

FEN BİLİMLERİ ENSTİTÜSÜ

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI



PROGRAMLAMA ÖĞRENME PERFORMANSINA ETKİ EDEN BİLİŞSEL  
FAKTÖRLERİN BELİRLENMESİ

VE

PROGRAMLAMA EĞİTİMİ İÇİN BİR MODEL ÖNERİSİ

DOKTORA TEZİ

EŞREF SEĞMEN

tarafından

DOKTORA

derecesi şartını sağlamak için hazırlanmıştır.

Kasım 2016

Program: Bilgisayar Mühendisliği

PROGRAMLAMA ÖĞRENME PERFORMANSINA ETKİ EDEN BİLİŞSEL  
FAKTÖRLERİN BELİRLENMESİ

VE

PROGRAMLAMA EĞİTİMİ İÇİN BİR MODEL ÖNERİSİ

DOKTORA TEZİ

EŞREF SEĞMEN

tarafından

OKAN ÜNİVERSİTESİ

Bilgisayar Mühendisliği Anabilim Dalına

Doktora

derecesi şartını sağlamak için sunulmuştur.

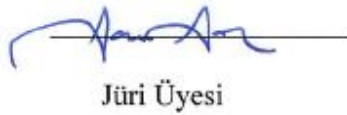
Onaylayan:

  
Danışman

Prof. Dr. Bekir Tevfik Akgün

  
2. Danışman

Prof. Dr. Yavuz Akpınar

  
Jüri Üyesi

Doç. Dr. Sanem Sariel

  
Jüri Üyesi

Yrd. Doç. Dr. Pınar Yıldırım

  
Jüri Üyesi

Yrd. Doç. Dr. Bahri Atay Özgövde

  
Jüri Üyesi

Yrd. Doç. Dr. Engin Toklu

Kasım 2016

Program: Bilgisayar Mühendisliği

## KISA ÖZET

Bu çalışma, bilgisayar programlama öğrenmeye etki eden bilişsel etmenler belirlenerek, bu etmenler temelinde bir programlama öğretim modeli geliştirmeyi amaçlamıştır. Çalışmada önce uzman ve deneyimsiz bilgisayar programcılarının problem çözme davranışları incelenmiş, sonra bilgisayar programlama sürecinde işe koşulan kavramsal ve işlemsel bilgiler literatür ışığında incelenmiş, bilgisayar programlama etkinliğinde altı bilişsel yapının işe koşulduğu gözlenmiştir. Söz konusu bilişsel yapılar şunlardır: “Algoritmik Düşünme”, “Problem Çözme”, “Ardışık ve Döngüsel Düşünme”, “Örüntü Tanıma”, “Soyut Düşünme Soyutlama ve Genelleme”, “Mantıksal Çıkarım ve Akıl Yürütme”. Bu yapıları ölçümleyen programlamayla ilgili bir bilişsel hazır bulunuşluk testi, bu yapılar üzerine kurulu bir bilgisayar programlama öğretim, ölçme ve değerlendirme seti geliştirilmiştir. Bu çalışma sırasında üretilen öğretim araçları programlama öğrenmeye yeni başlayan üniversite öğrencileriyle sınanmıştır.

Yapılan öğretim çalışması, laboratuvar ortamında, kod bloklarıyla programlamayı temele alan Blockly altyapısı üzerine kodlanarak üretilen araçlarla gerçekleştirilmiştir. Çalışma laboratuvar ortamında haftada dört saat olarak beş hafta sürmüş ve 44 öğrenci çalışmanın tümüne katılmıştır. Öğrenciler programlama etkinliğinde işe koşulan altı bilişsel yapıyla ilişkili olarak düzenlenmiş toplam 45 etkinliği geliştirilen düzenek içinde öğretmen yardımıyla tamamlamışlardır. Öğrencilerle yapılan çalışma öncesi ve sonrası öğrencilere programlamayla ilgili bilişsel hazır bulunuşluk testi ve

programlamaya ilişkin öz yeterlilik ölçeđi uygulanmıřtır. Ayrıca öđrencilere programlama başarı testi deney sonrası uygulanmıřtır.

Çalıřma sonunda programlamaya biliřsel hazır bulunuřluk testi ve programlamaya ilişkin öz yeterlilik ölçeđi uygulamalarından öđrencilerin elde ettikleri puanlar istatistiksel olarak analiz edilmiřtir. İlgili testlerin ayırt edicilik özelliklerine sahip olduđu gözlenmiřtir. Hazır bulunuřluk testi alt boyutlarının birbirleriyle iliřkili olduđu gözlenmiřtir. Ayrıca programlama başarı ön ve son testleri arasında son test lehine anlamlı fark tespit edilmiřtir. Bununla birlikte, uygulanan biliřsel hazır bulunuřluk testi alt boyutlarındaki yetkinliklerin tümünün önerilen programlama öđretimi çalıřmasıyla geliřtirilebildiđi gözlenmiřtir. Yapılan kuramsal ve laboratuvar ortamındaki uygulamalı çalıřmalardan ortaya çıkan sonuçlar tezin son bölümünde yorumlanmış ve tartıřılmıřtır.

Anahtar Kavramlar: Bilgisayar Programlama öđretimi, Algoritma, Problem Çözme, Soyut Düşünme, Örüntü Tanıma, Ardışık ve döngüsel düşünme, mantıksal düşünme ve Akıl Yürütme, Biliřsel Yetkinlikler.

## ABSTRACT

This study aimed to find out cognitive factors influencing computer programming, and develop a model for programming instruction on the basis of cognitive factors. The study, first, investigated novice and expert computer programmers' problem solving behaviours. It then reviewed the literature about conceptual and procedural knowledge employed in the process of computer programming. The investigation and the review revealed that six different cognitive constructs are used in computer programming activities; these constructs are “algorithmic thinking”, “sequential and circular thinking”, “pattern recognition”, “problem solving”, “abstraction and generalisation” and “logical thinking and reasoning”. The study then developed a cognitive readiness test for programming to measure those six constructs, proposed and tested a set of tools for computer programming instruction, and constructed a set of measurement and evaluation tool.

The developed tools for programming instruction, based on ideas of programming with ready-code blocks, were tested by a group of university students who just started to learn computer programming. The students (n=44) used the tools and facilities attached to the Blockly platform in a computer laboratory for four hours per week during a scheduled period of five weeks. The students completed a sum of 45 activities about six different cognitive constructs used in computer programming. An instructor provided additional help and guidance to the students during the process of instruction. A cognitive

readiness test for programming and a self-efficacy scale for computer programming were both administered to the students before and after the instructional activities. The statistical analysis of the data collected by these two tests showed that both tests have remarkable discrimination features, and six sub-constructs of the cognitive readiness test for programming meaningfully correlate with each other.

Also, a programming achievement test, administered as a post test, showed that the instructional tools and guidance helped students to learn some aspects of programming. Further statistical analysis demonstrated that the instructional tools and guidance were able to develop the students' all six cognitive constructs to be used in computer programming activities. The results of the theoretical analysis and laboratory experiments with the students are commented and discussed in the final section.

Keywords: Computer Programming Instruction, Algorithmic Thinking, Problem Solving, Abstraction, Pattern Recognition, Sequential and Circular thinking, Logical Thinking and Reasoning, Cognitive Structures.



Sevgili ođlum Emre ve sevgili eřim, Ebru'ya

## TEŞEKKÜR

Yıllar öncesinden İTÜ Kontrol ve Bilgisayar Mühendisliği bölümü öğrencisi olarak dersler aldığım ve yıllar sonra doktora öğrencisi olarak kendisinden tekrar dersler almanın mutluluğunu yaşadığım değerli hocam, çalışmanın bütün aşamalarında tecrübesi ve özverisi ile desteğini esirgemeyen, sayın Prof. Dr. Bekir Tevfik Akgün'e teşekkürlerimi bir borç bilirim.

Doktora çalışması yapmam konusunda beni cesaretlendiren ve teşvik eden. Çalışmanın ortaya çıkması sürecinde bilgi ve birikimleriyle tüm desteklerini esirgemeyen, yol gösteren, akademik altyapı eksiklerimi tamamlayan ve bölümün kaynaklarını çalışmalarım için kullanmama izin veren, on dört yıldır kendisi ile yarı zamanlı öğretim görevlisi olarak Boğaziçi Üniversitesi Bilgisayar ve Öğretim Teknolojileri Eğitimi bölümünde beraber ders vermekten onur duyduğum, hocam sayın Prof. Dr. Yavuz Akpınar'a çok teşekkür ederim.

Görev yaptığım gerek kamu ve gerekse özel sektör kurumlarına, kontrol ve bilgisayar mühendisi olarak programcı, uzman, yönetici, yarı zamanlı öğretim görevlisi ve danışman olarak meslek hayatımın yirmi yedi yılında tecrübeme katkılarından dolayı teşekkür ederim.

Yaşamım boyunca emeklerini ve desteklerini esirgemeyen anneme, rahmetli babama ve kardeşlerime minnet ve sevgilerimi sunarım.

Yaşamımın anlamını oluşturan sevgili eşime ve canım oğluma sevgilerimi sunarım.



# İÇİNDEKİLER

KISA ÖZET .....	i
ABSTRACT .....	iii
TEŞEKKÜR .....	vi
İÇİNDEKİLER .....	vii
TABLolar LİSTESİ .....	xi
ŞEKİLLER LİSTESİ .....	xiii
KISALTMALAR .....	xv
1. GİRİŞ .....	1
1.1 Problem Durumu .....	4
1.2 Araştırmanın Amacı .....	6
1.3 Araştırmanın Önemi .....	9
1.4 Araştırmanın Varsayımları .....	9
1.5 Tanımlar .....	9
1.6 Araştırma Kapsamının Sınırları .....	11
2 KURAMSAL ÇERÇEVE ARAŞTIRMALARI .....	12
2.1 Bilgisayar Programlama .....	12
2.2 Eğitim Bilimi .....	13
2.3 Bilişsel Beceriler .....	14
2.4 Temel Bilgisayar Programlama Eğitime Genel Bakış .....	16

2.5	Bilgisayar Programlama Eğitiminde Karşılaşılan Zorluklar .....	17
3	ÖNERİLEN PROGRAMLAMA EĞİTİMİ YETKİNLİKLERİ VE HİYERARŞİSİ .....	19
3.1	Programlama Yetkinlik Hiyerarşisi Önerisi .....	19
3.2	Algoritmik Düşünme .....	22
3.3	Problem Çözme .....	25
3.4	Ardışık ve Döngüsel Düşünme.....	28
3.5	Örüntü Tanıma.....	30
3.6	Soyut Düşünme, Soyutlama ve Genelleme .....	32
3.7	Mantıksal Çıkarım, Muhakeme, Akıl Yürütme.....	38
4	TEZ ÇALIŞMASININ AŞAMALARI.....	41
4.1.1	Eğitim Metoduna Uygun İzlençe Düzenleme .....	42
4.1.2	Bilişsel Hazır Bulunuşluk Ölçüm Testi .....	44
4.1.3	Programlaya İlişkin Öz Yeterlilik Ölçeği .....	47
4.2	Geliştirilen Eğitim Metodu .....	49
4.3	Metodoloji ve Önerilen Yöntem.....	50
4.3.1	Problem Temelli Programlama Öğretimi.....	52
4.3.2	Eğitim Konuları.....	55
4.3.3	Pekiştirme Soruları ve Seviye İlerleme Kavramı .....	61
4.4	Gerçekleştirilen Bilgisayarlı Eğitim Sistemi.....	66
4.4.1	Merkezi Birim .....	66

4.4.2	Kullanıcı Uygulama Geliştirme Arabirimi .....	68
5	UZMANLARLA YAPILAN GEÇERLEME ETKİNLİKLERİ .....	74
5.1	Uzman Programcılar İle Yapılan Çalışma .....	74
5.1.1	Uzman Programcılar İle Yapılan Çalışmanın Özeti .....	74
5.1.2	Uzmanların Birbirlerinden Ayıran Faktörler .....	75
6	ÖĞRENCİLERLE YAPILAN GEÇERLEME ETKİNLİKLERİ .....	80
6.1	Deneysel Çalışma Sonuçlarının Değerlendirilmesi .....	84
7	BULGULAR VE ÖNERİNİN DOĞRULANMASI .....	87
7.1	Bilişsel Hazır Bulunuşluk Test Madde Analizi Bulguları .....	87
7.2	Geçerleme Çalışması Bulguları .....	92
7.2.1	Bulgu 1: Yetkinlikler Eğitim Sonrası Yetkinlik Artış Oranları ..	94
7.2.2	Bulgu 2: Programlamaya İlişkin Öz Yeterlilik Ölçeği Değişimi	96
7.2.3	Bulgu 3: Değerlendirme Sınavı Başarım Sonuçları .....	97
7.2.4	Bulgu 4: BHB t-test Analizleri Alt Grup – Üst Grup .....	98
7.2.5	Bulgu 5: BHB Ön Test ve Son Test Analizleri .....	110
7.2.6	Bulgu 6: Çalışma Sonuçları Yetkinlikler Arası Korelasyonlar .	115
7.2.7	Eğitim Öncesi ve Sonrası Yetkinlik Değişimleri .....	116
8	SONUÇLAR .....	117
9	KAYNAKLAR .....	123
10	EKLER .....	128

11 ÖZGEÇMİŞ ..... 148



## TABLolar LİSTESİ

Tablo 1.1. Programlamada İŖe KoŖulan BiliŖsel Yapılar ve Temel Programlama İliŖkileri	5
Tablo 3.1. Programlamada İŖe KoŖulan BiliŖsel Yapılar	19
Tablo 4.1. Eđitim Metodu İin Tanımlanan TBÖ İzlencesi	43
Tablo 4.2. BHB Soru-Yetkinlik İliŖkisi Tablosu	44
Tablo 4.3. Öz Yeterlilik Öleđi 1.Sayfa	48
Tablo 4.4. Öz Yeterlilik Öleđi 2. Sayfa	49
Tablo 4.5. Grevlerin Yetkinliklere Gre Deđerlendirilmesi	54
Tablo 4.6. Eđitim Kategorileri	69
Tablo 5.1 Uzman Programcılarla alıŖma Sonuları	76
Tablo 6.1. Deneysel alıŖma Sonularının Deđerlendirme Tabloları-A	85
Tablo 6.2. Deneysel alıŖma Sonularının Deđerlendirme Tabloları-B	85
Tablo 7.1. Eđitim Öncesi BHB Ön Test Sonuları Madde Analizi	90
Tablo 7.2. Eđitim Sonrası BHB Son Test Sonuları Madde Analizi	91
Tablo 7.3. Deneysel alıŖma Sonuları	93

Tablo 7.4. BHB Eğitim Sonrası Yetkinlik Artış Oranları	95
Tablo 7.5. Programlamaya İlişkin Öz Yeterlilik Ölçeği Değişimi	96
Tablo 7.6. Değerlendirme Sınavı İstatiksel Değerleri	98
Tablo 7.7. Deney Sonuçları t-testi Toplu Sonuçlar	99
Tablo 7.8. Deney t-testi Sonuçları	111
Tablo 7.9. Eğitim Sonrası Yetkinlik Artış Oranları	115
Tablo 7.10. Çalışma Sonuçları Yetkinlikler Arası Korelasyonlar	115

## ŞEKİLLER LİSTESİ

Şekil 3.1. Önerilen Yetkinlik Hiyerarşisi	20
Şekil 3.2. Önerilen Programlama Eğitim Metodu ve Yetkinlik İlişkileri	20
Şekil 3.3. Örüntü Yakalama Örneği Fibonacci Hesaplayan Kod Parçası	32
Şekil 3.4. Somut ve Soyut Ortam Ayrımı Programlama İçin Örnek Uyg.	33
Şekil 3.5. Somut ve Soyut Ortam Örnek Uygulama Çıkış Ekranı	34
Şekil 3.6. Somut–Soyut İlişkileri	36
Şekil 4.1 BHB- Bilişsel Hazır Bulunuşluk Örnek Soru 1	45
Şekil 4.2. BHB- Bilişsel Hazır Bulunuşluk Örnek Soru 2	46
Şekil 4.3. BHB- Bilişsel Hazır Bulunuşluk Örnek Soru 3	47
Şekil 4.4. Geliştirilen Öğrenme Platformu ve Soru – Cevap	57
Şekil 4.5. Örnek Pekiştirme Sorusu -1	63
Şekil 4.6. Örnek Pekiştirme Sorusu -2	64
Şekil 4.7. Örnek Pekiştirme Sorusu -2 Çıkış Ekranı	64
Şekil 4.8. Örnek Pekiştirme Sorusu -3	65
Şekil 4.9. Örnek Pekiştirme Sorusu -3 Çıkış Ekranı	65

Şekil 4.10. Merkezi Eğitim Portal Yapısı	67
Şekil 4.11. Geliştirilen Eğitim Aracı İle Örnek Uygulama	70
Şekil 4.12. Geliştirilen Eğitim Aracı Uygulama Bölümleri	71
Şekil 5.1. Uzman Programcı Değerlendirme Çalışması	78
Şekil 7.1. Ölçme ve Değerlendirme Madde Güçlük İndeksi (p)	88
Şekil 7.2. Ölçme ve Değerlendirme Madde Ayırıcılık İndeksi (D)	89
Şekil 7.3. BHB Test Yetkinlik Artışı Ön Test-Son Test	94
Şekil 7.4. BHB Yetkinlikler Eğitim Sonrası Yetkinlik Artış Oranları	95
Şekil 7.5. Programlamaya İlişkin Öz Yeterlilik Ölçeği Madde Değişim	97
Şekil 7.6. Eğitim Sonrası Yetkinlik Artış Oranları	116



## KISALTMALAR

SS: Standart Sapma

SH: Standart Hata

Sd: Serbestlik derecesi

N: Aritmetik Ortalama

p: Anlamlılık düzeyi

TBÖ: Temel Bilgisayar Öğrenimi

BHB: Bilişsel Hazır Bulunuşluk

DS: Eğitim sonu değerlendirme sınavı: Eğitim alan bütün öğrencilere uygulanan ve eğitimin faydasını değerlendiren test

# 1. GİRİŞ

Bilgisayar programlama eğitimi; bilgi ve iletişim teknolojilerinde ortaya koyulan gelişmelerin sonucunda toplumların bu altyapılara hızlı adaptasyonunda önemli rol oynamaya başlamıştır. Halen programlama öğrenimi, orta okullar ve ilgili teknik orta öğretim ve üniversite eğitiminin bir parçası olarak görülmektedir. Ülkemiz yazılım sektörünün ihtiyaç duyduğu programcı kaynağının geliştirilmesi önemli bir konu olmanın yanı sıra, programlama öğretiminin öğrenciler üzerinde yararlı bilişsel etkilerinin olduğu da bilinmektedir. Bu doktora tezi; programlama eğitiminin 21. yüzyılın eğitim sistemi için gerekli ve önemli beceriler olarak görülen soyut düşünme, problem çözme, örüntü tanıma, algoritmik düşünme gibi yetkinlikler üzerindeki etkilerini araştırmak amacı ile gerçekleştirilmiştir. Araştırmanın ana amacı günümüz eğitim sistemlerinin merkezinde olan bu yetkinlik ve becerilerin programlama eğitimi ile nasıl etkilendiği ve programlama eğitiminin bu yetkinliklere etkisini belirlemek için yapılmıştır. Çalışmanın ilk bölümü yetkinlik tespitini, ikinci bölümü geliştirilen yöntem ve programlama eğitimini, son bölümü ise etkilerin ölçülmesini içermektedir.

