaziantep, Turkey
- Alonso, D. L. (1998). *The Effects of Individual Differences in Spatial Visualization Ability on Dual-Task Performance*. Doctor of Philosophy, University of Maryland.
- Alspaugh, C. A. (1972). Identification of Some Components of Computer Programming Aptitude. *Journal for Research in Mathematics Education*, 3(2), 89-98. doi: 10.2307/748667
- Altun, A. (2003). E-okuryazarlık. *Milli Eğitim Dergisi*, 158.
- Altun, A., & Mazman, S. G. (2012). Programlamaya ilişkin Öz Yeterlilik Algısı Ölçeğinin Türkçe Formunun Geçerlilik ve Güvenirlik Çalışması. *Eğitimde ve Psikolojide Ölçme ve Değerlendirme Dergisi*, 3(2), 297- 308.
- Alwin, D. F. (1994). Aging, Personality and Social Change: The Stability of Individual Differences Over the Adult Life-Span. In D. L. Featherman, R. M. Lerner & M. Perlmutter (Eds.), *LifeSpan Development and Behavior* (Vol. 12, pp. 135-185). Hillsdale, N.J: Lawrence Erlbaum Associates, Inc.
- Ambrosio, A. P., Costa, F. M., Almeida, L., Franco, A., & Macedo, J. (2011, 12-15 Oct. 2011). *Identifying cognitive abilities to improve CS1 outcome*. Paper presented at the Frontiers in Education Conference (FIE), 2011.
- Aşkar, P., & Davenport, D. (2009). An Investigation of Factors Related to Self-Efficacy for Java Programming Among Engineering Students. *The Turkish Online Journal of Educational Technology - TOJET*, 8(1).
- Austin, H. S. (1987). Predictors of Pascal programming achievement for community college students. *SIGCSE Bull*, 19(1), 161-164.
- Baddeley, A. (1992). Working memory. *Science*, 255(5044), 556-559.
- Baddeley, A. (1999). *Essentials of human memory*: Psychology Press/Taylor & Francis (UK).
- Baddeley, A., & Hitcher, G. (1974). Working memory. In G. H. Bower (Ed.), *The psychology of learning and motivation: Advances in research and theory* (pp. 47-87): Academic Press.
- Bandura, A. (1977). Self-efficacy: Toward a unifying theory of behavioral change. *Psychological Review*, 84, 191-215.

- Bandura, A. (1994). Self-efficacy. In V. S. Ramachaudran (Ed.), *Encyclopedia of human behavior*. (Vol. 4, pp. 71-81). New York: Academic Press.
- Baran, B., Dogusoy, B., & Cagiltay, K. (2007). *How do adults solve digital tangram problems? Analyzing cognitive strategies through eye tracking approach*. Paper presented at the 12th international conference on Human-computer interaction: intelligent multimodal interaction environments (HCI 2007), Berlin, Heidelberg.
- Bennedsen, J., & Caspersen, M. E. (2009). *Recalling Programming Competence*. Paper presented at the 9th Koli Calling International Conference on Computing Education Research, Koli National Park, Finland.
- Bergersen, G. R., & Gustafsson, J.-E. (2011). Programming Skill, Knowledge, and Working Memory Among Professional Software Developers from an Investment Theory Perspective. *Journal of Individual Differences*, 32(4), 201-209. doi: 10.1027/1614-0001/a000052
- Bergin, S., & Reilly, R. (2005). Programming: Factors that Influence Success. *ACM SIGCSE Bulletin.*, 37(1).
- Bergin, S., & Reilly, R. (2006). Predicting introductory programming performance: a multi-institutional multivariate study. *Computer Science Education*, 16(4), 303-323.
- Beyer, S., DeKeuster, M., Walter, K., Colar, M., & Holcomb, C. (2005). Changes in CS students' attitudes towards CS over time: an examination of gender differences. *SIGCSE Bull.*, 37(1), 392-396. doi: 10.1145/1047124.1047475
- Blasko, D., Holliday-Darr, K., Mace, D., & Blasko-Drabik, H. (2004). VIZ: The visualization assessment and training Web site. *Behavior Research Methods, Instruments, & Computers*, 36(2), 256-260. doi: 10.3758/BF03195571
- Blustein, J., & Satel, J. (2003). Spatial Ability and Information Shape: When do individual differences matter *Technical Report CS-2003-11*. Canada: Faculty of Computer Science Dalhousie University.
- Breiman, L. (2001). Random forests. *Machine Learning*, 45(5-32.).
- Breiman, L., & Cutler, A. (2004). Random Forests. 12.07.2013.tarihinde, http://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm adresinden erişilmiştir
- Bruyer, R., & Brysbaert, M. (2011). Combining Speed and Accuracy in Cognitive Psychology: Is the Inverse Efficiency Score (IES) a Better Dependent Variable than the Mean Reaction Time (RT) and the Percentage of Errors (PE)? *Psychologica Belgica*, 51(1), 5-13.
- Buston, P. M., & Elith, J. (2011). Determinants of reproductive success in dominant pairs of clownfish: a boosted regression tree analysis. *Journal of Animal Ecology*, 80(3), 528-538.
- Byrne, P., & Lyons, G. (2001). The Effect of Student Attributes on Success in Programming. *SIGCSE Bull.*, 33(3), 49-52.

- Campbell, S. G., & Norman, K. L. (2010). *Cognitive Abilities and the Measurement of World Wide Web Usability*. Paper presented at the Proceedings of the Human Factors and Ergonomics Society Annual Meeting.
- Capstick, C. K., Gordon, J. D., & Salvadori, A. (1975). Predicting Performance by University Students in Introductory Computing Courses. *ACM SIGCSE Bulletin*, 7(3), 21-29.
- Carello, C., & Moreno, M. A. (2005). Why nonlinear methods. *Tutorials in contemporary nonlinear methods for the behavioral sciences*, 1-25.
- Carroll, J. B. (1993). *Human Cognitive Abilities: A Survey of Factor-Analytic Studies*: Cambridge University Press.
- Carty, D. M. (2011). *An Analysis of Boosted Regression Trees to Predict the Strength Properties of Wood Composites*. Master of Science Degree, The University of Tennessee, Knoxville.
- Caspersen, M. E. (2007). *Educating Novices in the Skills of Programming*. PhD, University of Aarhus Denmark.
- Cegielski, C. G., & Hall, D. J. (2006). What makes a good programmer? *Communications of the ACM*, 49(10), 73-75.
- Charlton, J. P., & Birkett, P. E. (1999). An integrative model of factors related to computing course performance. *Journal of Educational Computing Research*, 20(3), 237-257.
- Chen, C. (2000). Individual differences in a spatial-semantic virtual environment. *Journal of the American Society for Information Science*, 51(6), 529-542. doi: 10.1002/(SICI)1097-4571(2000)51:6<529::AID-ASIS>3.0.CO;2-F
- Clarke, B., Fokoue, E., & Zhang, H. H. (2009). *Principles and theory for data mining and machine learning*. New York: Springer.
- Conway, A. R., Jarrold, C., Kane, M. J., Miyake, A., & Towse, J. N. (2007). Variation in working memory: An introduction 3-17.: Oxford University Press.
- Cox, A., Fisher, M., & O'Brien, P. (2005). *Theoretical Considerations on Navigating Codespace with Spatial Cognition*. Paper presented at the 17th Workshop of the Psychology of Programming Interest Group, Sussex University.
- Çevik, V. (2012). *Karmaşık bilişsel görev performansında çalışma belleği kapasitesinin ve öğretimsel stratejinin rolü*. Yayınlanmamış Doktora Tezi, Hacettepe Üniversitesi, Ankara.
- Daneman, M., & Carpenter, P. A. (1980). Individual differences in working memory and reading. *Journal of Verbal Learning and Verbal Behavior*, 19(4), 450-466. doi: [http://dx.doi.org/10.1016/S0022-5371\(80\)90312-6](http://dx.doi.org/10.1016/S0022-5371(80)90312-6)
- Daneman, M., & Merikle, P. M. (1996). Working memory and language comprehension: A meta-analysis. *Psychonomic Bulletin & Review* 1996, 3(4), 422-433.
- Davidsson, K., Larzon, L.-Å., & Ljunggren, K. (2010). Self-Efficacy in Programming among STS Students. *Reports*. Uppsala University, Department of

Information Technology. 20.05.2013.tarihinde,
<http://www.it.uu.se/edu/course/homepage/datadidaktik/ht10/reports/Self-Efficacy.pdf> adresinden erişilmiştir

- De'ath, G., & Fabricius, K. E. (2000). Classification and regression trees: a powerful yet simple technique for ecological data analysis. *Ecology*, 8(11), 3178–3192.
- Dehn, M. J. (2011). *Working memory and academic learning: Assessment and intervention*. Hoboken, New Jersey: John Wiley & Sons.
- deRaadt, M., Hamilton, M., Lister, R., Tutty, J., Baker, B., Box, I., & Tolhurs, D. (2005). *Approaches to learning in computer programming students and their effect on success*. . Paper presented at the 28th HERDSA Annual Conference: Higher Education in a Changing World (HERDSA 2005)
- Elith, J., Leathwick, J. R., & Hastie, T. (2008). A working guide to boosted regression trees. *Journal of Animal Ecology*, 77(4), 802-813.
- Emurian, H. H., Hu, X., Wang, J., & Durham, A. G. (2000). Learning Java: a programmed instruction approach using Applets. *Computers in Human Behavior*, 16(4), 395-422. doi: [http://dx.doi.org/10.1016/S0747-5632\(00\)00019-4](http://dx.doi.org/10.1016/S0747-5632(00)00019-4)
- Erdoğan, B. (2005). *Programlama başarısı ile akademik başarı, genel yetenek, bilgisayara karşı tutum, cinsiyet ve lise türü arasındaki ilişkilerin incelenmesi*. Yüksek lisans, Marmara Üniversitesi İstanbul.
- Erdoğan, Y., Aydın, E., & Kabaca, T. (2008). Exploring the psychological predictors of programming achievement. . *Journal of Instructional Psychology*, 35(3), 264-270.
- Fincher, S., Baker, B., Box, I., Cutts, Q., Raadt, M. d., Haden, P., . . . Tutty, J. (2005). Programmed to succeed?: A multi-national, multi-institutional study of introductory programming courses *Technical Report*. Computing Laboratory: University of Kent.
- Fisher, M., Cox, A., & Lin, Z. (2006). *Using Sex Differences to Link Spatial Cognition and Program Comprehension*. Paper presented at the 22nd IEEE International Conference on Software Maintenance (ICSM '06).
- Friedman, J. H. (2001). Greedy Function Approximation: A Gradient Boosting Machine. *The Annals of Statistics*, 29(5), 1189-1232. doi: 10.2307/2699986
- Glorfeld, L. W., & Fowler, G. C. (1982). Validation of a model for predicting aptitude for introductory computing. *SIGCSE Bull.*, 14(1), 140-143. doi: 10.1145/953051.801355
- Grant, N. S. (2003). *A study on critical thinking, cognitive learning style, and gender in various information science programming classes*. Paper presented at the Proceedings of the 4th conference on Information technology curriculum, Lafayette, Indiana, USA.
- Gürer, D., & Camp, T. (2002). An ACM-W literature review on women in computing. *SIGCSE Bull*, 34(2), 121-127.
- Haavisto, M.-L., & Lehto, J. E. (2005). Fluid/spatial and crystallized intelligence in relation to domain-specific working memory: A latent-variable approach.