Bilgisayar programlama bir yaratıcı problem çözme durumu olarak değerlendirilmekte ve bilişsel birçok yetkinliğin altyapısını oluşturan bir beceri olarak karşımıza çıkmaktadır. Programlama eğitimi bilişsel yetkinlikler tarafından etkilenmektedir, bilişsel bilimlerin ilgi alanına giren birçok konuyu ilgilendiren çalışmalar ile ilişkilendirilmelidir ve bu çalışmalardan faydalanılmalıdır. Bilişsel bilim geniş zihinsel aktivitelerle ilgilenen bir bilim dalıdır (Scott ve Nicolson, 1991). Bilgisayar mühendisliğinin bir alt kategorisi olarak programlama eğitimi; bilgisayar bilimleri,

eđitim bilimleri ve psikoloji altyapılarının ortak konusu olarak ele alınıp disiplinler arası niteliđi yönünden incelenmeye yakın zamanda başlanmış bir konudur. Bilgisayar programlama eđitimi ister orta öğretim, isterse üniversite eđitiminin bir parçası olarak düşünölsün, her düzeyde birçok bilişsel ve bilişsel olmayan beceriyi gerektirmektedir. Programlama eđitiminin başarısı birçok bilişsel faktör ile ilişkilendirilmekte ve bu eđitimde başarılı olan öğrencilerin genel başarısının da yüksek olması gerektiđi düşünülmektedir. Bilgisayar programlama eđitiminde başarı birçok bilişsel beceriye bađlıdır. Bilgisayar programlama eđitimi de birçok becerinin öğrencilere kazandırılması amacı ile dünyada bazı ölkelerin eđitim sistemine dâhil edilme yolunda önemli mesafeler kat etmiştir. Temel bilgisayar programlama öğrenimi, eđitim sisteminin altyapısını oluşturan temel bilimler eđitimlerinin yanında yerini almaktadır. Öğrenme çevreye uyum sürecidir (Yeşilyaprak ve diđerleri, 2015). Eđitim sistemi hedeflenen birçok beceriyi öğrencilere öğretirken, bilgisayar programlama eđitimini de kapsamına alarak hem bilgisayar sistemleri programlama için temel programlama eđitimi vermek ve hem de öğrencilere ek yeterlilikler kazandırmayı hedefleyebilir.

Programlama performansını belirleyen ya da bu performans ile ilişkili olan deđişkenleri bulmaya dönük olarak yapılan çalışmalar 1950'lerden bugüne kadar devam etmektedir. Bu konuda yapılan çalışmalar ele aldıkları deđişkenler, ölçme araçları, yöntem ya da bađlam bakımından deđişse de, temelde iyi programcı olmanın nedenlerini ya da bunu etkileyen faktörleri ve bu faktörler arasındaki ilişkiyi bulmaya dönüktür. Ancak, bireysel ve diđer faktörler üzerinde yapılan araştırmalar incelendiğinde, farklı bulgularla karşılaşılmaktadır (Altun, Mazman ve Demir, 2013). Önde gelen üniversiteler (Kent, Van Amsterdam, Berkeley, MIT, Stanford vb.) ve kurumlar

(Microsoft, Google, Oracle, Facebook vb.) ya doğrudan platform geliştirerek ya da destek vererek bilgisayar programlama eğitimine değişik katkılar sağlamaktadırlar. Yapılan çalışmaların bir kısmı, öğrenciye pratik yapma imkânı sağlayarak blok sistemleri ile söz dizim hatası yapmadan uygulama geliştirme mantığını öğretmeye çalışmaktadır (örn., Scratch, Snap, Tynker, Alice). Bazı öne çıkan platformlar, öğrenci ve öğretmenler için ayrı olanaklarıyla, eğitim alan ile eğitim verenler arasında iletişim platformu sağlayarak öğreticiye ve öğrenciye yardımcı bir araç haline gelmektedir. Platformların bazıları bilgisayarınıza bir uygulama yüklemenizi gerektirmesine rağmen, birçok yeni platform sadece web tarayıcı yardımı ile kullanılacak basitliğe ve kullanılabilirliğe sahiptir. Ayrıca tablet bilgisayarlarda hem Android hem de IOS desteği veren bazı platformlar ile Mac, Linux ve Windows başta olmak üzere farklı işletim sistemlerinde çalışan platformlar da mevcuttur.

Bilgisayar destekli öğrenme ortamlarının öğrenme sürecinde öğrenciyi merkeze alması, öğrencilere kendi yetenek, hız, öğrenme stratejisi ve ihtiyaçlarına göre öğrenme fırsatı sunması ve bilişsel farklılıkları dikkate alarak öğrenme sürecini farklı duylara hitap edecek şekilde zenginleştirilmesi gibi pek çok eğitsel faydası bulunmaktadır (Sevim, 2015). Çalışmanın temel amacı; temel bilgisayar eğitimi süreçlerine etki eden zihinsel yapıların incelenmesi ve geliştirilen metot ve programlama eğitim araçları ile programlama eğitimi alan öğrencilerin bahse konu yetkinliklerinde ortaya çıkacak olan değişikliklerin belirlenmesidir.

## 1.1 Problem Durumu

Bilgisayar programlama eğitimi sonunda yüksek başarımlı göstermiş öğrencilerin, diğer öğrencilerden farkları nedir? Bu eğitim hangi bilişsel yetkinlikler ile ne düzeyde ilişkilidir? Bilgisayar programlama eğitimi alan öğrencilerin başarısını hangi faktörler belirlemektedir?

Çalışma konusu temel seviye bilgisayar programlama eğitiminin başarı faktörlerini belirlemek ve eğitimin performansını etkileyen faktörlerin yapısını incelemektir. Bu çerçevede ilgili literatürün bilgisayar mühendisliği bakış açısı ile öğretim teknolojileri altyapılarını inceledikten sonra temel seviye bilgisayar programlama eğitimi için bir modele ulaşılması hedeflenmektedir.

Bize göre “Algoritmik Düşünme, Ardışık ve Döngüsel Düşünme, Örüntü Tanıma, Problem Çözme, Soyut Düşünme Soyutlama ve Genelleme ve Mantıksal Çıkarım ve Akıl Yürütme”, yetkinliklerinin kazandırılmasına da programlama eğitime katkı sağlanması için gereklidir.

Bilgisayar programlama eğitimine etki ettiği düşünülen bu altı bilişsel yetkinlik birçok farklı yetkinlik incelenerek araştırmamıza temel teşkil edecek şekilde tanımlanmıştır. Doktora tezinde sıralanan bilişsel yetkinlikler detayları ile tetkik edilmiş ve programlama eğitimine etkileri araştırılmıştır. Programlama eğitimi ve programlama öğretimi kavramları alanda birbirleri yerine kullanıldığından, bu çalışmada da aynı anlamla birbirleri yerine kullanılmıştır. Programlama öğretimi sürecinde değinilen konular şunlardır: a) Mantıksal yapılar ve Karar verme, b) Soyut yapılar ve değişkenler, c) Hesaplama, çözüm üretme ve problem çözme, d) Diziler, döngüler ve örüntü

oluřturma ve tanıma, e) Birden fazla döngüler (iç ie): Döngüler sürecinde çözümün her adımda alacağı deęerleri takip edebilme f) Algoritma oluřturma ve uçtan uca bir programın alıřması.

Tablo 1.1. Programlamada İře Kořulan Biliřsel Yapılar ve Temel Programlama İliřkileri

	<b>Biliřsel Yetkinlikler</b>	<b>Temel Bilgisayar Programlama</b>
1	Algoritmik Düşünme	Algoritma oluřturma ve uçtan uca programın alıřması ve sonuca ulaşma
2	Ardışık ve Döngüsel Düşünme	Birden fazla döngüler (iç ie) döngüler sürecinde çözümün her adımda alacağı deęerleri takip edebilme
3	Örüntü Tanıma	Diziler, döngüler, örüntü oluřturma ve tanıma
4	Problem Çözme	Hesaplama, çözüm üretme, sonuç ıkarma ve problem çözüme
5	Soyut Düşünme Soyutlama ve Genelleme	Deęişkenler, kodlama ve soyut yapılar
6	Mantıksal ıkarım, Muhakeme ve Akıl Yürütme	Mantıksal yapılar ve karar verme

Biliřsel yetkinlikler temel alınarak Tablo 1.1.'de sıralanan yetkinlikleri ölçümleyen bir biliřsel hazır bulunuřluk testi (BHB) ve bilgisayar programlama eęitim metodu geliřtirilmiřtir. Geliřtirilen eęitim metodu da üniversite öęrencileri üzerine sınanmıř ve sonuçları açıklanmıřtır.

## 1.2 Araştırmanın Amacı

Doktora çalışmasının dört temel amacı vardır: 1-Programlama eğitimine etki eden yetkinlikleri belirlemek, 2- İlgili zihinsel yapı ve yetkinlik seviyelerini ölçümlmek için bir ölçek geliştirmek, 3-Bilgisayar programlama eğitimi amaçlı eğitim metodu, içeriği, sınavları, anketleri ve kod yazmadan programlama yapılabilecek eğitim aracı ve platformu oluşturmak, 4-Üniversite öğrencilerine beş haftalık bir eğitim ile yetkinlikleri ölçümlenen öğrencilere temel bilgisayar programlama eğitimi verilmesi; ardından öğrencinin başarısı ve eğitim sonu yetkinlik seviyesi artışlarının belirlemektir.

Bilişsel hazır bulunuşluk (BHB) testi öğrencilerin seçilen altı bilişsel yetkinliklerinin ölçümü için geliştirilmiştir. Kavramsal ve içerik olarak yetkinlik ölçümü için uzman görüşleri ile yetkinlik ölçekleri oluşturulmuştur. İkinci adım olarak bilişsel yetkinlikler temel alınarak bir eğitim altyapısı tasarlanmış ve gerçekleştirilmiştir. Temel bilgisayar programlama eğitimi için önerdiğimiz eğitim metodunun tüm içeriği ve aşamaları seçilen yetkinlikler temel alınarak oluşturulmuştur. Eğitim metodunun sınanması amacı ile Google Blockly (blockly, 2016) altyapısını kullanan sürükle bırak kolaylıklarını kullanabilen, kod yazma gereksinimi olmadan program yazılabilen bir programlama eğitim aracı geliştirilmiştir. Bu eğitim aracı öğrencilerin algoritma ve senaryolarını yazım hatası yapmadan programlarını geliştirme ve eğitim sırasındaki görevlerini tamamlamaları amacı ile kullanılacaktır. Bu metot ve eğitim altyapısı öğrencilerin öncelikle algoritma oluşturmaya ve problem çözmeye odaklanmalarını sağlamak ve diğer sorunlarla mümkün olduğunca az uğraşmak zorunda kalmalarını sağlamak amacı ile geliştirilmiştir. Yapılan tüm çalışmalar derlenerek bir devlet üniversitesinin Bilgisayar ve Öğretim Teknolojileri Bölümü birinci sınıf öğrencilerinin zorunlu ilk

programlama derslerinde deneysel olarak gerçekleştirilmiştir. Beş haftalık deneysel çalışmalara katılan ve geliştirilen eğitim metodu ile temel programlama öğrenimlerini tamamlayan öğrenciler eğitim öncesi ve sonrası aynı BHB testine tabi tutularak gelişimleri gözlenmiştir.

Bilgisayar programlama eğitimi sürecinde ve başarısında etkili olduğunu düşündüğümüz bilişsel yetkinlikler temel alınarak bir bilişsel hazır bulunuşluk (BHB) testi hazırlanmıştır. Bu test ile eğitim başında ve sonunda öğrencinin yetkinlik düzeyinin ölçülmesi amaçlanmaktadır. Seçilen yetkinlikler temel alınarak bir eğitim metodu geliştirilmiş ve eğitim altyapısı ve görevleri bu yetkinlikler ile aynı hedefe yöneltilmiştir.

Çalışmanın başında sorulan sorular araştırma soruları şunlardır: Belirlenen yetkinlikleri yüksek ve düşük öğrencilerin bilgisayar programlama başarısı arasında anlamlı bir fark var mıdır? İşe koşulan altı yetkinlik eğitim sonunda ne yönde değişim göstermektedir? Bu altı yetkinlik temel alınarak, bir bilgisayar programlama eğitimi metodu, geliştirilebilir mi? Bilgisayar programlama eğitimine etki eden bilişsel faktörler nelerdir? Programlama eğitimi, öğrenme sürecinde hangi bilişsel faktörlere önemli katkılar sağlamaktadır? Bu etki gözlemlenip ölçülebilir mi?

Algoritma geliştirme bir problem çözme durumudur. Algoritma geliştiren bireyler için konuyu şu sorular ile genişletebiliriz.

a) Bilgisayar programlama eğitimine etki eden bilişsel veya bilişsel olmayan faktörleri (değişkenleri) sınıflandırabilir miyiz?



b) Belirlenen bu faktörler göz önünde bulundurularak elektronik öğrenme temelli bir uygulama (e-Öğrenme) altyapısı geliştirebilir miyiz?

c) Başarılı bir programcı için gerekli bilişsel yeterlilikler ne düzeyde tespit edilebilir?

d) Programlama eğitimi tamamlanmış deneklerde ilgili yeterliliklerin gelişim durumu aynı yöntem ile ölçülebilir mi?

e) Bireylerin bilgisayar programlama öğrenimindeki başarısı ile problem çözme yetenekleri ve bilgisayar öz yeterliliği arasında anlamlı bir ilişki var mıdır?

Tez çalışmasında aşağıda sıralanan beş ayrı hipotez oluşturulmuş ve bu hipotezleri destekleyen sonuçlar bulunmuştur.

**Hipotez 1:** Belirlenen altı bilişsel yetkinlik seviyesi ölçümlenebilir.

**Hipotez 2:** Belirlenen altı bilişsel yetkinlik bilgisayar programlamadaki başarı ile doğrudan ilintilidir.

**Hipotez 3:** Altı bilişsel yetkinlik önerilen eğitim metodu ile geliştirilebilir.

**Hipotez 4:** Önerilen eğitim metodu bilgisayar programlama öz yeterliliğini artırır.

**Hipotez 5:** Önerilen bilgisayar programlama eğitimi metodu temel bilgisayar programlamaya anlamlı olarak etki eder.

### 1.3 Araştırmanın Önemi

Bilgisayar mühendisliği günümüzün ve geleceğin en önemli alanlarından biridir. Bilgisayar mühendisliği, belirli bir hedefe ve sisteme dayalı olarak ve işbölümü yapılarak, belirli prensipler, yöntemler ve araçlar kullanılarak bilgisayar programlarının tasarımı, geliştirilmesi, sınanması ve bakımı konularını ele alan mühendislik dalıdır. Bilgisayar programlama eğitimi, bilgisayar bilimleri, eğitim bilimleri ve psikoloji altyapılarının ortak konusu olarak fazla incelenmemiş bir alan olarak karşımıza çıkmaktadır. Bu alanda yapılacak çalışmalar günümüz temel programlama eğitimine metot kazandırma ve bir eğitim aracı ortaya çıkarma potansiyeline sahiptir. Araştırmayı oluşturan; yetkinliklerin araştırılması, oluşturulan ve sınanan “Bilişsel Hazır Bulunuşluk” (BHB) testi, geliştirilen yetkinlik tabanlı eğitim metodu, tasarlanıp geliştirilen eğitim aracı ve yapılan deneysel çalışmalarda elde edilen sonuçlar bilgisayar mühendisliği alanına ve temel programlama öğretimine katkılar sağlayabilir.