Learning and Individual Differences, 15(1), 1-21. doi:
<http://dx.doi.org/10.1016/j.lindif.2004.04.002>

- Hagan, D., & Markham, S. (2000). Does It Help to Have Some Programming Experience Before Beginning a Computing Degree Program? *ACM SIGCSE Bulletin*, 32(3), 25-28.
- Halpern, D. F. (2000). *Sex differences in cognitive abilities*. Mahwah, NJ, USA: Lawrence Erlbaum Associates.
- Hannay, J. E., Arisholm, E., Engvik, H., & Sjoberg, D. I. K. (2010). Effects of Personality on Pair Programming. *Software Engineering, IEEE Transactions on*, 36(1), 61-80. doi: 10.1109/TSE.2009.41
- Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: data mining, inference and prediction*: Springer-Verlag.
- Holvikivi, J. (2010). Conditions for Successful Learning of Programming Skills. In N. Reynolds & M. Turcsányi-Szabó (Eds.), *Key Competencies in the Knowledge Society* (Vol. 324, pp. 155-164): Springer Berlin Heidelberg.
- Hoskinson, P. (2012). Brain Workshop – a Dual N-Back game 20.12. 2012 tarihinde, <http://brainworkshop.sourceforge.net/> adresinden erişilmiştir
- Howard, E. V. (2002). *Can We Teach Introductory Programming as a Liberal Education Course? Yes, We Can!* . Paper presented at the Information Systems Education Conference San Antonio, TX.
- Irons, D. M. (1982). *Cognitive correlates of programming tasks in novice programmers*. Paper presented at the Proceedings of the 1982 Conference on Human Factors in Computing Systems, Gaithersburg, Maryland, USA.
- Jarrold, C., & Towse, J. N. (2006). Individual differences in working memory. *Neuroscience*, 139(1), 39-50. doi:
<http://dx.doi.org/10.1016/j.neuroscience.2005.07.002>
- Jegade, P. O. (2009). Predictors Of Java Programming Self Efficacy Among Engineering Students. *International Journal of Computer Science and Information Security*, 4(30).
- Jenkins, T. (2002). *On the difficulty of learning to program* . Paper presented at the 3rd Annual Conference of the LTSN Centre for Information and Computer Sciences Loughborough University.
- Jonassen, D. H., & Grabowski, B. L. (1993). *Handbook of individual differences learning and instruction*: Routledge.
- Jones, S. J., & Burnett, G. (2008). Spatial Ability and Learning to Program. *Human Technology*, 4(1), 47-61.
- Jones, S. J., & Burnett, G. E. (2007). Spatial skills and navigation of source code. *SIGCSE Bull.*, 39(3), 231-235. doi: 10.1145/1269900.1268852
- Jun, S.-H. (2013). Boosted regression trees and random forests *Statistical Consulting Report*. Department of Statistics, UBC.
- Karabak, D., & Güneş, A. (2013). Ortaokul Birinci Sınıf Öğrencileri İçin Yazılım Geliştirme Alanında Müfredat Önerisi. *Eğitim ve Öğretim Araştırmaları Dergisi*, 2(3), 175-181.

- Khooshabeh, P., & Hegarty, M. (2010). *Representations of Shape during Mental Rotation*. Paper presented at the Association for the Advancement of Artificial Intelligence Spring Symposium: Cognitive Shape Processing.
- Kimura, D. (2000). *Sex and Cognition*: The MIT Press.
- Kozhevnikov, M., & Hegarty, M. (2001). A dissociation between object manipulation spatial ability and spatial orientation ability. *Memory & Cognition*, 29(5), 745-756. doi: 10.3758/BF03200477
- Lau, W. W. F., & 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), 696-712.
- Lau, W. W. F., & Yuen, A. H. K. (2011). Modeling programming performance: Beyond the influence of learner characteristics. *Computers & Education*, 57(1), 1202-1213.
- Lawton, C. (2010). Gender, Spatial Abilities, and Wayfinding. In J. C. Chrisler & D. R. McCreary (Eds.), *Handbook of Gender Research in Psychology* (pp. 317-341): Springer New York.
- Lehman, S., Bruning, R., & Horn, C. (1983). A tool for improving critical thinking in web-based instruction: Two experimental studies. *The CLASS project*. The Center for Instructional Innovation of the University of Nebraska.
- Leverington, M. E. (2010). *The Five Step Programming Process Master*, University of Nevada
- Liaw, A., & Wiener, M. (2002). Classification and Regression by random Forest. *R News*, 2(3), 18-22.
- Linn, M. C., & Petersen, A. C. (1985). Emergence and Characterization of Sex Differences in Spatial Ability: A Meta-Analysis. *Child Development*, 56(6), 1479-1498. doi: 10.2307/1130467
- Luft, C. D. B., Gomes, J. S., Priori, D., & Takase, E. (2013). Using online cognitive tasks to predict mathematics low school achievement. *Computers & Education*, 67(0), 219-228. doi: <http://dx.doi.org/10.1016/j.compedu.2013.04.001>
- Mancy, R. (2007). *Explicit and implicit learning: the case of computer programming* Doctoral dissertation, University of Glasgow.
- Mancy, R., & Reid, N. (2004). *Aspects of Cognitive Style and Programming*. Paper presented at the 16 th Workshop of the Psychology of Programming Interest Group (PPIG 16). Carlow, Ireland.
- Martin, V. M. A., & Beena, T. L. A. (2012). Prediction of Association among Numerical Aptitude, Programming Skills, Trait Emotional Intelligence on Students Performance. *International Journal on Computer Science and Engineering* 4(9), 1639-1646.
- Mauai, E., & Temese, E. (2012). Exploratory Study on Factors that Impact/Influence Success and failure of Students in the Foundation Computer Studies Course at the National University of Samoa. *Journal of Emerging Trends in Computing and Information Sciences*, 3(5), 767-773.

- Mayer, R. E., Dyck, J. L., & Vilberg, W. (1986). Learning to program and learning to think: what's the connection? *Commun. ACM*, 29(7), 605-610. doi: 10.1145/6138.6142
- Mazman, S. G., & Altun, A. (2013). Individual Differences in Spatial Orientation Performances: An Eye Tracking Study. *World journal on educational technology*, 5(2), 266-280.
- McGee, M. G. (1979). Human spatial abilities: Psychometric studies and environmental, genetic, hormonal, and neurological influences. *Psychological Bulletin*, 86(5), 889-918. doi: 10.1037/0033-2909.86.5.889
- Mead, J., Gray, S., Hamer, J., James, R., Sorva, J., Clair, C. S., & Thomas, L. (2006). A cognitive approach to identifying measurable milestones for programming skill acquisition *Working group reports on ITiCSE on Innovation and technology in computer science education %@ 1-59593-603-3* (pp. 182-194). Bologna, Italy: ACM.
- MEB. (2012). Bilişim Teknolojileri ve Yazılım Dersi (5, 6, 7 ve 8. Sınıflar) Öğretim Programı. 05.09. 2013 tarihinde, <http://ttkb.meb.gov.tr/www/ogretim-programlari/icerik/72> adresinden erişilmiştir
- Mendez, G. (2008). *Tree-based methods to model dependent data*. Doctoral Philosophy Arizona State University.
- Merrienboer, J. J. G. V., & Paas, F. G. W. C. (1990). Automation and schema acquisition in learning elementary computer programming: Implications for the design of practice. *Computers in Human Behavior*, 6(3), 273-289.
- Michelson, G. (2004). *Accurate identification of potential computer programmers*. Paper presented at the Grand Challenge in Computing: Education, 29-31 March, Tyneside.
- Milic, J. (2009). Predictors of success in solving programming tasks. . *The Teaching of Mathematics*, 12(1), 25-31.
- Miller, G. A., Galanter, E., & Pribram, K. H. (1960). *Plans and the structure of behavior*. New York, US: Adams Bannister Cox.
- Morrison, M., & Newman, T. S. (2001). A Study of the Impact of Student Background and Preparedness on Outcomes in CS I . *ACM SIGCSE Bulletin*, 33(1), 179-183.
- Murphy, L., McCauley, R., & Westbrook, S. (2006). Women Catch Up: Gender Differences in Learning Programming Concepts. *SIGCSE Bull*, 38(1), 17-21.
- Ormerod, T. (1990). Human Cognition and Programming. In J. M. Hoc, T. R. G. Green, R. Samurçay & D. J. Gillmore (Eds.), *Psychology of programming* (pp. 63-82). London: Academic Press.
- Pajares, F. (1996). Self-Efficacy Beliefs in Academic Settings. *Review of Educational Research*, 66(4), 543-578.
- Pak, R., Rogers, W. A., & Fisk, A. D. (2006). Spatial Ability Subfactors and Their Influences on a Computer-Based Information Search Task. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 48(1), 154-165.

- Pena, C. M., & Tirre, W. C. (1992). Cognitive factors involved in the first stage of programming skill acquisition. *Learning and Individual Differences*, 4(4), 311-334. doi: [http://dx.doi.org/10.1016/1041-6080\(92\)90017-9](http://dx.doi.org/10.1016/1041-6080(92)90017-9)
- Pillay, N., & Jugoo, V. R. (2005). An Investigation into Student Characteristics Affecting Novice Programming Performance. *SIGCSE Bulletin*, 37(4), 107-110.
- Prensky, M. (2008). Programming: The New Literacy. *Edutopia Magazine*.
- Ramalingam, V., LaBelle, D., & Wiedenbeck, S. (2004). *Self-efficacy and mental models in learning to program*. Paper presented at the Proceedings of the 9th annual SIGCSE conference on Innovation and technology in computer science education, Leeds, United Kingdom.
- Ramalingam, V., & Wiedenbeck, S. (1998). Development and Validation of Scores on a Computer Programming Self-Efficacy Scale and Group Analyses of Novice Programmer Self-Efficacy. *Journal of Educational Computing Research*, 19(4), 367 - 381.
- Ratcliff, R. (2002). A diffusion model account of response time and accuracy in a brightness discrimination task: Fitting real data and failing to fit fake but plausible data. *Psychonomic Bulletin & Review*, 9(2), 278-291. doi: 10.3758/BF03196283
- Rizzo, A., Buckwalter, J. G., Larson, P., Rooyen, A. v., Kratz, K., Neumann, U., . . . Thiebaut, M. (1998). *Preliminary findings on a virtual environment targeting human mental rotation/spatial abilities*. Paper presented at the 2nd European Conference on Disability, Virtual Reality and Associated Technologies, Sweden.
- Robins, A., Rountree, J., & Rountree, N. (2003). Learning and Teaching Programming: A Review and Discussion. *Computer Science Education*, 13(2), 137-172.
- Rowan, T. C. (1956). *Psychological Tests and Selection of Computer Programmers*. Paper presented at the Meeting of the Association.
- Samsudin, K. A., & Ismail, A. (2004). The improvement of mental rotation through computer based multimedia tutor. *Malaysian Online Journal of Instructional Technology*, 1(2), 24-34.
- Schapiro, R. E. (2003). *The Boosting Approach to Machine Learning An Overview*. Paper presented at the MSRI Workshop on Nonlinear Estimation and Classification, New York, USA.
- Shepard, R. N., & Metzler, J. (1971). Mental rotation of three-dimensional objects. *Science*, 171, 701-703.
- Shneiderman, B., & Mayer, R. (1979). Syntactic/semantic interactions in programmer behavior: A model and experimental results. *International Journal of Computer & Information Sciences*, 8(3), 219-238. doi: 10.1007/BF00977789
- Shute, V. J. (1991). Who is likely to acquire programming skills? *Journal of Educational Computing Research*, 7(1), 1-24. doi: 10.2190/VQJD-T1YD-5WVB-RYPJ