### 1.4 Araştırmanın Varsayımları

Beş haftalık çalışmaya katılan öğrencilerin çalışma süresince ve sonunda yapılan görüşmelerde ve testlerde sorulan sorulara samimiyetle cevap verdikleri varsayılmıştır.

### 1.5 Tanımlar

Metoda ilişkin izlençe (Müfredat): Geliştirilen kavramsal altyapı gereğince verilecek temel bilgisayar programlama eğitiminin içeriğinin, süresinin ve çıktılarının belirlenmesi amacı ile çizilen çerçeve klasik programlama eğitimi çerçevesi ile örtüşmektedir.

BHB (Bilişsel Hazır Bulunuşluk Ölçeği): Otuz üç sorudan oluşmaktadır ve bilişsel yetkinliklerin ölçümü için hazırlanmış ve uzman programcılar tarafından değerlendirilmiştir. Eğitimin başında ve sonunda uygulanan test öğrencilerin yetkinlik gelişimi ve programlama eğitimi arasındaki korelasyonu verecektir.

BHB Ön Test: Eğitim öncesi yapılan BHB yetkinlik değerlendirme testi. Deney süreci eğitiminden önce öğrencilerin altı yetkinlik seviyesini ölçümlmek amacı ile gerçekleştirilen test.

BHB Son Test: Eğitim sonrası yapılan BHB yetkinlik değerlendirme testi. Deney süreci sonunda eğitiminden sonra öğrencilerin altı yetkinlik seviyesinin gelişimini ölçümlmek amacı ile gerçekleştirilen test.

Programlamaya ilişkin öz yeterlilik ölçeği: Öğrencilerin programlamada kendi düzeylerini ne derece yeterli gördüklerine dönük algıyı ölçmek amacı ile kullanılmaktadır.

Geliştirilen metodun sunduğu görevler: Eğitim sürecinde izleneye uygun olarak geliştirilen algoritma tasarım ve çözüm senaryolarından oluşan görevler setidir.

Başarı testi: Eğitim alan tüm öğrencilere uygulanan ve eğitimin öğrencilere katkısını gösteren testtir.

Eğitim sonu değerlendirme anketi: Öğrencilerin eğitim, eğitim aracı ve eğitim metodu ile ilgili görüşlerini belirttikleri ankettir.

## 1.6 Arařtırma Kapsamının Sınırları

Deneysel alıřma sre olarak, Bilgisayar ve Eđitim Teknolojileri Blm 1. sınıf đrencilerine CET 102 dersi ierisinde dnemin ilk beř haftasını kapsayacak řekilde sınırlandırılmıřtır.

Alt programlar, dosya iřlemleri, giriř ıkıř ileri dzey ktphane kullanımı eđitimin dıřında bırakılmıřtır. Belirlenen eđitim ieriđini kapsayacak řekilde eđitim haftaları dzenlenmiřtir.

Kırk drt đrenci deneysel alıřmaya dhil edilmiřtir. alıřmaya katılım gnlllk esasına gre olmuřtur. Benzer alıřmalardaki đrenci sayıları da bu dzeydedir (etin, 2012; Cořar,2013; Erdođan, 2005; Gltekin, 2006).

## 2 KURAMSAL ÇERÇEVE ARAŞTIRMALARI

Kavramsal ve kuramsal bir altyapı oluşturmak amacı ile Bilgisayar Bilimleri, Yazılım Mühendisliği, Eğitim Bilimleri ve Bilişsel Bilimleri içeren ilgili kaynaklar incelenmiştir. Bu amaç doğrultusunda temel bilgisayar programlama eğitimi için gerekli ve etkili olan yetkinlikler belirlenmiştir. Yapılan literatür incelemeleri aşağıdaki bölümlerde bilim dallarına göre özetlenmektedir.

### 2.1 Bilgisayar Programlama

Yazılım, genel olarak bir işin program kodları üretilerek yapıldığını belirtir; tanımı, donanım dışında kalan ve kullanıcının/programcının kodlama yaparak istenilen bir işi veya görevin yerine getirilmesi için oluşturduğu programlar/kodlar kümesidir (Çölkesen, 2002). Program kodu bir işin yapılması için algoritmik ifadeyi gösteren programın herhangi bir programlama diline dayanılarak, o dilin deyimler, fonksiyonları ve özellikleri ile elde edilmiş program parçasıdır (Çölkesen, 2002). Programlama basitten zora doğru bir dizi program kodları geliştirmek ve sistematik bir yaklaşım ile olmalıdır. 1978’de Brian Kernigham ve Dennis Ritchie, klasik olmuş kitapları The C Programming Language’da “ Hangi dilde olursa olsun, yazılacak program hep aynıdır.” şeklinde yazmışlardı (Flanagan, 2004). Ülkemizde temel bilgisayar programlama öğretimi büyük çoğunlukla ilgili meslek liseleri, meslek yüksekokulları ve mühendislik bölümlerinde üniversite eğitiminin bir parçası olarak ortaya çıkmıştır. Lise ve ilköğretim programlarında müfredata girmesi konusu günümüzde yaygınlaşmaya başlamıştır. Üniversitelerde bazı bölümlerde zorunlu, bazı alanlarda ise seçimlik dersler

ile öğretilmektedir. Programlama öğrenimi; öğrencilerin uzman programcı olmalarından çok, programlamada edindikleri düşünme teknikleri ile problem çözme odaklı yaklaşım edinmelerinin bir aracı olarak görülmektedir.

## 2.2 Eğitim Bilimi

Temel bilgisayar programlama eğitimi, eğitim bilimleri ve eğitim altyapıları ile doğrudan ilişkilidir. Eğitim bilimleri uzun yıllar içerisinde geliştirilmiş birçok sistemleri barındırmaktadır. Eğitim: Fiziksel uyarımlar sonucu beyinde biyo-kimyasal değişiklikler oluşturma süreci şeklinde tanımlanabilir (Sönmez, 2011). Bu sürecin sonunda eğitim hedeflerine ulaşmak eğitimin performansı ile doğru orantılıdır. Eğitimde önce hedefler belirlenir. Hedefleri sürecin sonunda oluşması istenen durum olarak tanımlanabilir. Bu hedeflere ulaşmak için gerekli olan müfredat oluşturulur. Sönmez'e göre Müfredat beş ögeden oluşmaktadır.

1-Kişide bulunması gereken özellikler,

2-Hedefin göstergesi olan davranışlar,

3-İçerik ve konunun örüntüsü (üniteler),

4-Her davranışı öğrencilerin her birine kazandıracak eğitim durumları,

5-Her davranışı her bir öğrencinin kazanıp kazanmadığını, kazandıysa ne derece kazandığını yoklayan sınav durumları (Sönmez, 2011).

Programlama eğitimi için geliştirilecek bir izlençe de bu ögeleri barındırmalıdır. Geliştirilen eğitim altyapısı hedefleri önceden belirli ve tasarlanan metot ile gerekli becerileri kazandırmak üzere hazırlanmıştır.

### 2.3 Bilişsel Beceriler

Temel bilgisayar programlama eğitimine etki eden bilişsel beceriler üzerine çalışılırken literatürde tanımlanmış ve üzerinde çalışılmış birçok beceri araştırıldı. Aşağıdaki paragraflarda yer alan 19 adet yetkinlik, bilişsel beceri ve beceri türleri öğrenmenin genel konusunu tam olarak incelemekten ziyade programlama öğrenmeye olan ilişkileri açısından ele alındı.

Altun ve arkadaşlarının yaptığı çalışmada veri toplama aracı olarak uzamsal yönelim testi, zihinsel döndürme testi, görsel uzamsal bellek testi, sözel bellek testi, programlamaya ilişkin öz yeterlilik algısı ölçeği kullanılmış ve programlama performansı, akademik not ortalamaları ve matematik başarı puanları ise ilgili öğretim üyelerinden elde edilmiştir (Altun ve ark., 2013). Bu çalışma programlama eğitimine yukarıda sayılan faktörlerin etkisi incelenmiştir; IES\_Uzamsal Yönelim, IES\_Zihinsel Döndürme, IES\_Görsel Uzamsal Bellek, Sözel Bellek, Akademik Ortalama, Matematik Başarısı sayılan faktörler arasındadır.

Altun ve arkadaşları çalışmalarında; programlama performansı olarak programlama dilleri-I dersinden aldıkları yılsonu geçme notları alınarak yüksek grup ve düşük gruplar oluşturmuş; düşük grup ve yüksek grup için en belirleyici değişkenler arasında özyeterlilik, sözel bellek ve matematik başarıları bulunmaktadır. Her iki grupta da

aralarında en fazla benzerlik olan değişkenler arasındaki ilişki sözel bellek, matematik başarısı ve programlamaya ilişkin öz yeterlilik algısı olarak bulunmuştur.

Bilgisayar programlamanın özellikle küçük yaştaki öğrencilerin bilişsel alandaki gelişimlerine katkısı öteden beri çalışılan bir konudur (ör. Papert, 1980; Billings, 1983; Clements ve Gullo, 1984). Bilgisayar programlamanın küçük yaştaki öğrencilere katkısını 7 yaşındaki gruplarla inceleyen Clement ve Gullo (1984), programlama yapan öğrencilerin farklı düşünce (yaratıcılık) yetileri ile üst biliş yetenekleri ve yönlendirme yeteneklerinin programlama yapmayanlardan daha yüksek olduğunu bulgulamışlardır (Akpınar ve Altun,2014).

Solmaz yetkinliklere yönelik çalışmasında yaptığı alıntısında, “Program yazmak bilişsel olarak emek istemekte ve gelişmiş algoritmik düşünme, öğrenme ve problem çözme becerileri gerektirmektedir (Apiola ve Tedre, 2012; Zapušek ve Rugelj, 2013).” Sönmez ayrıca: “Genel olarak programlama için gerekli olan beceriler çeşitli kaynaklarda soyutlama, problem çözme, genelleme, transfer ve eleştirel düşünme gibi başlıklar altında ifade edilmektedir (Gomes ve Mendes, 2007; Gundurao vd., 2010)” alıntısını yapmış ve kavramları çalışmıştır.

Coşar (2013), 7. sınıfta öğrenim gören öğrenciler ile problem temelli öğrenme ortamında bilgisayar programlama çalışmalarının akademik başarı, eleştirel düşünme eğilimi ve bilgisayara yönelik tutuma etkilerini gözlemiştir.



## 2.4 Temel Bilgisayar Programlama Eğitimine Genel Bakış

Bilgisayar programlama, farklı becerilerin aynı anda kullanılmasını gerektiren bir yaratıcı problem çözme ve üretim sürecidir. Bu becerileri kazandırmak için gerekli programlama eğitimi farklı kurum ve kademelerde kimi zaman zorunlu dersler olarak, kimi zaman da isteğe bağlı kurslar şeklinde verilmektedir. Eğitimlerde başarıyı etkileyen olası faktörler arasında öğrencilerin programlamaya karşı tutumu, bilgisayar (ya da bilişim teknolojileri) okuryazarlığı konusunda ön yeterlikleri ve seçilen dil ile amaçlanan becerilerin uyumu sayılabilir. Örneğin kullanılan dilden bağımsız olarak programlama becerisi, mantıksal düşünme ve algoritma oluşturma gibi birçok alanda problem çözmeye yönelik becerileri ve hatta analitik düşünme becerisini de kazandırabilir.

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. Ancak, böylesine önem taşıyan ve toplumda ilgi uyandıran programlama performansına etki eden bireysel ve diğer faktörler üzerinde yapılan araştırmalar incelendiğinde, farklı bulgularla karşılaşmaktayız (Altun, 2013). Yılmaz'a göre (2013), programlama başarısı toplam varyansının %23'ü demografik özellikler, problem çözme ve analitik düşünme becerisi, kişisel öz yeterlilik ve motivasyon değişkenleriyle, %18'iyse öz düzenleyici öğrenme stratejileri ile açıklanabilmektedir.

Günümüzde bilgisayar programlama eğitimine yönelik üniversitelerin başını çektiği çalışmalardan derlenen başlıca örnekler Tablo 1.1. de sıralanmıştır. Sıralamadığımız birçok örnek de mevcuttur. Önde gelen bazı üniversiteler (bakınız: Kent, Van Amsterdam, Berkeley, MIT, Stanford) ve kurumlar (bakınız: Microsoft, Google, Oracle, facebook) ya doğrudan geliştirerek ya da destek vererek bilgisayar programlama eğitimi amaçlı platformlara büyük katkı sağlamaktadırlar. Özellikle üniversitelerin platformların geliştirilmesindeki öncülükleri .

Bilgisayar programlama eğitimi için düşünülmüş oldukça başarılı platformlar mevcuttur. Geliştirilen sistemlerin bazıları öğrenciye kolay pratik yapma imkânı sağlayan blok sistemleri ile kural hatası yapmadan uygulama geliştirme mantığını öğretmeye çalışmaktadır. Bu platformlar öğrenci ve öğretmen bölümlerine sahiptir ve eğitim alan ile eğitim verenler arasında iletişimi sağlar. Güncel teknolojiye sahip platformlar, eğitim sisteminde öğreticiye yardımcı bir araç haline gelmektedir. Platformlar ücretsiz ve ücretli faydalanma imkânlarını sunmaktadır. Bu platformlara örnek olarak Snap (snap.berkeley.edu) gösterilebilir.

## **2.5 Bilgisayar Programlama Eğitiminde Karşılaşılan Zorluklar**

Bilgisayar programlama, farklı becerilerin aynı anda kullanılmasını gerektiren bir problem çözme ve üretim sürecidir. Bu becerileri kazandırmak için gerekli programlama eğitimi ve öğretimi farklı kurum ve kademelerde kimi zaman zorunlu dersler olarak, kimi zaman da isteğe bağlı kurslar şeklinde verilmektedir. Eğitimlerde başarıyı etkileyen olası faktörler arasında öğrencilerin programlamaya karşı tutumu, bilgisayar (ya da bilişim teknolojileri) okuryazarlığı konusunda ön yeterlikleri ve

seçilen dil ile amaçlanan becerilerin uyumu sayılabilir. Bu faktörlerin biri ya da bir kaç öğrencilerin programlama konusundaki başarılarını etkilediği gibi, bu becerilerin farklı alanlarda kullanılmasını da engellemektedir (Ersoy, Madran, ve Gülbahar; 2011). Kullanılan dilden bağımsız olarak programlama becerisi, mantıksal düşünme ve algoritma oluşturma gibi beceriler ve hatta analitik düşünme becerisini de kazandırabilir.

“Program geliştirme eğitiminin başarısı” üzerinde çok sayıda çalışma olmasına rağmen, bu konu yeterli olgunluğa ulaşmamış olup programlama öğretimine dönük yerleşik yaygın yöntemler yeterince geliştirilememiştir. Program geliştirme eğitimi genel olarak bu güne kadar üniversite eğitiminin bir parçası olarak karşımıza çıktı. Uygulama geliştirmenin ilk adımı olmanın dışında içerik ve yaklaşım olarak başlangıç seviyesi programlamanın öğretilmesi ve geniş kitlelere ulaşacak bir sistem ve altyapı oluşturulması için yapılan çalışmalar yeterli değildir.

Temel seviye programlama eğitimi proje ve ödev yapılarak pekiştirilmesi gereken ve laboratuvar çalışması ihtiyacı yüksek olan bir eğitimidir. Öğrenci sayısına bağlı olarak değerlendirme yapacak eğitmenin değerlendirme ve dönüt ve düzeltme için gereken zamanı bulamamasından dolayı gerekli sayıda proje yapılamamaktadır. Bu sorunu gidermede bir bilişim altyapısı işe koşulabilir. Öğrenme sürecinde gerekli olan yoğun insan desteğini bilişim altyapısı ile azaltmak mümkün olacaktır.

### 3 ÖNERİLEN PROGRAMLAMA EĞİTİMİ

#### YETKİNLİKLERİ VE HİYERARŞİSİ

Bilgisayar programlama başarısına etki eden altı temel bilişsel yetkinlik seçilmiştir (Tablo 2.1.). Bu altı yetkinlik yan yana aynı düzeyli ya da hiyerarşik olarak bir düzende görülebilir. Bilgisayar programlama etkinliği algoritma geliştirme etkinliği olarak ele alındığında, algoritmik düşünme yetkinliğinin diğer tüm alt yetkinlikleri kapsadığı söylenebilir.