- Simon, Cutts, Q., Fincher, S., Haden, P., Robins, A., Sutton, K., . . . Tutty, J. (2006). *The ability to articulate strategy as a predictor of programming skill*. Paper presented at the Proceedings of the 8th Australasian Conference on Computing Education - Volume 52, Hobart, Australia.
- Simon, Fincher, S., Robins, A., Baker, B., Box, I., Cutts, Q., . . . Tutty, J. (2006). *Predictors of success in a first programming course*. Paper presented at the Proceedings of the 8th Australasian Conference on Computing Education - Volume 52, Hobart, Australia.
- Sorby, S. A., Drummer, T., Hungwe, K., & Charlesworth, P. (2005). *Developing 3-D Spatial Visualization Skills for Non-Engineering Students*. Paper presented at the ASEE Annual Conference, Portland, Oregon.
- St Clair-Thompson, H. L., & Botton, C. (2009). Working memory and science education: exploring the compatibility of theoretical approaches. *Research in Science & Technological Education*, 27(2), 139-150. doi: 10.1080/02635140902853616
- Stalcup, K. A. A. (2005). *Multimedia Learning: Cognitive Individual Differences And Display Design Techniques Predict Transfer Learning With Multimedia Learning Modules*. Doctor of Philosophy, Graduate Faculty of Texas Tech University.
- Stanney, K. M., & Salvendy, G. (1995). Information visualization; assisting low spatial individuals with information access tasks through the use of visual mediators. *Ergonomics*, 38(6), 1184-1198. doi: 10.1080/00140139508925181
- StatSoft, I. (2013). Electronic Statistics Textbook. 20.05. 2013 tarihinde, <http://www.statsoft.com/textbook/> adresinden erişilmiştir
- Sterling, G. D., & Brinthaup, T. M. (2003). Faculty and Industry Conceptions of Successful Computer Programmers. *Journal of Information Systems Education*, 14(4), 417-424.
- Thurstone, L. L. (1950 ss.518). *Some primary abilities in visual thinking*. Paper presented at the American Philosophical Society.
- Townsend, J. T., & Ashby, G. (1978). Methods of Modeling Capacity in Simple Processing Systems. In J. Castellan & F. Restle (Eds.), *Cognitive theory* (Vol. 3, pp. 200-239). Erlbaum: Hillsdale
- Tüzün, H. (2007). *Programlama 2.0: Programlama Eğitiminde Yenilikçi İnternet Teknolojilerinin Kullanılması*. Paper presented at the Akademik Bilişim'07 - IX. , Dumlupınar Üniversitesi, Kütahya.
- UNESCO. (2002). Information and Communication Technology in Education: a curriculum for schools and programme of teacher development. In E. Khvilon (Ed.). Paris.
- Vainio, V., & Sajaniemi, J. (2007). Factors in novice programmers' poor tracing skills. *SIGCSE Bulletin*, 39(3), 236-240. doi: 10.1145/1269900.1268853
- Vicente, K. J., & Williges, R. C. (1988). Accommodating individual differences in searching a hierarchical file system. *International Journal of Man-Machine Studies*, 29(6), 647-668. doi: [http://dx.doi.org/10.1016/S0020-7373\(88\)80072-5](http://dx.doi.org/10.1016/S0020-7373(88)80072-5)

- Wagenmakers, E.-J., Maas, H. J., & Grasman, R. P. P. (2007). An EZ-diffusion model for response time and accuracy. *Psychonomic Bulletin & Review*, 14(1), 3-22. doi: 10.3758/BF03194023
- Webb, N. M. (1984). Microcomputer Learning in Small Groups: Cognitive Requirements and Group Processes. *Journal of Educational Psychology*, 76(6), 1076.
- Weinberg, G. M. (1971). *The psychology of computer programming*. New York: Van Nostrand Reinhold.
- White, G. L., & Sivitanides, M. P. (2002). Theory of the Relationships between Cognitive Requirements of Computer Programming Languages and Programmers' Cognitive Characteristics. *Journal of Information Systems Education*, 13(1), 59-68.
- Wiedenbeck, S. (2005). *Factors Affecting the Success of Non-Majors in Learning to Program*. Paper presented at the First international workshop on Computing education research, Seattle, WA, USA.
- Wiedenbeck, S., LaBelle, D., & Kain, V. R. N. (2004). *Factors affecting course outcomes in introductory programming*. Paper presented at the 16th Annual Workshop of the Psychology of Programming Interest Group.
- Wiedenbeck, S., Sun, X., & Chintakovid, T. (2007). *Antecedents to end users? Success in learning to program in an introductory programming course*. Paper presented at the Visual Languages and Human-Centric Computing, New York.
- Wilson, B. C. (2002). A Study of Factors Promoting Success in Computer Science Including Gender Differences. *Computer Science Education*, 12(1-2), 141-164.
- Wilson, B. C., & Shrock, S. (2001). *Contributing to success in an introductory computer science course: a study of twelve factors*. Paper presented at the Proceedings of the thirty-second SIGCSE technical symposium on Computer Science Education, Charlotte, North Carolina, USA.
- Wright, R., Thompson, W., Ganis, G., Newcombe, N., & Kosslyn, S. (2008). Training generalized spatial skills. *Psychonomic Bulletin & Review*, 15(4), 763-771. doi: 10.3758/PBR.15.4.763
- Young, T. M. (2007). *Parametric and Non-Parametric Regression Tree Models of the Strength Properties of Engineered Wood Panels Using Real-time Industrial Data*. Doctor of Philosophy, University of Tennessee Knoxville.
- Yousoof, M., Sapiyan, M., & Kamaluddin, K. (2007). *Reducing Cognitive Load in Learning Computer Programming*. Paper presented at the World Academy of Science, Engineering and Technology.
- Yurdugül, H., & Aşkar, P. (2013). Learning Programming, Problem Solving and Gender: A Longitudinal Study. *Procedia - Social and Behavioral Sciences*, 83(0), 605-610. doi: <http://dx.doi.org/10.1016/j.sbspro.2013.06.115>
- Zimmerman, B. J. (2000). Self-efficacy: An essential motive to learn. *Contemporary Educational Psychology*, 25, 82-91.