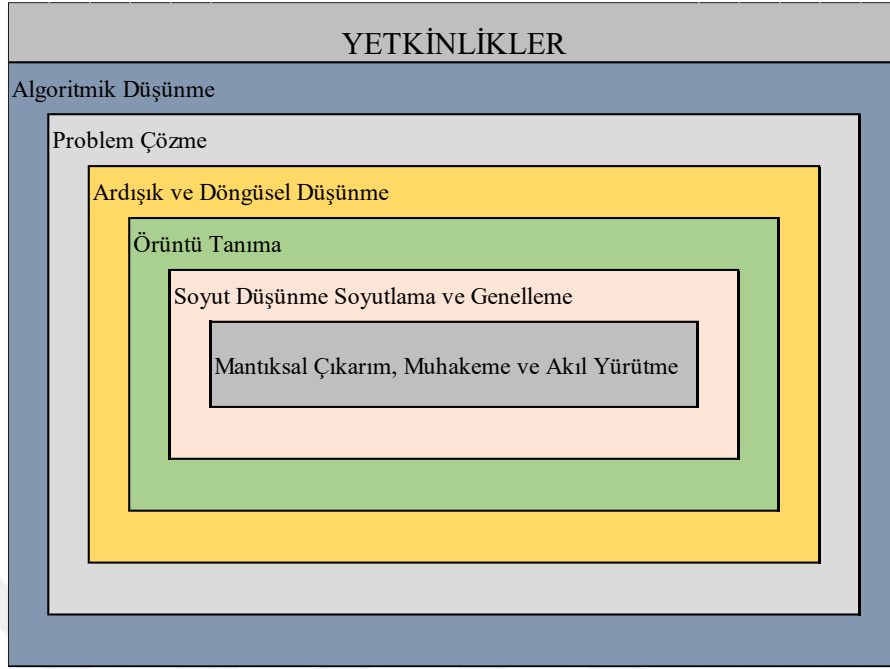
Tablo 3.1. Programlamada İşe Koşulan Bilişsel Yapılar

	YETKİNLİK
1	Algoritmik Düşünme
2	Problem Çözme
3	Ardışık ve Döngüsel Düşünme
4	Örüntü Tanıma
5	Soyut Düşünme Soyutlama ve Genelleme
6	Mantıksal Çıkarım, Muhakeme ve Akıl Yürütme

Aşağıda öneriler ayrıntılı olarak ele alınmaktadır.

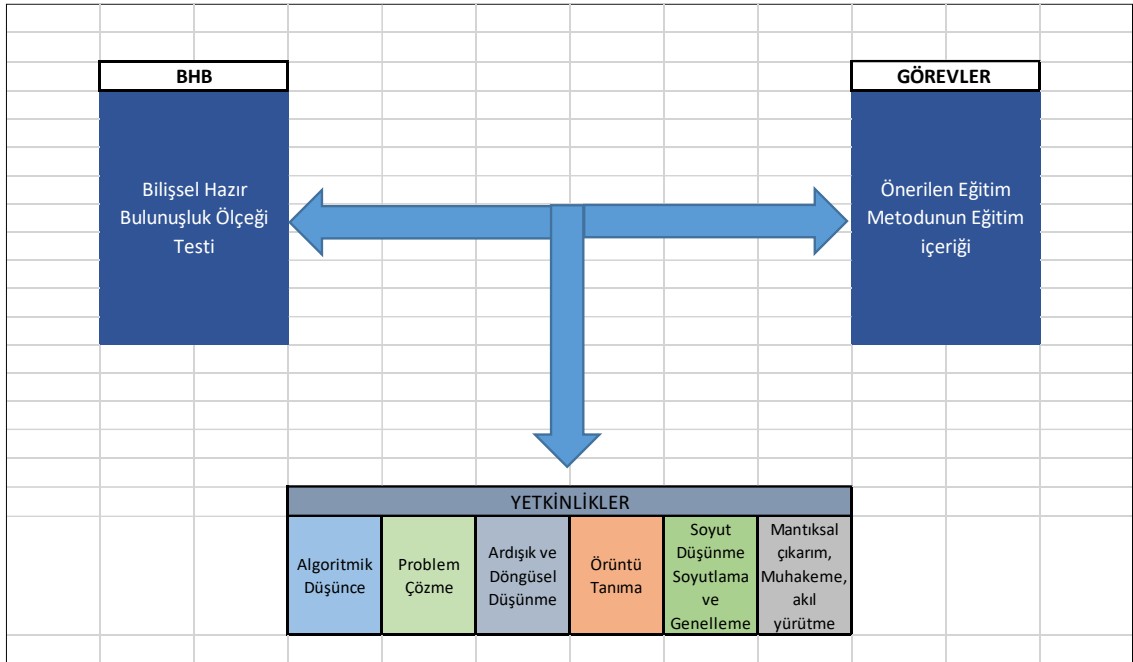
#### 3.1 Programlama Yetkinlik Hiyerarşisi Önerisi

Bilgisayar programlama eğitimi başarısına etki eden yetkinliklerin ve seçilen altı temel yetkinliğin arasında bir hiyerarşik ilişki olduğu önceki kaynaklarda yer almamaktadır. Bu çalışmada 6 yetkinliğin bir hiyerarşik düzen içinde olabileceği gözlenmekle beraber bu konuda doğrulayıcı bir çalışma yapılmamıştır. Aşağıdaki paragraflarda hiyerarşik yapı varsayımı açıklanmaktadır.



Şekil 3.1. Yetkinlik Hiyerarşisi

Özellikle algoritmik düşünme yetkinliğinin tüm alt yetkinlikleri kapsadığını ve bunlardan belirli oranlarda etkilendiği varsayılmaktadır.



Şekil 3.2. Önerilen Programlama Eğitim Metodu ve Yetkinlik İlişkileri

En altta “Mantıksal çıkarım, Muhakeme, Akıl yürütme” tüm yetkinliklerin temelini oluşturmakta ve öğrenmenin birinci basamağını oluşturmakta ve tüm yetkinliklerin temel dayanağını teşkil etmektedir.

İkinci sırada “Soyut Düşünme Soyutlama ve Genelleme” olarak belirlediğimiz bilişsel yetkinlik yer alır. Bu yetkinlik “Mantıksal çıkarım, Muhakeme, Akıl yürütme” yetkinliğini de kapsayan somut dünya ve soyut dünya arasındaki geçişleri ve matematiksel ifadeler ile eşleştirebilme ya da sembollerle temsil edebilmeyi ifade eder.

Üçüncü sıradaki yetkinlik “Örüntü Tanıma” yetkinliğidir. Örüntü tanıma yetkinliği özü itibari verilen bir yapı içindeki elemanlar arası ilişkileri tespit etmeyi niteler; somut ve soyut öğeleri anlamlandırmayı gerektirdiği gibi bir yapıda öğeler arasındaki ilişkilerin altında yatan ilkeleri belirlemeyi gerektirir.

Dördüncü sıradaki “Ardışık ve Döngüsel Düşünme yetkinliği örüntü tanıma yetkinliğini de kapsayan belirli koşullar gerçekleşene kadar tekrarlanan örüntülerin düzenlenmesi işlemidir. Döngülerin belirli bir düzen içinde sıralanması ve amaca uygun şekilde sonuçlar üretilmesi bu tür düşünme yetkinliği seviyesi ile ilintilidir. Ardışık ve döngüsel düşünme yetkinliği örüntü tanıma yetkinliğini temel almaktadır.

Beşinci sırada “Problem Çözme” yetkinliği yer alır. somut ve soyut olarak verilenlerden yola çıkarak, örüntüleri tespit edip, gerekiyorsa döngüleri dikkate alıp bilinmeyene ulaşma sürecidir.

En üst seviyesinde “Algoritmik Düşünme” yetkinliği yer almaktadır. Programlama eğitimi aynı zamanda algoritma tasarımı öğretme olarak da tanımlanabilir.

### 3.2 Algoritmik Düşünme

Algoritmik düşünme bir şekilde oluşmuş algoritmayı anlamak ve kurmakla bağlantılı kabiliyetler havuzudur (Futschek, 2006). Algoritma, belirlenmiş bir hedefi olan bir işi yapmak veya bir problemi çözmek için tanımlanan, açıkça belirlenmiş bir başlangıç ve son durumu olan sıralı sonlu işlemler kümesidir. Algoritma bir sorunu, problemi çözmek ya da belirli bir hedefe ulaşmak için izlenen yola denir. Algoritmik düşünme bir problemin çözümlerini aşamalandırma ve doğru sıraya koyma eylemi olarak da değerlendirilebilir. İşlem dizisinin nasıl yapılacağına karar vermek, süreci aşamalara bölmek, aşamaları doğru sıralamak ve her aşamanın alt süreçlerini belirlemek algoritmik düşünmenin içinde yer alır. Son aşamada tüm süreçleri kapsayan bir kurallar bütünü inşa etmek ve harekete geçmek gelir. Algoritma girdiyi çıktıya çevirmek için uygulanacak komutlar dizisidir (Alpaydın,2011). Amaç girdiye bağlı olarak çıktının belirlenmesidir (Alpaydın, 2010). Aynı girişler aynı şartlar altında sürekli aynı sonuçları oluşturabilecek bir sistemin zihinde tasarlanması işleminin derecesi algoritmik düşünme yeteneğinin seviyesini belirlemektedir. Algoritmik program tasarımı, verilen bir problemin bilgisayar ortamında çözülecek biçimde adım adım ortaya koyulması ve herhangi bir programlama aracı ile kodlanması sürecidir (Çölkesen, 2002). Bu tanımlardan yola çıkarak problemin çözüm senaryosunun adım adım gerçekleştirmek için her adımda ortaya çıkacak muhtemel yapılacaklara karar vererek çözüme ulaşma sürecine de algoritma diyebiliriz.

Algoritmik düşünme bir problemin çözümlerini aşamalandırma ve doğru sıraya koymadır. Bu yapının temel taşları olarak şunlar sıralanabilir: İşlem dizisinin nasıl yapılacağına karar verme, süreci aşamalara bölme, aşamaları doğru sıralama, her

aşamanın alt süreçlerini belirleme ve son aşamada tüm süreçleri kapsayan bir kurallar bütünü inşa etme ve harekete geçme. Burada aynı girişlerin aynı şartlar altında sürekli aynı sonuçları oluşturabileceği bir sistem oluşturmak hedeflenir. Bu sistemi oluşturmanın ve zihinde tasarlanması işleminin derecesi, algoritmik düşünme yeteneğinin seviyesini de göstermektedir.

Gelişmiş anlamda algoritmik düşünme yetkinliği seviyesi, algoritmanın tasarımının doğruluk (correctness) ve güvenilirlik (reliability) temel koşulları altında verimli çalışması, parçalanabilirlik (modularity), sadelik (simplicity), işlevsellik (functionality), genişletilebilirlik (extensibility) ve sağlamlık (robustness) arasındaki dengenin sağlanma derecesi ile doğru orantılı karmaşık bir kavramdır.

Algoritmik düşünme, bir problem veya durum karşında muhtemel sonuçlardan en uygununu elde etmeye dönük çözüm için şartları ve hedefi analiz ederken, bütünü koruyarak adım adım süreçlere ayırma ve bu süreçlerin sebep sonuç ilişkisi içerisinde en uygun çözüme ulaştıracak şekilde tasarlama yöntemidir. Algoritmik düşünme algoritma kavramı anlaşılmadan tam anlaşılabilir. Algoritma, problemi çözmek veya bir görevi tamamlamak için tasarlanan yol olarak da tanımlanabilir. Temel özelliği problemi çözmek veya bir görevi tamamlamak için oluşturulan bir algoritmanın aslında, bir başlangıç durumundan açıkça belirlenmiş bir son duruma giden, sonlu işlemler kümesidir. Yani algoritma belli bir işi veya görevi adım adım ortaya koymaktır (Çölkesen, 2002). Başka bir deyişle algoritma, karşılaşılan sorunu çözmek veya belirli bir hedefe ulaşmak için analizler sonucu tasarlanan yol olarak da tanımlanabilir.



Algoritmik düşünme yetkinliği bir tanımlama ve problemin çözümünde tüm adımları tek tek ele alarak neden sonuç ilişkisi içerisinde tasarlanması işlemlerinin seviyesi ile ilgilidir. Algoritmik düşünme yetkinliğinin seviyesi hiyerarşik olarak ardışık ve döngüsel düşünme ve örüntü tanıma yetkinliğini de ihtiva eder. Geliştirilen bir algoritma, doğası gereği süreçleri adım adım tasarlar ve süreçlerin parametrelerindeki değişikliklerin hangi adımda ne tür değişikliklere uğraması gerektiği veya uğradığı bir sonraki adımın neden-sonuç ilişkisi içerisinde doğru işi yapabilmesini betimlemek zorundadır. Döngü içerisinde bir sonraki adımda yapılması gereken işlemin zihinde canlandırılması en azında birinci ve ikinci adımın ardında (n) inci adımda değişkenler ve süreçler nasıl değişiyor sorusunun yanıtlanması geliştirilen algoritmanın doğru çalışmasının ön şartıdır. Algoritmik düşünme ile ilintili örnek bir etkinlik şu olabilir:

Verilen sayıda taban genişliği olan yıldızlardan oluşan bir piramit oluşturan ve çıkış ekranına yazdıran işlemin algoritmasını tasarlayınız ve blok sistemi ile programını geliştiriniz.

Program çıktısı aşağıdaki gibi olacaktır.

```

*
* * *
* * * * *
* * * * * *

```

Çözüme yönelik algoritma tasarımı: Taban genişliği verilen bir piramit oluşturulması için öncelikle iç içe iki döngünün zihinde canlandırılması gerekmektedir. Döngülerden

biri satırları ikincisi ise yan yana basılacak yıldızları oluşturacaktır. Ardışık ve döngüsel düşünmenin yanında yıldızlar, hangi satırda kaç tane olacağına karar verilmesi örüntü tanıma yetkinliğinin, ne zaman boşluk ne zaman yıldız basılacağı ise mantıksal çıkarım ve karar verme yetkinliklerinin seviyesi ile ilgili olarak aslında birçok yetkinliği birlikte ilgilendirmektedir.

Algoritmik düşünme, bir problemi parçalara ayırmak sureti ile adım adım çözme yetkinliğidir. Algoritma geliştirme temel programlama eğitiminin bir numaralı hedefidir. Karar verme sistemleri, soyut - somut dünya, problem çözme, örüntü tanıma, döngüsel düşünme yetkinlikleri algoritmik düşünmenin alt basamaklarını ve temellerini oluşturmaktadır.

### **3.3 Problem Çözme**

Büyük bir keşif büyük bir problemi çözer ama zaten her problemin çözümünde bir nebze olsun keşif vardır (Polya, 1990). Problem çözme karşılaşılan durum ve problem karşısında bu durumdan kurtulmak ve problemi çözmek amacı ile yapılan strateji ve düşünce ve davranışların bütünü olarak tanımlanır. Karşılaşılan durum daha önceki şemaların kullanılması ile açıklanamadığında çocuğun, daha önce edindiği şemayı değiştirerek yeni bir şema oluşturmasına uyum sağlama denir (Dilci, 2014). Kişinin içinde yaşadığı çevreye uyum sağlama durumunda ortaya çıkan olumsuzluklar ve çatışmalar sorun olarak değerlendirilir ve bu durumdaki birey ve grupların ortaya çıkan olumsuzluklar ve çatışmalarla yapacakları mücadeleye de problem çözme denir.

Bilgisayar programlama ve algoritma üretme bir yaratıcı problem çözme sürecidir. Temel programlama eğitimi bir dizi yetkinliğin yanında problem çözme yetkinliğiyle de yakından irtibatlıdır. Öğrenme problem çözmeye yönelik bir süreçtir (Topses, 2006). Temel programlama öğretimi de problem çözme ve zihinsel süreçleri yakından ilgilendirir. Günümüz eğitim stratejilerinde problem çözme temel yetkinlik olarak göze çarpmaktadır. Öğrencilerde problem çözme becerilerinin geliştirilmesi, öğretim süreçlerinin temel amaçları arasındadır (Topses, 2006). Başka bir açıdan: Bir problemi tanımlama, sunma, alternatif çözümler belirleme ve aralarından bir tanesini seçme ve çözümü planlı bir şekilde uygulamaya koymayı içeren bir öğrenme yöntemidir (Coşar, 2013).

Problem çözme sürecini George Polya dört adımda incelemektedir.

- a. Problemi anlama: Bilinmeyen nedir, veriler nelerdir, koşul nedir.
- b. Plan hazırlama: Veriler ile bilinmeyen arasında bağlantıyı bulmak.
- c. Planı uygulama: Planı yerine getirmek.
- d. Geriye bakma: Bulunan çözümün irdelenmesi (Polya, 1990).

Problem çözme aşamaları (sorunun tanımlanması, sorunun sebeplerinin belirlenmesi, çözüme ilişkin planlamanın yapılması, planların uygulanması ve sonuçların gözden geçirilmesi) göz önünde bulundurulduğunda; programlama, kod geliştirme, algoritma tasarlama eylemlerinin problem çözme kavramını barındırdığı anlaşılır. Bu açıdan baktığımızda bilgisayar programlama ve algoritma üretme bir yaratıcı problem çözme sürecidir. Temel programlama öğretimi sürecinde öğrencinin problem çözme yetkinliklerinin geliştirilmesi göz önünde bulundurulmalıdır. Eğitim öncesi problem

özme yetkinliđinin ölçümlenebilmesi ve eđitim sırasında da geliřtirilmesine yönelik alıřmalar yapılmalıdır.

Problem karřı karřıya kalındıđında ařılması gereken bir durum olarak tanımlandıđında; her program yazımı ve algoritma üretme süreci az veya çok bir problem özme süreci olarak tanımlamak dođru olacaktır. Geliřtirilen eđitim metodunun görevlerinden bir tanesi ile problem özme yetkinliđinin temel programlama eđitimi ile ilintisi řöyle açıklanabilir:

Görev (Problem): Yařları verilen (girilen) üç kardeřin (Ali, Veli, Ayře) yař olarak büyük olanı bulma iřleminin algoritmasını tasarlayınız ve blok sistemi ile programını geliřtiriniz.

Programlamada karřılařtırma altyapısını öğretmeyi amaçlayan bu görev bařlı bařına bir problemdir ve öncelikle zihinde bu problemin zihinsel süreçlerle birlikte özülmesi gerekmektedir. özüm olarak kardeřler ikiřer ikiřer yařları itibari ile karřılatılmalı ve büyük olan tekrar diđerisi ile karřılařtırılmalıdır. Sonuç algoritma tasarlama veya program yazma řeklinde olsa da problem özme yöntemi aynıdır. Geliřtirilen program artık her türlü yařta kardeřlerin en büyüđünü bulacaktır.