Zin, A. M., Idris, S., & Subramaniam, N. K. (2006). *Student modeling for learning of object-oriented programming in e-learning environment*. Paper presented at the The 6th SEAAIR Annual Conference, Langkawi.

EKLER

EK 1: Uzamsal Yönelim Testi Yönerge Ekranları

EK 2: Görsel-Uzamsal Bellek Testi Yönerge Ekranları

EK-3: Programlamaya İlişkin Özyeterlilik Ölçeği

EK-4: Araştırma İçin Etik Komisyon İzni

EK 1: Uzamsal Yönelim Testi Yönerge Ekranları

Uzamsal-Mekansal Yönelim Testi

Bu test nesnelerin uzaydaki yönelimlerini belirleme ve mekandaki yerlerini zihinde canlandırabilme becerinizi ölçmek üzere hazırlanmıştır.

Her defasında ekranda 8 tane nesne resmi göreceksiniz.

Sizden istenen;

- kendinizi nesnelerin merkezinde görmeniz
- istenilen bir nesneye doğru yönelmeniz
- yöneldiğiniz nesneyi tam karşınıza almanızdır.

Bu durumda diğer nesnelerin yerleri değişmeyecek, fakat sizin yeni konunuza göre yönelimleri değişecektir.

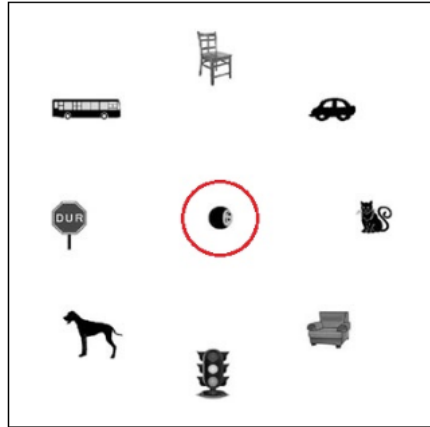
Size verilecek yeni nesnenin değişiklik sonrasındaki yönelimini belirlemeniz ve doğru kutucuğu işaretlemeniz istenmektedir.

Örnek görüntülerle yönergeye devam etmek için tıklayınız.



Ekran 1

Her seferinde size ilk olarak aşağıda soldaki şekil verilecektir.
Ve kendinizi sağdaki gibi merkezdeki figür olarak hayal etmeniz istenecektir.



İleri

Ekran 2

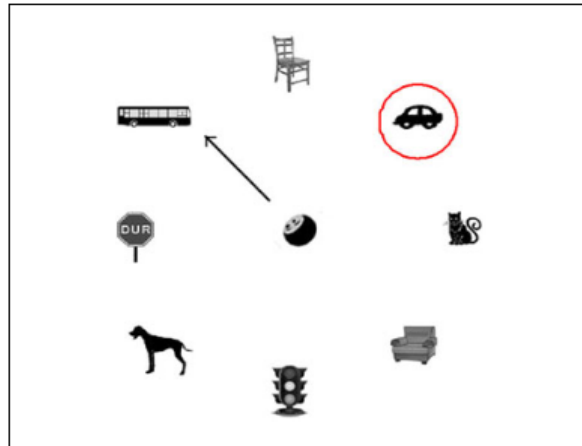
Daha sonra sizden yüzünüzü bir nesneye dönmeniz istenecektir.
Örneğin aşağıdaki şekilde otobüse.



İleri

Ekran 3

Yüzünüz otobüse dönükken sizden başka bir nesnenin konumunu göstermeniz istenecektir.
Örneğin aşağıdaki şekilde arabanın.



İleri

Ekran 4

Her defasında aşağıdaki örnekte olduğu gibi tam karşınıza aldığınız nesnenin (otobüs) yönelimini koruyarak, istenilen diğer nesnenin (araba) yeni yönelimini gösteren kutucuğu işaretlemeniz gerekmektedir.

Size verilen ilk dizilim **Söylenilen ilk nesneyi tam arşınıza aldığınızdaki yeni dizilin** **İstenilen ikinci nesnenin yeni dizilimdeki yerini göstereceğiniz cevap anahtarı.**

Örnek videoyu izlemek için tıklayınız. [▶](#)

Ekran 5

Merkezdeki figür okuduğunuzu hayal edin
Saat yönünde yüzünü trafik lambasına dönün.

Şimdi dur işaretini gösteren kutucuğu işaretleyin.

[Deneme Sorularına Başla](#)

[Tekrar İzle](#)


Ekran 6 (Örnek Çözüm videosu)

EK 2: Görsel-Uzamsal Bellek Testi Yönerge Ekranları


Görsel-Uzamsal Bellek Görevi

Hoşgeldiniz

Bu test sizin görsel-uzamsal bilgileri depolama ve döndürebilme becerinizi ölçmek üzere hazırlanmıştır.
Size 3 setten oluşan sorular sorulacaktır.
Görev esnasında size aşağıdaki 4 farklı sayıdan birinin düz (normal) ya da ayna görüntüsü verilecektir.



Bu görüntüler aşağıdaki örneklerde olduğu gibi farklı yönlerde sunulacaktır.



[İleri](#)

Ekran -1

Örneğin aşağıdaki gibi sayı görüntüsü düz ya da ayna olacak şekilde gelecektir.

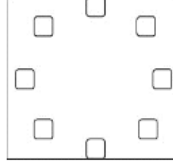


Resmin düz mü yoksa ayna görüntüsü mü olduğuna karar vererek düz ise "D" tuşuna ayna görüntüsü ise "A" tuşuna basınız.

[İleri](#)

Ekran-2

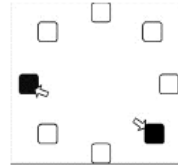
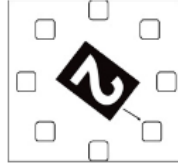
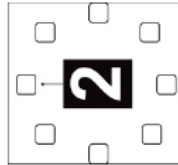
Daha sonra sayıların tepe noktalarının hangi yönde döndürüldüğünü belirleyiniz. Bunun için size 8 farklı yönü işaret eden kutucukların olduğu aşağıdaki cevap anahtarı verilecektir.



İleri

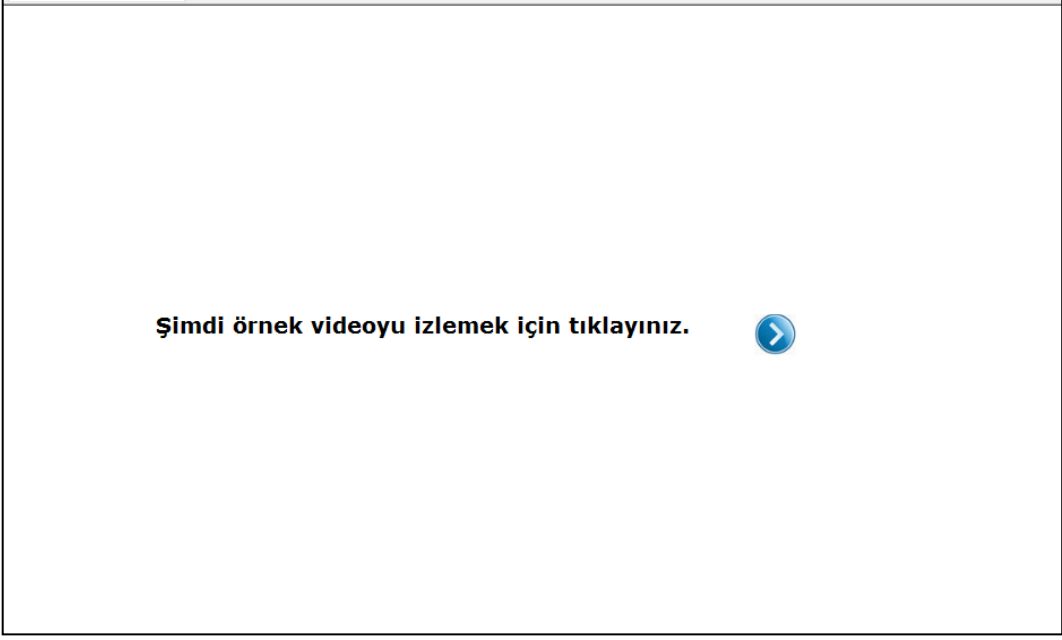
Ekran 3

Sizden aşağıdaki örnekte olduğu gibi sayıların tepe noktalarının yönünü belirleyerek doğru kutucukları işaretlemeniz istenmektedir.

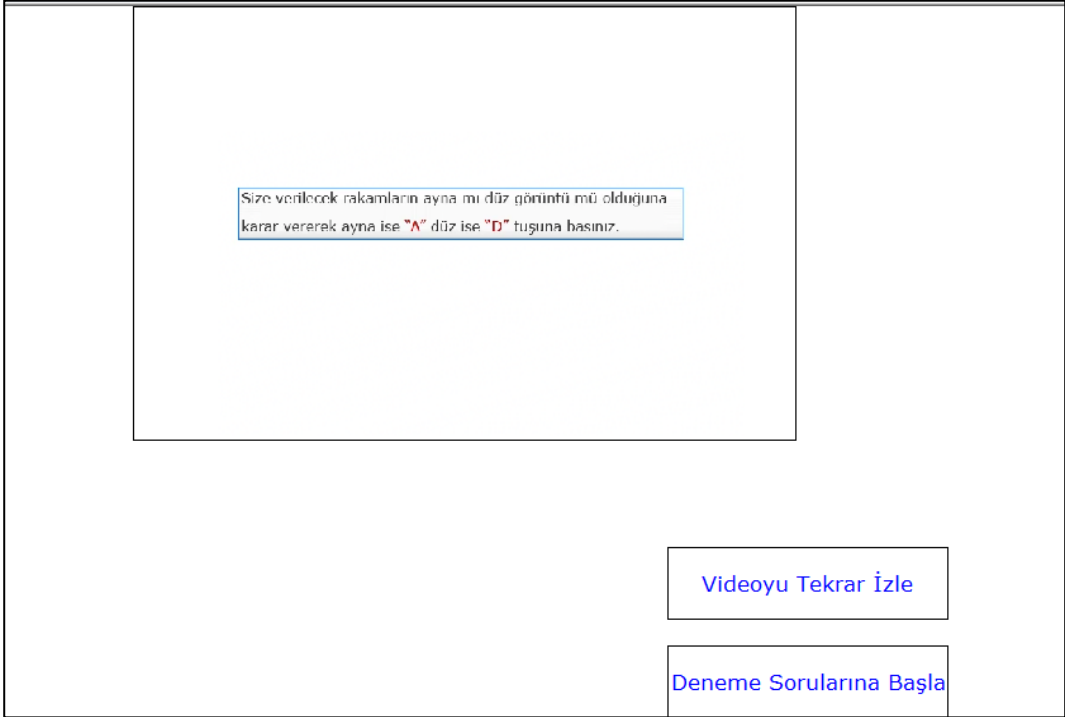


İleri

Ekran 4



Ekran 5



Ekran 6 (Örnek Çözüm videosu)

EK-3 Programlamaya İlişkin Özyeterlilik Ölçeği

Merhaba, bu anket sizin programlama görevlerinde kendinize ilişkin öz yeterlilik algınızı belirlemek üzere hazırlanmıştır. Programlama dili olarak, kullandığınız tüm programlama dillerini (C, C++, Visual Basic, Java vb.) ya da spesifik herhangi birini esas alabilirsiniz. Araştırmaya olan katkınızdan dolayı şimdiden teşekkür ederiz.