Problem özme yetkinliđinin geliřtirilmesi programlama eđitimi açısından önemlidir. Programlama eđitimine bařlamadan bu yetkinliđin ölçülmesi eksiklerin tamamlanması açısından önem arz etmektedir.

### 3.4 Ardışık ve Döngüsel Düşünme

“Sırasal Düzen (Sequential Desing) pek çok farklı doğum bölüklerinin tekrar tekrar test edilmesi yaklaşımıdır” (Onur, 1986). Tekrar eden olguları takip edebilmek ve ardından gelmesi muhtemel olguyu doğru tahmin edebilmek birçok yetiyi bünyesinde barındıran karmaşık bir çözümlenme ve sonuca ulaşma yetkinliği olarak göze çarpmaktadır. Ardışık düşünme yetkinliği sırası ile örüntü tanıma, soyut düşünme ve mantıksal çıkarım ve karar verme yetkinliklerinin etkilediği bir yetkinlik olması gerekir. Ardışık ve döngüsel düşünme özünde örüntü tanımayı ihtiva eder. Kişiler algılama yeteneklerine göre Somut (Concrete) ve Soyut (Abstract) algılayanlar olmak üzere ikiye ayrılırlar. Algıladıkları verileri düzenleme yeteneklerine göre Ardışık (Sequential) ve Random (Random) olmak üzere ikiye ayrılırlar (Ekici, 2002).

Ardışık ve döngüsel düşünme olarak tanımlanan ve temel bilgisayar öğretimi için gerekli ve önem arz eden bu yetkinlik öğrenme ve bilişsel gelişim için önemli bir konu olmaktadır. Öğrenme bireyin sahip olduğu zihinsel yapılar ve bilişsel süreçlerin sonunda gerçekleşir. Zihinsel yapılar duyuşsal kayıt, çalışan bellek ve uzun süreli bellektir (Yılmaz, 2005). Bu süreçler içerisinde Ardışık Zihinsel Düzenleme Yeteneği bilgilerin düzenli ve aşama, aşama öğretilmesini gerektiren yerleştirme yeteneğidir. Bilgilerde mantıki bir sıralamanın ve sistematik bir yapının olması gerekmektedir (Ekici, 2002). Ardışık ve döngüsel işlem başka bir deyiş ile tekrarlı işlem, aynı hedefe yönelik bir işlem grubunu belirli bir sayıda veya belirli bir şart sağlanana kadar tekrar etmesidir. Döngüsel işlem aynı işlemi farklı girişler ve durumlarda farklı sonuç verecek şekilde ardışık olarak önceden belirlenmiş bir sayıda veya belirli bir koşula bağlı olarak sonlandıracak şekilde düzenlenebilir. Ardışık ve döngüsel işlemin hangi döngüde ve

hangi adımlarda nasıl davranacağını zihinde canlandırılması bu yetkinliğin derecesine işaret eder. Örüntü tanıma yetkinliği bu yetkinliğin temel aldığı yetkinlik olarak kabul edilebilir.

Bilgisayar programlama sürecinde programcı geliştirmeye çalıştığı algoritmanın, gereklerini yerine getirmek için problemin çözümünü oluşturacak adımları doğru sıra ile ve doğru kararlar vererek doğru sonucu oluşturacak şekilde gerçekleyebilmesi için problemi analiz edip senaryonun tümünü veya bir bölümünü zihninde canlandırması gerekmektedir. Programcı bu süreçte insan beyninin hem kısa süreli hem de uzun süreli belleğini etkin olarak kullanmaktadır. İnsan, fizyolojisi gereği çevresel uyaranlar duyuşal kayıt olarak kayıt edilmektedir. Uzun süreli bellek, kısa süreli bellekten gelen bilginin uzun süreler için depolanmasında görev alır. Uzun süreli bellek, sinir hücreleri arasındaki elektriksel etkinlik gibi andan ana değışen olaylardan bağımsızdır (Ayhan, 2007). Kısa süreli belleğin hem kapasitesi hem de bilgileri tutabilme süresi kısıtlıdır (Yılmaz, 2005). Kısa süreli bellek algı süreçleri ile çözülecek problemin veya örüntü oluşturma tanıma ve ilişki kurma yetkinliklerinin gücü ve başarımı ile doğrudan ilişkili görülmektedir. Örüntü tanıma yetkinliği bu yetkinliğin temel aldığı yetkinlik olarak kabul edilebilir. Döngüsel işlem veya tekrarlı işlem (iterasyon, İng. iteration), aynı işlem grubunu belirli bir koşul sağlanana kadar tekrar tekrar yapmak için yönlendirir. Döngüsel işlem aynı işlemi farklı girişler ve durumlarda farklı sonuç verecek şekilde ardışık olarak önceden belirlenmiş bir sayıda veya belirli bir koşula bağı olarak sonlandıracak şekilde düzenlenebilir. Ardışık ve Döngüsel işlemin hangi döngüde ve hangi adımlarında nasıl davranacağını zihinde canlandırılması ve soyutlama bu yetkinliğin derecesine işaret eder.

Temel programlama eğitimi örnek görevleri ile ardışık ve döngüsel düşünme yetkinliğine açıklayıcı bir örnek şu olabilir: Verilen (girilen) BÖLÜNEN ve BÖLEN gibi iki sayının sadece toplama ve çıkarma işlemleri yaparak kalanlı bölme yapan ve çıkış ekranına yazdıran işlemin algoritmasını tasarlayınız.

Öncelikle problem zihinde tanımlanır ve çözüm aşamalandırılır. Örüntüler tanımlanır. BÖLEN, BÖLÜNEN'den çıkarılır, kalan bölenden büyük mü? Kontrolü yapılır. Eğer büyükse işlem tekrar edilir. Kalan bölenden küçük olana kadar işleme devam edilir. Kalan BÖLEN'den küçük olduğunda kaç defa çıkarma işlemi yapıldı ise BÖLÜM'ü gösterecektir. Böylece KALAN elde edilir.

Döngüsel düşüncenin zihinde oluşturulması ve ne kadar döndürüleceği ve her döngüde BÖLÜNEN, BÖLEN, KALAN ve kaç defa döndürüldüğü gibi değerlerin değişiminin dikkate alınarak soyut olarak gerçekleştirilmesi problemin çözümünü oluşturmaktadır. Örüntü tanıma yetkinliği bu yetkinliğin temel aldığı yetkinlik olarak kabul edilebilir.

### 3.5 Örüntü Tanıma

Biliş terimi dikkat, örüntü tanıma, duyuşsal kayıtlar, bilişsel nörobilim ve bellek gibi pek çok zihinsel süreçleri ve yapıları içerir (Koçak, 2015) Temel bilgisayar eğitimi açısından bakıldığında, bilişsel gelişimin önemli bir parçası olan örüntü tanıma yetkinliğinin problem çözme yetkinliğinin hiyerarşik olarak üzerine bulunması gereklidir. Örüntü tanıma özünde bir problem çözme faaliyeti olarak tanımlanabilir. Bilişsel gelişimin amacı, soyut şekilde akıl yürütme, varsayımlar, durumlar hakkında

mantıksal düşünme, kuralları karmaşık ve daha yüksek yapıda örgütlemedir (Koçak, 2015). Aynı zamanda örüntü tanımayı da ihtiva etmektedir.

Örüntü Tanıma örüntünün olay veya objelerin belirli bir düzene bağlı olarak birbirini takip etmesi ile ortaya çıkan bir durumdur. Matematikte belirli olan bir kurala göre düzenli bir şekilde tekrarlanan şekil veya sayı dizisine örüntü denmektedir. Bu yetkinlik olaylar arasında neden sonuç bağlantılarını kurabilmek, nesnelere ve kavramlar arasındaki ilişki düzenini anlamak, ayrıca tekrar eden yapıların sistematikliğini, mantığı ve kurallarını belirlemek ve bir sonraki adım için öngörü ve tahmin yapabilme yeteneği olarak adlandırılır.

Temel bilgisayar programlama eğitiminin önemli bir bölümü döngüler ve bunlarla yapılan işlemleri içermektedir. Döngü ve dizilerle yapılan işlemler hesaplamalar belirli bir sayıda döngü sonucunda oluşacak hesaplamalar sonucunda ortaya çıkacak değerler tam anlamıyla örüntü oluşturmaktadır. Belirli bir hesaplama ile bir değer döngünün her durumunda farklı ve bir önceki değeriyle ilişkili olarak hesaplanması ve bu işlemin algoritmasının oluşturulmadan zihinde canlandırılması programlama öğrenen açısından önemli bir katkı teşkil edecektir.

Aşağıda örnek bir görev ve Şekil 3.3.'de bunu karşılayan Java dilinde bir kod parçası verilmiştir.

Örnek: 1'den başlayarak 10 adet Fibonacci sayı dizisini ekrana yazdıran bir program geliştiriniz.



```

int deger_a=1, deger_b=1, deger_c;

    for (int i = 0; i < 10; i++) { //İlk 10 fibonacci sayısını yazdırmaktadır.

        System.out.print(" "+ deger_a);

        deger_c= deger_a+ deger_b;

        deger_a= deger_b;

        deger_b= deger_c;

    }

```

Şekil 3.3. Örüntü Yakalama Yetkinliği Örneği Fibonacci Hesaplayan Kod Parçası

Program geliştirici problemi önce anlayarak bir çözüm geliştirecektir. 10 defa tekrar eden bir örüntüyü tahmin edecek ve zihninde canlandıracaktır. Döngünün her adımında değişkenlerin alacağı değeri anlamak en azından başlangıç ve bitiş durumlarını tahmin etmek durumundadır. Bu yetkinlik programlama eğitimi başında ölçümlenmeli ve eğitim süreci bu yetkinliği geliştirecek şekilde oluşturulması gerekmektedir.

### 3.6 Soyut Düşünme, Soyutlama ve Genelleme

Programlama öğrenimi ve program geliştirme sürecinde, geliştirilecek kodun amacına ve problemin senaryosuna göre gerçek dünyada bulunan bir problemi programlama dilinin kodlama altyapısına uygun olarak bilgisayarın (derleyicinin) anlayacağı dizilimle sembolleştirmek gerekir. Bilgisayarda program çalıştırma; klavyeden veya herhangi bir arabirimden giriş işlemlerini yapma, bilgisayar sisteminin içyapısındaki işlemlerle veriler (sonuçlar) üretme ve bunları çıkış arabirimleri ile yaşadığımız somut dünyada anlaşılabilir işaretlere dönüştürmekten ibarettir. Burada gerçek dünyanın değerleri sayılar veya karakter katarları şeklinde giriş arabirimleri aracılığı ile bilgisayar

içine aktarılır. Soyut dünyada geliştirdiğimiz işlemler yapılıır. Çıkış birimleri ile soyut dünyadan somut dünyaya bilgi aktarımı yapılır. Dikkat edilirse soyut ortamda (zihinde) geliştirilen tüm kodlar, bilgisayar içinde (Bellek, CPU ve diğer unsurlar) yürütülür ve veriler işlenir. Kod üretme eylemi soyut ortamda gerçekleşecek işlemlerin öğrenci zihninde canlandırması ile başlar. Bu nedenle soyut düşünme yetkinliğinin programlama öğrenimine önemli bir etkisinin olduğunu düşünmekteyiz. Bu yetkinliğin bilgisayar programlama eğitimine başlamadan ölçümlenmesi ve eğitim süreçlerin bu yetkinliği geliştirecek şekilde tasarlanması gereklidir.

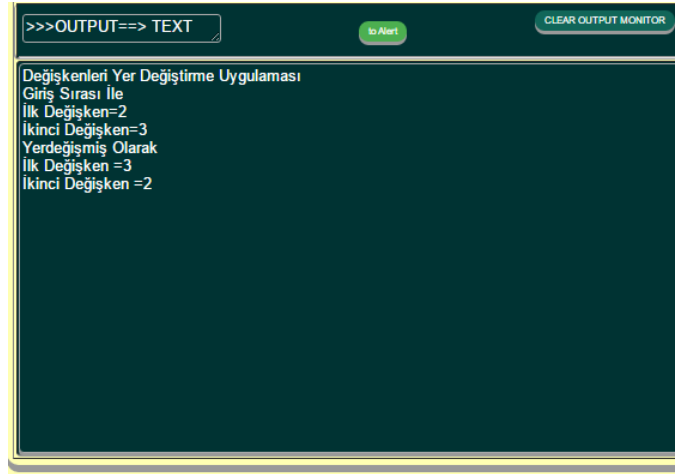
The image shows a Scratch code editor with a JavaScript script. The script is designed to swap the values of two variables, 'ilkDegisken' and 'ikinciDegisken'. It starts by clearing the stage and displaying a title. Then, it prompts the user to enter values for both variables. After displaying the input order, it uses a temporary variable 'temp' to store the value of 'ilkDegisken' before swapping it with 'ikinciDegisken'. Finally, it displays the swapped values.

```

clear | " Değişkenleri Yer Değiştirme Uygulaması "
set ilkDegisken to | prompt for number | with message | " İlk Değişkenin Değerini giriniz "
set ikinciDegisken to | prompt for number | with message | " İkinci Değişkenin Değerini giriniz "
println | " Giriş Sırası İle "
print | " İlk Değişken= "
println | ilkDegisken |
print | " İkinci Değişken= "
println | ikinciDegisken |
set temp to | ilkDegisken |
set ilkDegisken to | ikinciDegisken |
set ikinciDegisken to | temp |
println | " Yerdeğiştirilmiş Olarak "
print | " İlk Değişken = "
println | ilkDegisken |
print | " İkinci Değişken = "
println | ikinciDegisken |

```

Şekil 3.4. Somut ve Soyut Ortam Ayrımı Programlama İçin Örnek Uygulama

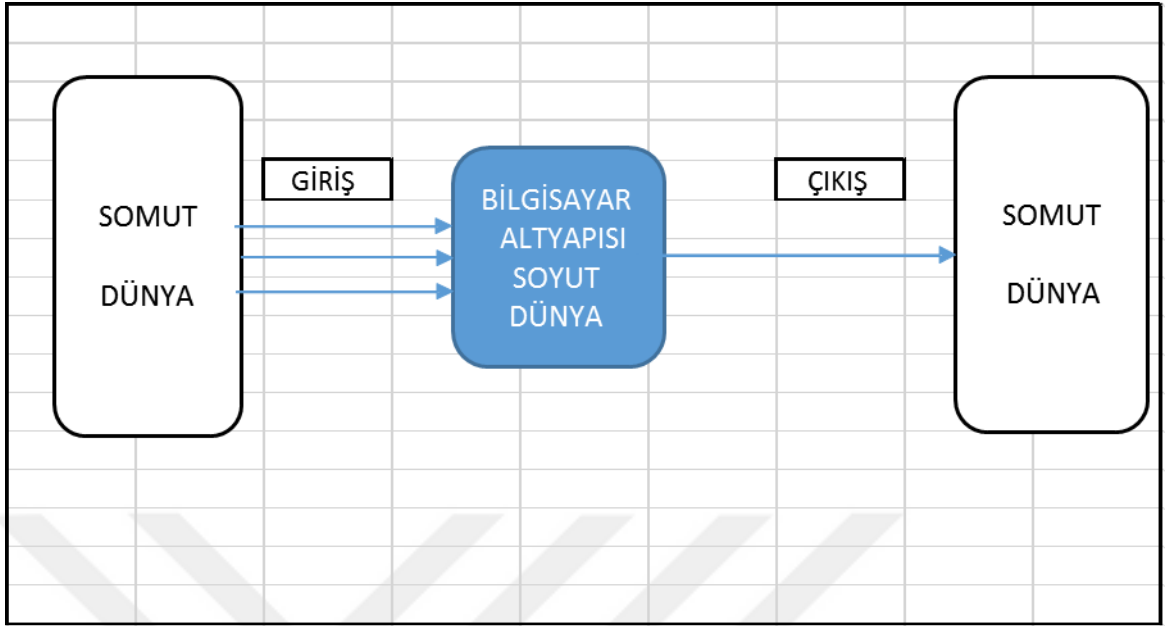


Şekil 3.5. Somut ve Soyut Ortam Örnek Uygulama Çıkış Ekranı

Basit bir örnek ile açıklamak gerekirse şöyle bir görev verilebilir: Klavyeden girilen iki sayısal değer iki değişkene atayınız. Bu iki değişken önce sırası ile çıkış birimine yazdırınız ve daha sonra değişkenlerin içerikleri yer değiştirilerek tekrar çıkış birimine yazdırınız. Tezin bir aşaması olarak geliştirilen programlama eğitim aracı ekranında örnek göreve ilişkin program ve sonuçları Şekil 3.4.’te gösterilmiştir.

Program çalıştırıldığında “ilkDeğişken” değişkenine “2” ve ardında “ikinciDeğişken” değişkenine “3” değeri atandı. Artık soyut bir ortamda bulunan verileri giriş sırasına göre çıkış arabirimimize göndererek somut dünyada erişilebilir ve görülebilir hale getirmiş oldu. Daha sonra bizim için soyut dünya olan bilgisayar belleğindeki değişkenlerin içerikleri yer değiştirilir. Tekrar değişkenler çıkış ekranına yazdırılır. Gerçek dünyada göz ile görülebilir giriş çıkışların arasında soyut ortamda gerçekleşen yer değiştirme işlemi, programcının bilişsel süreçleri ile zihninde canlandırarak gerçekleştirdiği bir tasarımdır. Soyut düşünme ve zihinde canlandırabilme yetkinlikleri programlama öğretimi için önemli özellikler ve yetkinliklerdir.