Arş. Gör. Sacide Güzin Mazman

Prof. Dr. Arif Altun

Hacettepe Üniversitesi

Bilgisayar ve Öğretim Teknolojileri Eğitimi Bölümü

Cinsiyetiniz: Erkek Kadın

Yaşınız: 18-25 26-35 35-41 41+

Sınıf: 1. 2. 3. 4. Lisansüstü

Bölüm:

Kaç yıldır program yazıyorsunuz?

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


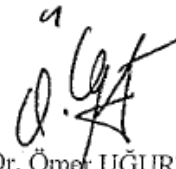
Hangi programlama dillerini kullanabiliyorsunuz?

Ek-3 Devam

Aşağıdaki programlamaya ilişkin verilen görevleri yaparken kendinize olan güveninizi 1 ile 7 arasında derecelendirerek belirtiniz. (“1= Hiç Güvenmiyorum”, “2=Genellikle Güvenmiyorum”, “3= Biraz Güveniyorum”, “ 4= %50/50” “5= Oldukça Güveniyorum”, “6=Genellikle Güveniyorum”, “7=Tamamen Güveniyorum”)

	Hiç güvenmiyorum	Genellikle güvenmiyorum	Biraz güveniyorum	50/50	Oldukça güveniyorum	Genellikle güveniyorum	Tamamen güveniyorum
	1	2	3	4	5	6	7
1. "Merhaba Dünya" mesajının görüntülenebileceği bir program yazabilirim..							
2. Üç sayının ortalamasını hesaplayan bir program yazabilirim.							
3. Verilen herhangi bir sayı dizisinin ortalamasını hesaplayan bir program yazabilirim..							
4. İstenilenler açıkça tanımlandığında bir problemin çözümüne yönelik oldukça karmaşık ve uzun bir program yazabilirim.							
5. Yazacağım bir programı modüler bir biçimde organize edip tasarlayabilirim							
6. Yazdığım uzun ve karmaşık bir programdaki tüm hataları ayıklayabilir ve çalışabilir hale getirebilirim.							
7. Uzun, karmaşık ve birden fazla dosya gerektiren bir programı kavrayabilirim.							
8. Bir programın daha okunabilir ve açık olması için uzun ve karmaşık kısımları yeniden yazabilirim.							
9. Çevrede bir sürü dikkat dağıtıcı olsa bile programa odaklanma yollarını bulabilirim.							

EK-4: Araştırma İçin Etik Komisyon İzni

 HACETTEPE ÜNİVERSİTESİ GENEL SEKRETERLİK	YAZI İŞLERİ MÜDÜRLÜĞÜ 06100 Sıhhiye-Ankara Telefon: 0 (312) 305 1008-1039 • Faks: 0 (312) 310 5552 E-posta: yazimd@hacettepe.edu.tr
Sayı: B.30.2.HAC.0.70.01.00/ 431- 3701	08 AĞ 2012
EĞİTİM FAKÜLTESİ DEKANLIĞINA	
İlgi: 10.08.2012 tarih ve 3972 sayılı yazımız.	
Fakülteniz Bilgisayar ve Öğretim Teknolojileri Eğitimi Bölümü Bilgisayar ve Öğretim Teknolojileri Eğitimi Anabilim Dalı öğretim üyesi Prof.Dr. Arif ALTUN 'un TÜBİTAK Hızlı Destek Programına (1002) başvurabilmek için hazırladığı " Programlama Performansını Etkileyen Faktörlerin Bilişsel Tabanlı Bireysel Farklılıklar Temelinde Modellenmesi " konulu çalışma, Üniversitemiz Senatosu Etik Komisyonunun 29 Ağustos 2012 tarihinde yapmış olduğu toplantıda incelenmiş olup, etik açıdan uygun bulunmuştur.	
Bilgi edinilmesini saygılarımla rica ederim.	
 Prof. Dr. Ömer UĞUR Rektör a. Rektör Yardımcısı	
Ek: Tutanak	

HACETTEPE ÜNİVERSİTESİ ETİK KOMİSYONU
TOPLANTI TUTANAĞI

Toplantı tarihi: 29 Ağustos 2012

Toplantı saati: 14:00

Toplantı yeri: Rektörlük Yönetim Kurulu Toplantı Salonu

Gündemi

Araştırma Anketlerinin değerlendirilmesi

	Sayı	Tarih	Karar
1	4273	07.08.2012	Uygun
2	4412	14.08.2012	Uygun
3	4501	23.08.2012	Uygun
4	4547	24.08.2012	Uygun
5	4759	27.08.2012	Uygun
6			
7			
8			
9			
10			

ASLI GİBİDİR



Aysel TAŞKIN
Yazı İşleri Müdür V.

A. Taşkin

[Handwritten signatures and initials]

ÖZGEÇMİŞ

Kimlik Bilgileri

Adı Soyadı : Sacide Güzin MAZMAN

Doğum Yeri : TOKAT

Medeni Hali : Bekar

E-posta: s.guzin@gmail.com

Eğitim

Yüksek Lisans : Hacettepe Üniversitesi
Bilgisayar ve Öğretim Teknolojileri Öğretmenliği (2007-2009)

Lisans: Ankara Üniversitesi Eğitim Bilimleri Fakültesi
Bilgisayar ve Öğretim Teknolojileri Öğretmenliği (2002-2007)

Lise: Zile Dinçerler 75. Yıl Anadolu Lisesi (1998-2002)

Yabancı Dil

İngilizce

İş Deneyimi

2009 - 2013 Araştırma Görevlisi- Hacettepe Üniversitesi

2007-2009 Ankara Üniversitesi Uzaktan Eğitim Merkezi