Program analiz ve geliştirme aşamasında zihinsel süreçler çok önemlidir. Gerekli tüm ilgili planlar ve tasarımlar zihinde oluşturulur ve çözüme gidilmeye çalışılır. Geliştirilen uygulama çalıştırılarak zihinde oluşturulan tasarımın doğru çalışıp çalışmadığı kontrol edilir. Bilgisayar programının çalışması ile elde edilen sonuçlar ve somut program süreçleri aslında soyut ortamı oluşturur. Soyut düşünce ve somut dünya ile ilişkileri kurmak programcı için vazgeçilmez öğelerdir. Bu bağlamda, matematiğe yönelik; inanç, soyut düşünme düzeyi ve problem çözme becerisinin herhangi birindeki artış öğrencilerin başarısını arttırmaktadır (Ağaç, 2013). “Somut işlemler evresindeki çocukların düşünce sistemleri somut gerçekler üzerine kurulmuştur. Mantıkları ile bu gerçekleri yeniden düzenleyebilir, sınıflandırabilir hatta çeşitli değişikliklere gidebilirler. Dünya'nın hemen her yerinde çocukların okula bu evrede başlamaları bir tesadüf değildir. Bu evredeki çocuklar gerçek nesnelere ilgili problemleri zihinlerinden çözebilirler. Aynı zamanda belleklerinde yer alan ezber bir bilgiyle ilişkili sorulara yanıt verebilirler. Ancak ekonomik durum, toplumsal hareketlilik, uluslararası ilişkiler gibi soyut konulardaki sorulara yanıt vermekte güçlük çekerler. Çünkü bu soruların yanıtları soyut düşünme ve koordinasyon gerektirir” (Eripek,1998).



Şekil 3.6. Somut–Soyut İlişkileri

Soyut gerçek dünyaya ve içinde bulunulan ortama aittir. Soyut, nesnelerin ve nesnelere arasındaki ilişkilerin, insan zihninde kavramsallaştırılması ve izdüşümlerinin oluşturulması sonucu ortaya çıkar. Somut düşünme gerçek dünyada şu an ve burada bulunan nesnelere ve arasındaki ilişkiler aracılığı ile yapılan düşünme şeklidir. Soyut kavramı somut varlıklardan hareketle insan zihninde gerçekleşen, kavramlar ve arasındaki ilişkiler ile ifade etmektedir. Soyut düşünce her öğrencide aynı ölçüde gelişmez. Farklı bireylerde farklı seviyelerde gelişim sağlamış olabilir. Gelişim derecesi, derece meydana gelir. İnsanlardaki çoğu değişiklikler bir gecede ansızın meydana gelmez. Soyut bir matematik problemini yanıtlayamayan öğrenciler de bu yeteneğini geliştirebilir. Ancak bu belirli bir süre alacaktır (Eripek, 1998).

Soyut düşünce, fiziksel nesnelere bağımsız olarak tanımlanan kendine has kurallar, özellikler ve ilişkiler ile oluşturulur. Soyut kavramlar belirli bir yaklaşıklıkla bir ya da

birden çok fiziksel nesneyle temsil edilebilir. Somut düşünceden soyut düşünceye geçiş sırasında bir kavramın bilgi içeriğinde azaltma veya indirgeme yapılabilir. Bu indirgeme, gerçek dünyanın karmaşıklığını azaltmak ve bilginin daha rahat işlenmesi ve ilişkilendirilmesi için yapılır. Soyut düşünce ve soyutlamanın bir başka önemli özelliği de kavramların nesnelere uzaklaştırılması işlemi olmasıdır. Soyut düşünce yaşadığımız ortamın ve gerçek dünyanın karmaşıklığını basitleştirme stratejisi olarak kullanılabilir. Soyut düşünme yetkinliği gerçek dünya kavramlarının karmaşıklığını azaltarak ve sınıflandırarak zihinde çözümler üretebilme derecesi ile orantılıdır. Herhangi bir işlem yapmadan çözüm üretmeden önce işlemi zihinde canlandırılmasıdır. Bu yetkinlik öğrencilerde belirli yaşlarda gelişmeye başlar. Soyut İşlemler Evresinde (11-15 Yaşlar) bulunan çocuklar bir olayı değişik yönlerden görebilirler. Bir problemin çözümünde ya da bir karar verme sürecinde soyut düşünebilirler. Problemden yer alan değişkenler arasında ilişki kurup denenceler geliştirebilirler. Geliştirdikleri denenceleri denerler. Böylece problem çözümünde sistemli bir yol izlerler (Eripek, 1998). İkinci bir zorluk ise programlamanın soyut bir yapıya sahip olması ve öğrencinin bu yapının nasıl işlediğini ve sonuçta çözüme nasıl ulaştığını zihinlerinde canlandıramamasıdır. Bunun sonucunda dersin anlaşılması ve kavranabilmesi zorlaşmaktadır (Demir, 2015).

Soyut düşünme ve soyutlama tüm bu bahsi geçen yetkinliklerin içerinde; Karar verme, Mantıksal Çıkarım, Muhakeme ve Akıl Yürütme yetkinliğini kapsayan ve üzerine inşa edilen bir yetkinlik olduğunu düşünmektedir.

### 3.7 Mantıksal Çıkarım, Muhakeme, Akıl Yürütme

Akıl yürütme yeteneği bireyin geçmiş yaşantıları ile ilgili bilgilerini kullanarak problem çözme yeteneği olarak adlandırılmaktadır (Topses, 2006). Akıl yürütme bilinç seviyesi bir işlemdir. Bilinçlilik, gerçeklerle uyumu önde tutan mantıksal düşüncenin egemen olduğu bir durumdur (Dilci, 2014). Bilinç süreçleri sonunda geçmiş tecrübelerden edinilen çıkarımların birleşerek o anki duruma özel bir sonuç oluşturması işlemine Mantıksal çıkarım denir. Mantıksal çıkarımlar ve muhakemeler sonucu karşılaşılan durumlar karşısında kararlar verilir. Karar verme ve çıkarım sistemleri ile ilgili çalışmalarda en büyük sorunlardan biri, gerçek yaşam problemlerinin çeşitli türlerde belirsizlikler içermesi; bilginin kesin, eksiksiz ve objektif olmamasıdır (Teymur, 2008). Karar verme de karşılaşılan durum karşısında gösterilecek potansiyel reaksiyonu da belirlemektedir. Karar verme, zihinsel süreçlerin sonunda, çeşitli alternatifler arasından birinin seçilmesi sürecidir (Demirer, 2012). Olaylara ilişkin olasılık hesapları yaparak iki veya daha fazla seçenek arasından bir seçim yapma yetisi, süreci veya bu amaçla kullanılan yöntemdir (Budak, 2000). Karar verme, Mantıksal Çıkarım, Muhakeme ve Akıl Yürütme bilişsel fonksiyonlar kümesi olarak programlama başarısı için temel bir yetkinliktir. Somutu, örüntüyü ve problemi anlamak için bu temel bilişsel yetkinlikler gereklidir.

Temel bilgisayar programlama göz önünde bulundurularak bir değerlendirme yapılacak olursa: Karar verme, mantıksal çıkarım, muhakeme ve akıl yürütme geliştirilmesi gereken ilk adım ve yetkinlik olarak ortaya çıkmaktadır. Programlamanın en temel unsurları aritmetik ve mantıksal işlemler ve bunların sonuçlarına göre karar verme mekanizmalarının olduğu aşikârdır.

Temel bilgisayar programlama eğitiminin ilk adımları arasında karar verme sistematığı yer alır. Geliştirilecek programlar, çözülecek problemler ve tasarlanacak algoritmaların etkinliği bu yetkinlik ile ilgilidir. Bu nedenle programlama eğitimi sırasında bu yetkinliğin ölçülmesi ve eğitim sırasında geliştirme çalışmaları yapılması önemlidir.

Yönetim faaliyetinin özellikleri incelendiğinde, biri karar verme diğeri uygulama olmak üzere iki yön bulunduğu görülmektedir. Bunlardan birincisi fikir ve yargı oluşturma, ikincisi ise icra terimleriyle ifade edilmektedir (Sağır, 2006). Karar verme konusundaki çalışmalar, karar verme yeterliğinin yaşa bağlı olarak gelişip olgunlaştığını ve karar verme yeterliliği için gereken belli bilişsel becerilerin öğrenilebilir olduğunu ortaya koymuşlardır. Programlama adımlarında en önemli unsurlarından biri karar verme işlemidir. “if” deyiimi, ilgilendirilen şart" ifadesini denetleyerek, ifadenin değerine göre uygun adımlardan birisi ile yola devam etme işlemini gerçekleştirmek için kullanılır. Şekil 3.7. “if” bir kontrol deyimidir. Programcı, şartlı dallanmalar, sayesinde oluşan farklı şartlara karşı tasarladığı farklı adımları yürüten dinamik programlar ile şartlara göre değişik hesaplamalar veya işlemler yapan uygulamalar geliştirir. Şartlı dallanma, temel programlama öğrenimi için en temel kontrol yapısıdır. Program geliştirilirken ortaya çıkacak durumları ön gören ve uygun dallanmalar ile doğru sonuç vermesini sağlayan kodlar geliştirilmesi gereklidir. Bu temel bilgisayar programlama eğitiminde göz önünde bulundurulması gereken bir zorluktur. Adım adım yapılacak tüm işlemler dizisini tasarlamak ve her durumda nasıl davranılacağını hangi adımların atlanıp hangi adımların işleme tabi tutulacağını tespiti Mantıksal çıkarım, Muhakeme, akıl yürütme (Karar verme) yetkinliği ile doğru orantılıdır.



Temel bilişsel fonksiyonlar olarak Mantıksal çıkarım, Muhakeme, akıl yürütme temel bilgisayar programlama eğitimi başarısı için temel ve ilk yetkinlik olarak değerlendirilecektir. İlk seviyenin üzerindeki tüm yetkinlikler bir alt seviye yetkinliği kapsayacağından “Soyut Düşünme Soyutlama ve Genelleme” yetkinliği bu yetkinliği kapsamaktadır.



## 4 TEZ ÇALIŞMASININ AŞAMALARI

Temel programlama öğrenme performansına etki eden bilişsel faktörlerin belirlenmesi ve programlama eğitimi için bir model önerisi kapsamında deneysel eğitim geçerleme sonuçlarının ortaya çıkarılması ve irdelenmesi eylemleri aşağıda sıralanan aşamalarda gerçekleşmiştir. Değerlendirdiğimiz altı yetkinlik temel alınarak;

1-Önerilen eğitim metodunun çerçevesini oluşturan bir izlençe (Müfredat) oluşturulması,

2-Bilişsel hazır bulunuşluk (BHB) testi geliştirilmesi,

3- Programlamaya ilişkin öz yeterlilik ölçeği kullanılması,

4-Eğitim metodu önerisi ve eğitim platformu gerçekleşmesi,

5- Öğretim amaçlı görev ve senaryolarının hazırlanması,

6-Eğitim sonu sınavı,

7-Eğitim sonunda öğrencilerin değerlendirmesi için bir anket geliştirilmesi

yapılmıştır.

Sıralanan aşamaların detayları bu bölümde anlatılmıştır. Geliştirilen ve daha önceki çalışmalardan alınan deneysel çalışma içerikleri ise bir sonraki bölümde nasıl uygulamaya koyulduğu detayları ile anlatılacaktır.

#### 4.1.1 Eğitim Metoduna Uygun İzence Düzenleme

Temel programlama öğretiminin etkin şekilde uygulanabilmesi ve geçерleme deneysel çalışmasına dönüştürülebilmesi amacı ile dünyada kabul görmüş TBÖ Temel Bilgisayar Öğretimi izlenceleri rehber alınmıştır. Bu tür izlenceler birçok eğitim kurumunun ortak olarak tespit ettiği unsurlardan oluşmaktadır. Pek çok programlama dili öğretimi kaynaklarında benzer bir çerçeve mevcuttur. Tasarlanan eğitim metodunun çerçevesini oluşturacak izence çalışması belirlediğimiz yetkinlikler göz önünde bulundurularak derlenmiştir.

TBÖ izence temel kavramları, temel değişkenleri giriş/çıkış işlemleri ile somut ortam değerlerinin giriş komutları yardımı ile soyut ortama aktarılması ve yapılan işlemler sonunda tekrar somut dünyaya çıkış işlemleri vasıtası ile dönüştürülmesini karşılaştırma operatörleri ve mantıksal işlemler ile karar vermeyi ve problem çözme yetkinliklerini içerir. Diziler ve döngüler, örüntü tanıma ve döngüsel düşünmenin temellerini öğretecek şekilde organize oluştur. Son olarak sıralama algoritmaları, algoritmik düşünme yetkinliği ile doğrudan ilgili sırası ile karar verme, soyut düşünme, örüntü tanıma, ardışık ve döngüsel düşünme ve problem çözme, yetkinliklerini kapsayacak şekilde değerlendirildiği görülecektir.

Tablo 4.1. Eğitim Metodu İçin Tanımlanan TBÖ İzlenesi

MODÜL	AÇIKLAMA
1	1.1 Programlamaya giriş - Temel Kavramlar
	1.2 Uygulama Geliştirme Aracı Eğitimi
2	2.1 Temel Giriş/Çıkış Fonksiyonları
	2.2 Soyut düşünme ve Programlama
3	3.1 Değişkenler Bellekte saklama (Atama)
	3.2 Veri Tipleri, Değişkenler ve Sabitler
4	4.1 Operatörler
	4.2 Aritmetik Operatörler
	4.3 Atama Operatörleri
5	5.1 Temel Kütüphane Fonksiyonları
	5.2 Matematiksel Fonksiyonlar
	5.3 Standart Kütüphane Fonksiyonları
	5.4 Karakter Üzerinde İşlem Yapan Fonksiyonlar
6	6.1 Karşılaştırma Deyimleri, Karar verme Kavramı
	6.2 Karşılaştırma Operatörleri ve Mantıksal Operatörler
	6.3 if, if-else Yapısı
	6.4 switch - case Yapısı
7	7.1 Döngüler - İteratif ve döngüsel düşünme. Ardışık İşlemler ve Programlama
	7.2 while Döngüsü (Şartlı Döngü)
	7.3 do ... while Döngüsü
	7.4 for Döngüsü (Belirli sayıda döngü)
	7.5 Sonsuz Döngü (Sonsuz Döngüden belirli bir şart ile çıkma)
	7.6 break Deyimi
	7.7 continue Deyimi
	7.8 İç içe Geçmiş Döngüler Birden fazla boyutta döndürme
8	8.1 Ardışık İşlemler ve Dizi Kavramı
	8.2 Dizilerin Bildirimi
	8.3 Dizilere Başlangıç Değeri Verme
	8.4 Dizileri Yazdırma/Okuma
	8.5 Sıralama (Sorting)
	8.6 Karakter Dizileri (Strings)
	8.7 Çok Boyutlu Dizi Kavramı
9	9.1 Ardışık İşlemler ve Arama/Sıralama kavramları
	9.2 Arama Algoritmaları
	9.3 Sıralama algoritmaları
10	10.1 Fonksiyonlar I (Alt programlar)
	10.2 Fonksiyon Kavramı
	10.3 Parametre ve Argüman Kavramı
	10.4 Fonksiyon Bildirimi
	10.5 Geri Dönüş Değerleri
	10.6 void Fonksiyonlar
	10.7 Fonksiyon Parametreleri

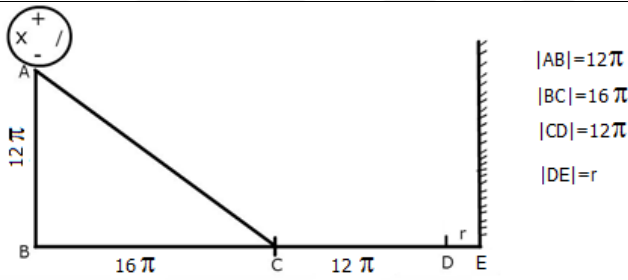


BHB Testi 4 kişilik 4-12 yıl tecrübeli uzman bilgisayar programcılarına incelenmiş ve çözdürülmüştür. Uzmanlar hangi sorunun hangi yetkinliği ölçtüğü ile fikirlerini birbirinden habersiz oluşturmuşlardır. Son olarak bir araya gelen uzmanlar tarafından her soru için ortak yetkinlik etki ağırlık tablosu (Tablo 4.2.) son haline getirilmiştir.

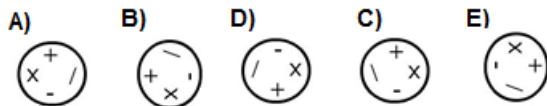
Örnek Soru 1:

Top A noktasından şekildeki gibi serbest bırakıldığında

$r = 2 \text{ cm}$  olduğuna göre



31) C noktasında topun şekli ne olur ?



Şekil 4.1. BHB- Bilişsel Hazır Bulunuşluk Örnek Soru 1

Örnek Soru 2:

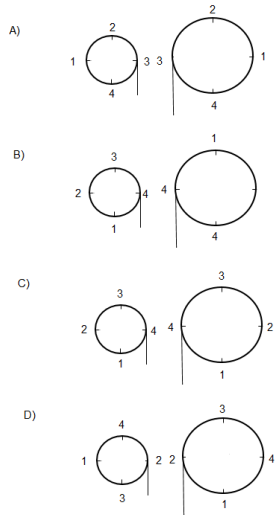
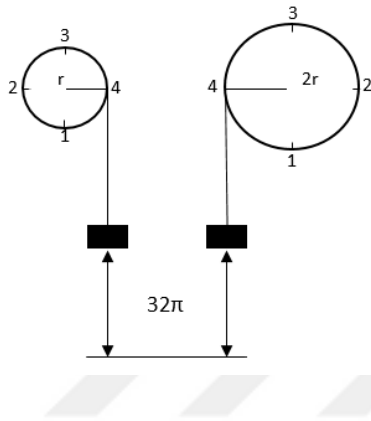
Kütleler serbest aynı anda serbest bırakılıyor.

Kütleler yere dokunduğu anda çemberlerin konumları ne olur ?

1. Çemberin yarıçapı= $r$

2. Çemberin yarıçapı= $2r$

$r=8$  cm



E) Hiçbiri

Şekil 4.2. BHB- Bilişsel Hazır Bulunuşluk Örnek Soru 2

Örnek Soru 3:

Bir kişi 6 km doğuya doğru yürüyor ve sonra güneye dönüp 2 km daha gidiyor. Tekrar doğuya dönüp 2 km daha ilerliyor. Sonra kuzeye dönüp 8 km ilerliyor. Bu durumda başlangıçtaki yerine olan uzaklığı ne kadardır?

A ) 18 B ) 10 C ) 16 D ) 12 E ) 8

Şekil 4.3. BHB- Bilişsel Hazır Bulunuşluk Örnek Soru 3

BHB “Bilişsel Hazır bulunuşluk” testini örnek soruları yukarıda sunulmaktadır. Şekil 4.1., Şekil 4.2. ve Şekil 4.3. Sorular tüm yetkinlikleri ölçmektedir.

#### 4.1.3 Programlamaya İlişkin Öz Yeterlilik Ölçeği

Programlamaya ilişkin öz yeterlilik ölçeği ilk olarak Ramalingam ve Wiedenbeck (1998) tarafından üniversite öğrencilerinin bilgisayar programlama yapmaya ilişkin algılanan öz yeterliliklerini ölçmek üzere geliştirilmiştir. Ölçekte toplamda 32 soru yer almakta olup sorular 7’li Likert tipinde hazırlanmış ve 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 (Altun ve Mazman, 2012).

Bu çalışmada, bilgisayar programlama için Ramalingam ve Wiedenbeck ‘in “Öz Yeterlilik Ölçeği” yerine programlamaya ilişkin öz yeterliliği ölçmek üzere Altun ve Mazman tarafından Türkçeye uyarlanmış şekliyle kullanılmıştır. Tez araştırmalarında kullanılması için gerekli izinler alınmıştır. Böylece, Türkçeye uyarlama ve doğrulama



testleri ve geçerlilik çalışmaları tamamlanmış bir ölçek üzerinde çalışmak mümkün olmuştur. Tablo 4.3 ve 4.4 bu ölçeği göstermektedir.

Tablo 4.3. Öz Yeterlilik Ölçeği 1.Sayfa

### Programlamaya İlişkin Öz Yeterlilik 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.

**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?**

Tablo 4.4. Öz Yeterlilik Ölçeği 2. Sayfa

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.							

## 4.2 Geliştirilen Eğitim Metodu

Çalışmamızın temel amacı programlama eğitimine yeni bir bakış açısı getirmek ve programlama eğitimi başarısına etki eden yetkinlikleri belirleyerek bu yetkinliklerin gelişmesine yardımcı olacak eğitim araçları ve içeriği geliştirerek başarıyı yüksek bir eğitim metodu oluşturmaktır.

Günümüzde yapılan çalışmalar ve konu uzmanlarının yaklaşımı bilgisayar bilimleri ve bilgisayar programlama öğreniminin milli eğitim sisteminde matematik, fen, biyoloji, fizik, kimya ve cebir gibi ana unsurlarından biri olması yönündedir. Algoritmik düşünme eğitimi önümüzdeki yıllarda temel eğitim ve öğretimin önem vermesi gereken başlıca konulardan olacaktır.

### 4.3 Metodoloji ve Önerilen Yöntem

Önerilen öğrenim metodu, bireysel ve grup eğitimine uygun sınıf içi çalışmalarda eğitime yardımcı olan teknolojik bir altyapı şeklinde tarif edilebilir. Temel bilgisayar programlama eğitimi verme amaçlı geliştirilen izleneye uygun ve bilişsel yeterliliğe dayanan eğitim materyalleri ile desteklenmiş (video, yazılı doküman, problemler ve çözümleri, algoritma örnekleri vb.) olarak tasarlanan eğitim sistemi, platform ve bu platformun çalıştırıldığı bir bilgisayar sınıfı teknolojik alt yapıyı oluşturmuştur.

Eğitim aşağıda özetlenen adımlardan oluşmuştur.

- a. Bireysel kullanıcılar (öğrenciler) sisteme kayıt olur.
- b. Bilişsel hazır bulunuşluk testine girerler.
- c. Programlamaya ilişkin öz yeterlilik algısı ölçeği uyarınca değerlendirilirler.

Eğitim sisteminin kullanılmasını ve etkili olması için uygulamanın ve sistemin tüm detaylarını anlatan bir yardım sistemi platformda mevcuttur. Sistemi anlatan bir video kaydı ilk görev olarak verilir. Sistemin ve metodun tanıtılması ilk görev olarak tamamlandıktan eğitim başlar.

Dokuz seviye olarak geliştirilen izlenceye uygun her seviyede Őu b6l6mler vardır.

**Okuma:** Konu anlatımı dokümanı; izlenceye göre teorik altyapıyı oluşturmak için gerekli konu anlatımını içeren detaylı dokümanın çalışılması gerekmektedir.

**İzleme:** Beş dakika süren konu ile ilgili dikkat edilmesi gereken noktalar ve örnek bir problem çözümü içeren videonun izlenmesi gerekir.

**Örnek:** Problem çözüme (Algoritma oluşturma) her seviye için en az beş örnek problem senaryosu oluşturulmuştur. Problemlerin yazılı olarak anlatımı yapılır ve sadece gerekli elemanlar (Programlama elemanları) verilerek çözümün oluşturulması istenecektir. Her problem (senaryo) belirlenmiş sırada bireysel öğrenciye sunulur ve sistem kuralları içerisinde çözüme ulaşması istenir. Çözüm oluşturmak blok altyapısı ile kolayca (sürükle-bırak) gerçekleştirilebilir şekilde tanzim edilmiştir. İstendiği takdirde %25 - %50 -%75 ve %100 çözüm yardımları alınabilir. Verilecek ipuçları ve çözüm kümesi eğitimin aksamasına neden olmaması için parametrik düzenlenebilir.

Pekiştirme sorularını cevaplandırma ve dönüt verme süreci: İçerik uygulamanın anlatımı sırasında örnekler üzerinden detaylandırılır. Sistemde 10 farklı pekiştirme yapısı kullanılır. Öğrenme aşaması problemlerin çözüm kümelerini doğru sonuç oluşturacak şekilde verilen işlemlerin tamamlanmasını gerektirir.

Bütün Seviyelerin bitmesi ile kapanış testinin tamamlanması gereklidir. Testin tamamlanması sonucunda geçer not alan öğrenciler eğitimi tamamlamış olurlar.

Geliştirilen eğitim sisteminin kullanılması, yapılan testler ve yapılan her faaliyet eğitim amaçlı aktivite zaman ve içerik olarak kayıt altına alınır. Tamamlanan her eğitim teorik çalışmanın yanı sıra yazılı, görsel, örnek problemlerin ve pekiştirme sorularının nasıl cevaplandırıldığı (doğru, yanlış veya çeldiriciye aldanmış) ve ne kadar tekrardan sonra doğru cevaba ulaştığı ve hatta ne kadar sürede cevap verdiği ve görevleri tamamladığı ile ilgili süre bilgileri de kayıt altına alınır. Bu istatistiksel bilgiler eğitimin geri beslemesi olarak kullanılır. Öğrenciler kendi seviyelerini ve sistemdeki durumlarını her an takip edebilirler.

**Sistemin Yönetim Modülü:** Önerilen eğitim metodunu uygulamasının esnek olması ve yönetilebilir olmasını planlanmıştır. Henüz geliştirilmesi yapılmayan yönetici modülünden eğitimciler; dokümanları, görsel materyalleri, problemleri ve pekiştirme sorularını oluşturabilir. Oluşturulan eğitim materyali belirlenen bir onay mekanizmasından sonra sistemde kullanılır. Yönetici modülü ile eğitim kurumları ve eğitimciler kendi eğitim sistemlerini oluştururlar ve kendi sınıflarına veya öğrenci gruplarına sistemden modüller seçerek özel bir izlenim oluşturabilirler.

### 4.3.1 Problem Temelli Programlama Öğretimi

Öğretim konusuna giren belirli bir güçlüğü ya da engelin ortaya çıkması ve bunların çözülmesine ilişkin yeterlilik çabalarını geliştirmek, problem çözme yolu ile öğrenmenin temel niteliğini oluşturur (Topses, 2006). Coşar Iglesias'dan şu alıntıyı yapmıştır; “Problem temelli öğrenme (PTÖ), öğrencilerin gerçek yaşam problemlerinden esinlenerek kendi öğrenmeleriyle ilgili temel bilgileri edinirken, problem çözme ve eleştirel düşünme gibi diğer üst seviye becerileri de kazandıran ve

onları geliştiren bir öğrenme yöntemidir (Iglesias, 2002).” (Coşar, 2013). Temel bilgisayar programlama öğretimi için geliştirilen metot, problem temelli yaklaşım sergilemektedir. Geliştirilen yöntem uygulama geliştirme aracı ile verilen görevleri ve problemleri çözmek ve uygulamaları geliştirmek üzerine kurgulanmıştır.

Senaryolar, veri giriş ve çıkış işlemleri, değişkenler, atama, sayı ve metin türleri, kontrol deyimleri, döngüler gibi programlama eğitimi için gerekli olan içeriği sağlayacak şekilde oluşturulmuştur. Çözüm için gerekli blok listesi ve çözümü oluşturan blok ve yerleşim dosyası oluşturulmuştur.

Temel 45 algoritma üzerinden görev (problem-senaryo) tanımlanmıştır.

Her görev şu beş adet dosyaya sahiptir:

1-Görev: Açıklama görev tanımı verilenler istenenler.

2-Araç seti: Bu görevi yapmak için kullanılacak blok türleri. Algoritma seviyesine göre “araçseti” değişebilecek çözüm için gerekli minimum eleman sayısı içerecek şekilde düzenlenir.

3-Çözüm için başlangıç değerler ve ilk gereksinimler dosyası

4-Çözüm: Bloklardan oluşan çözüm xml dosyası. Adım adım çözümü gösterebilir.

5-Teknik yardım sayfası temel programlama ve arabirim yardımlarını da içerir.

Tablo 4.5.’de örnek birkaç görev ve bunlara ait yetkinlik ağırlıklar gösterilmiştir.

Tablo 4.5. Görevlerin Yetkinliklere Göre Değerlendirilmesi

GÖREVLER	Algoritmik Düşünme	Problem Çözme	Ardışık ve Döngüsel Düşünme	Örüntü Tanıma	Soyut Düşünme Soyutlama ve Genelleme	Mantıksal Çıkarım, Muhakeme, Akıl yürütme
Verilen (girilen) iki pozitif integer değerini iki değişken tanımlanarak atama işlemlerinin yapılması ve çıkış ekranına yazdırılması ardından iki değişken değerlerini aritmetik ortalamasını bularak tekrar çıkış ekranına yazdırılması işleminin algoritmasını tasarlayınız ve blok sistemi ile programını geliştiriniz.	3	3	1	1	2	2
1 den 7 e kadar sayıların karelerini çıkış ekranına alt alta yazdıran işleminin algoritmasını tasarlayınız ve blok sistemi ile programını geliştiriniz.	2	1	1	3	3	1
Girilen sayı bir önceki sayıdan küçükse işlemi tamamlayan ve sayıların toplamını hesaplayan ve çıkış ekranına yazdıran işleminin algoritmasını tasarlayınız ve blok sistemi ile programını geliştiriniz.	4	3	4	2	3	3
Verilen(girilen) BOLUNEN ve BOLEN gibi iki sayının sadece toplama ve çıkarma işlemleri yaparak kalanlı bölme yapan ve çıkış ekranına yazdıran işlemin algoritmasını tasarlayınız ve blok sistemi ile programını geliştiriniz.	4	4	4	4	4	4

**Değerlendirme:** Görevin yetkinlik ağırlıkları ile yapılır. Eğitim metodunun altyapısını oluşturan 45 adet görev 4 kişilik 4-12 yıl tecrübeli uzman bilgisayar programcılarına incelenmiştir. Uzmanlar hangi sorunun hangi yetkinliği ölçtüğü ile fikirlerini birbirinden habersiz oluşturmuşlardır. Son olarak bir araya gelen uzmanlar her soru için ortak yetkinlik etki tablosunu (Tablo 4.5) oluşturmuşlardır. Değerlendirme yetkinliğin göreve etkisini 1-4 arası puanlama yapılarak oluşturulmuştur.

- 1-Yetkinliğin geliştirilmesine etkisi yok
- 2- Yetkinliğin geliştirilmesine etkisi kısmen
- 3- Yetkinliğin geliştirilmesine etkisi orta üstü
- 4- Yetkinliğin geliştirilmesine etkisi yüksek

#### **4.3.2 Eğitim Konuları**

Algoritma problemleri 9 ayrı konuyu geliştirecek şekilde düzenlenmeye çalışıldı. Bunlar aşağıda sıralanmaktadır. İlerleyen paragraflarda ise her bir konu için bazı görev örnekleri verilmiştir.

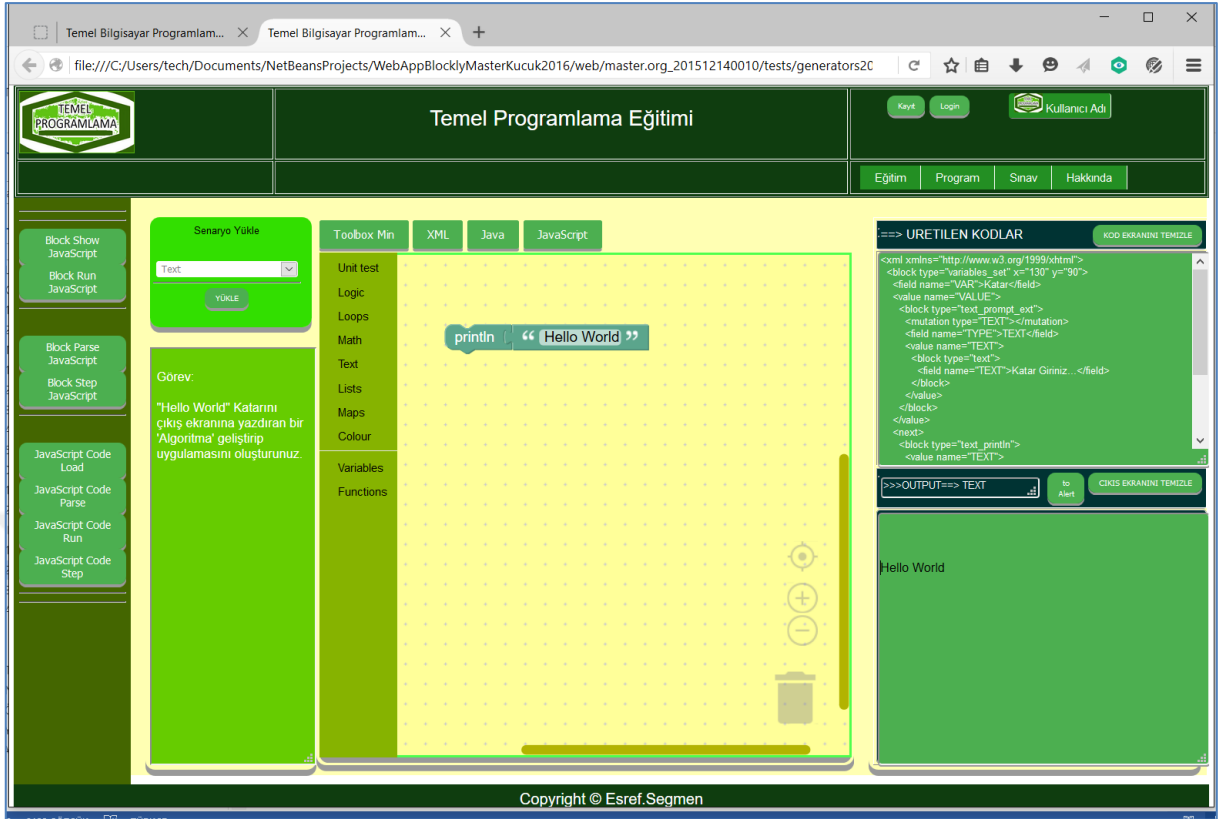
1. Soyut düşünme ve Programlama
2. Değişkenler Bellekte saklama (Atama)
3. Karşılaştırma kavramı



4. İteratif (kurgusal döngüsel) düşünme. Ardışık İşlemler ve Programlama
5. İteratif (kurgusal döngüsel) düşünme. Belirli bir sayıda döndürme
6. İteratif (kurgusal döngüsel) düşünme. Sonsuz döngü kavramı ve şart oluşana kadar döndürme
7. Ardışık İşlemler ve Dizi Kavramı
8. Birden fazla boyutta döndürme ve Çok Boyutlu Dizi Kavramı
9. Ardışık İşlemler ve Arama/Sıralama kavramları

### **Soyut Düşünme Örnekleri**

1. Çıkış ekranına “Hello World” yazdırılması işleminin algoritmasını tasarlayınız ve blok sistemi ile programını geliştiriniz. Otomatik olarak oluşacak kodları inceleyiniz.



Şekil 4.4. Geliştirilen Öğrenme Platformu ve Soru - Cevap

2. Verilen (girilen) bir değerin çıkış ekranına yazdırılması işleminin algoritmasını tasarlayınız ve blok sistemi ile programını geliştiriniz. Otomatik olarak oluşacak kodları inceleyiniz.
3. Verilen (girilen) bir değerin iki katının ve karesinin çıkış ekranına yazdırılması işleminin algoritmasını tasarlayınız ve blok sistemi ile programını geliştiriniz. Otomatik olarak oluşacak kodları inceleyiniz.

### **Soyut Düşünme-Değişkenler ve Bellekte Saklama (Atama) Örnekleri**

1. Verilen (girilen) bir değerin soyut bir değişkene atanması ve saklanması ardından çıkış ekranına yazdırılması işleminin algoritmasını tasarlayınız ve blok sistemi ile programını geliştiriniz. Otomatik olarak oluşacak kodları inceleyiniz.
2. Verilen (girilen) iki pozitif tamsayı değerin iki değişken tanımlanarak atama işlemlerinin yapılması ve çıkış ekranına yazdırılması ardından iki değişken değerlerini yer değiştirime yapılarak tekrar çıkış ekranına yazdırılması işleminin algoritmasını tasarlayınız ve blok sistemi ile programını geliştiriniz. Otomatik olarak oluşacak kodları inceleyiniz.

### **Soyut Düşünme, Değişken Bellekte Saklama ve Karşılaştırma Kavramları Örnekleri**

1. Verilen (girilen) tamsayı değerin pozitif veya negatif sayı olup olmadığına göre sınıflandırmasını yapan ve sonucun çıkış ekranına yazdırılması işleminin algoritmasını tasarlayınız ve blok sistemi ile programını geliştiriniz.
2. Verilen (girilen) iki pozitif tamsayı değeri iki değişken tanımlanarak atama işlemlerinin yapılması ve çıkış ekranına yazdırılması ardından iki değişken değerlerini karşılaştırma yapılarak küçük olan değerin “Küçük Değer :-----” şeklinde çıkış ekranına yazdırılması işleminin algoritmasını tasarlayınız ve blok sistemi ile programını geliştiriniz.

**Soyut Düşünme, Değişken Bellekte Saklama, Karşılaştırma Kavramı, İteratif (Kurgusal Döngüsel) Düşünme, Ardışıl İşlemler Örnekleri**

Soyut düşünme, değişken bellekte saklama, karşılaştırma kavramı, iteratif (kurgusal döngüsel) düşünme, belirli bir sayıda döndürme

1. 1 den 7 e kadar sayıların karelerini döngü ile çıkış ekranına alt alta yazdıran işleminin algoritmasını tasarlayınız ve blok sistemi ile programını geliştiriniz. Otomatik olarak oluşacak kodları inceleyiniz.

Sayı: 1      Karesi: 1

2. “0” girilene kadar sayıların toplamını hesaplayan ve çıkış ekranına yazdıran işleminin algoritmasını tasarlayınız ve blok sistemi ile programını geliştiriniz. Otomatik olarak oluşacak kodları inceleyiniz.

Sayı: 1      Karesi: 1

3. Verilen (girilen) n değeri için  $f(x)=\sum x^2-1$  değerini bulan ve çıkış ekranına yazdıran işlemin algoritmasını tasarlayınız ve blok sistemi ile programını geliştiriniz. Otomatik olarak oluşacak kodları inceleyiniz.

4. Verilen 10 tabanında pozitif tamsayı değerinin ikili sayı sistemine dönüşüm yapan işlemin algoritmasını tasarlayınız ve blok sistemi ile programını geliştiriniz. Otomatik olarak oluşacak kodları inceleyiniz.

5. Verilen (girilen) BOLUNEN ve BOLEN gibi iki sayının sadece toplama ve çıkarma işlemleri yaparak kalanlı bölme yapan ve çıkış ekranına yazdıran işlemin algoritmasını tasarlayınız ve blok sistemi ile programını geliştiriniz. Otomatik olarak oluşacak kodları inceleyiniz.

**Soyut Düşünme, Değişken Bellekte Saklama, Karşılaştırma Kavramı, İteratif (Kurgusal Döngüsel) Düşünme, Ardışıl İşlemler Dizi Kavramı Örnekleri**

1. Rasgele üretilmiş  $n$  elemanlı bir diziyi başka bir diziyeye kopyalayan ve çıkış ekranına yazdıran işleminin algoritmasını tasarlayınız ve blok sistemi ile programını geliştiriniz. Otomatik olarak oluşacak kodları inceleyiniz.

2. Rasgele üretilmiş  $n$  elemanlı bir dizinin elemanlarının toplamını bulan ve sonucu çıkış ekranına yazdıran işleminin algoritmasını tasarlayınız ve blok sistemi ile programını

**Soyut Düşünme, Değişken Bellekte Saklama, Karşılaştırma Kavramı, İteratif (Kurgusal Döngüsel) Düşünme. Ardışıl İşlemler Çok Boyutlu Dizi Kavramı Örnekleri**

1. Değerleri aşağıda verilen  $3 \times 3$  elemanlı iki boyutlu bir dizinin Matrisin sonucu çıkış ekranına yazdıran işleminin algoritmasını tasarlayınız ve blok sistemi ile programını geliştiriniz. Otomatik olarak oluşacak kodları inceleyiniz.

1	2	1
2	4	6
3	5	1

2. Rasgele üretilmiş  $n \times n$  elemanlı iki boyutlu bir dizinin Matrisin Devriğini (Transpose) bulan ve sonucu çıkış ekranına yazdıran işleminin algoritmasını tasarlayınız ve blok sistemi ile programını geliştiriniz. Otomatik olarak oluşacak kodları inceleyiniz",

#### **Arama/Sıralama Algoritmaları Örnekleri**

1. Değerleri verilen bir dizi elemanlarının içerisinde yine değeri verilen bir sayıyı bulan buldu ise dizinin hangi sırasında bulunduğunu (dizinin index'ini) çıkış ekranına yazdıran bulunamadı ise bulunamadı yazdıran işleminin algoritmasını tasarlayınız ve blok sistemi ile programını geliştiriniz. Otomatik olarak oluşacak kodları inceleyiniz.

2. Değerleri verilen bir dizi elemanlarını Kabarcık Sıralama Algoritması (Bubble Sort) kullanarak sıralayan ve sonucu çıkış ekranına yazdıran bulunamadı ise bulunamadı yazdıran işleminin algoritmasını tasarlayınız ve blok sistemi ile programını geliştiriniz. Otomatik olarak oluşacak kodları inceleyiniz.

#### **4.3.3 Pekiştirme Soruları ve Seviye İlerleme Kavramı**

Belirli bir seviye eğitimi alan öğrenci bulunduğu seviyenin materyallerini (doküman, görsel malzeme ve uygulama geliştirme çalışmalarını) tamamladıktan sonra bulunulan

seviyenin amaç ve hedeflerini ne düzeyde yakaladığını belirlenmesi ve sisteme geri beslemede bulunması için pekiştirme soruları üretilmiştir. Oluşturulan her seviye ve her problem için pekiştirme soruları vardır. Seviyenin gerektirdiği bilgi ve deneyim hedeflerine ulaşabilmek için bu pekiştirme sorularına doğru cevap vermek şarttır. Pekiştirme soruları 10 ayrı kategoride değerlendirilebilir:

1. Problemin çözümü için eksik bloğun bulunması,
2. Çözümü verilen Blok yapısının doğru çalışacağı veya hataya eden olacağının tespiti,
3. Problemin çözümü için karışık blok yapısının doğru sıraya koyulması,
4. Problemin çözümü için blok yapısının hangi iki bloğunun yer değiştirmesi gerektiğinin bulunması,
5. Problemin belirli bir çözümü oluşturan blok yapısının olası sonuçlarını tespit etmek,
6. Problemin çözümünü oluşturacak eksik Kod parçasının belirlenmesi,
7. Yanlış olan (hataya neden olacak) kod parçasının tespiti,
8. Problemin çözümü için karışık kod yapısının doğru sıraya koyulması,
9. Problemin çözümünü oluşturan kod parçasının olası sonuçlarını tespit etmek,
10. Eksik kod satırının yazarak eklenmesi ve yazım hatalarına dikkat ederek ekleme.

Her seviye içerdiği problem ile ilgili pekiştirme sorularına doğru cevap vermek sureti ile aşılabılır. Verilen yanlış cevaplar öğrenciyi konuyu anlaması için ipuçlarına yönlendirir. Tüm pekiştirme soruları da doğru cevaplandıktan sonra bir sonraki seviye konuya geçilir.

## Örnek Pekiştirme soruları

**Soru 1** Verilen (girilen) Karakterlerin rakam veya harf olmalarına göre sınıflandırma yapan ve sonuçları çıkış ekranına yazdıran işleminin blok sistemi ile programını geliştirilmiştir. Siyah olarak gösterilen parçanın yerine hangi seçenekte bulunan blok programın doğru sonuç vermesini sağlayacaktır.

The image shows a Scratch code editor with a yellow background. The code is as follows:

```

set karakter to prompt for text with message degeri girin
if
do
else
  print say

```

The 'do' block is blacked out. Below the code are four options:

- a) print karakter
- b) repeat while do
- c) print say
- d) true

Şekil 4.5. Örnek Pekiştirme Sorusu -1



Soru 2: Blok diyagram çalıştığında çıkış ekranı hangi seçenekte olur?

```

printn " SIRALAMA ALGORITMASI "
printn " "
set Dizi to create list with 1, 9, 3, 5
print " Dizi Degerleri: "
printn Dizi
printn " "
count with f from 1 to 10 by 1
do
  count with f from 1 to 10 by 1
  do
    if in list Dizi get # i < in list Dizi get # j
    do
      set Gecici to in list Dizi get # i
      in list Dizi set # i as in list Dizi get # j
      in list Dizi set # j as Gecici
    end if
  end do
end do
print " Siralı Dizi Degerleri: "
printn Dizi
  
```

Şekil 4.6. Örnek Pekiştirme Sorusu -2



Şekil 4.7. Örnek Pekiştirme Sorusu -2 Sonuç Ekranı

Soru 3: Aşağıdaki karışık sırada verilen blokların Girilen bir karakter katarını çıkış ekranına yazdıracak algoritmayı oluşturacak şekilde sıralayınız?

a)

```
println " Girilen deđeri çıkış ekranına yaztıran uygulama "
set 'Giris' to prompt for 'text' with message "Giriniz"
println 'Giris'
```

b)

```
println 'Giris'
println " Girilen deđeri çıkış ekranına yaztıran uygulama "
set 'Giris' to prompt for 'text' with message "Giriniz"
```

c)

```
println 'Giris'
println " Girilen deđeri çıkış ekranına yaztıran uygulama "
```

d)

```
set 'Giris' to
println " Girilen deđeri çıkış ekranına yaztıran uygulama "
```

Şekil 4.8. Örnek Pekiştirme Sorusu -3

```
println 'Giris'
set 'Giris' to prompt for 'text' with message "Giriniz"
println 'Giris'
```

```
>>>OUTPUT=>>> TEXT [to Alert] [CIKIS EKRANINI TEMIZLE]
Girilen deđeri çıkış ekranına yaztıran uygulama
Programming Education
```

Sonuç: Çıkış Ekranı

Girilen Deđer: "Programming Education"

Şekil 4.9. Örnek Pekiştirme Sorusu -3 Çıkış Ekranı

#### 4.4 Gerçekleştirilen Bilgisayarlı Eğitim Sistemi

Metodolojisi geliştirilen eğitim sisteminin teknolojik altyapısı iki ana birimden oluşmaktadır. Bunlar;

- 1- Merkezi Birim,
- 2- Kullanıcı Uygulama Geliştirme Arabirimidir.

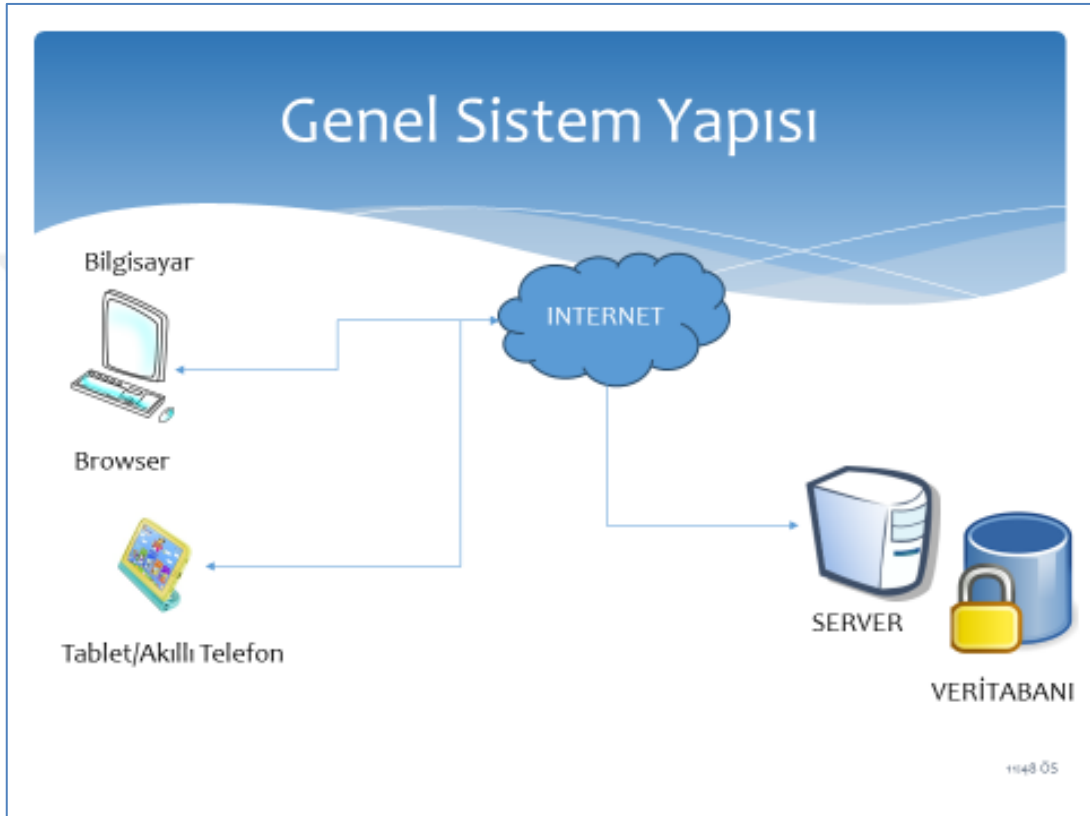
Bu temel işlevleri gerçekleştiren uygulama sadece bir sınıf ve bir eğitmen tabanında çalışmaktadır. Sistemin daha geniş ölçekte uygulanması daha sonraki aşamalara bırakılmıştır.

Eğitim sistem yazılımlarının geliştirilmesinde şelale yazılım geliştirme (Waterfall) gereği şu aşamalar yürütülmüştür: Gereksinim analizi, sistem analizi, sistem tasarımı, kod tasarımı, kodlama, test süreçleri uygulaması, kullanıma alma.

##### 4.4.1 Merkezi Birim

Tasarlanan ve geliştirilen bilgisayar programlama öğrenim setinin mühendislik çalışması; en az doksan adam/gün sürmüştür. Merkezi sistem Java teknolojileri, JSF, Hibernate (ORM Object/Relational Mapping aracı) ve MySql veri tabanı gibi açık kaynak kodlu altyapılar kullanılarak geliştirilmiştir.

Eđitim altyapısı e-eđitim altyapısı olarak dzenlenmiřtir. Dnyanın her yerinden ulařılabilecek durumda olan ara yz bireysel kullanıcılar, eđiticiler, eđitim kurumları tarafından gerekli kayıt yapıldıktan sonra kullanılabilir.



řekil 4.10. Merkezi Eđitim Portal Yapısı

Bireysel kullanıcılar, kayıt yapmalarının ardından ilgili formlar ve testler sonrası platformu kullanmaya bařlarlar. Eđiticilerin kayıt olmaları ardından eđitim verdikleri sınıfın ođrencileri ve eđitim materyali atamasından sonra etkileřimli olarak eđitici- ođretici iliřkisi sađlanır. Kurumlar: Kurumların kayıtları, eđiticilerin kayıt ve tanımlanması, kurum eđitim materyallerinin tanımlanması ve oluřturulması ile bařlar.

Merkezi yapı tüm sistemin oluşturulması geliştirilmesi ve yeni içeriklerin eklenmesi için bir yönetici modülü içermektedir. Yönetim modülü öğrencilerin, eğiticilerin ve kurumların yönetilmesini sağlar. Yeni oluşturulacak içerikler (görsel ve diğer) ve görevler bu modül aracılığı ile düzenlenir ve onay mekanizmasının ardından kullanılır hale gelir.

#### **4.4.2 Kullanıcı Uygulama Geliştirme Arabirimi**

Teorik ve izlenebilir uygun eğitim altyapısını oluşturmak için geliştirilen kullanıcı uygulama geliştirme arabirimi Açık Kaynak kodlu; Blockly altyapısı kullanılarak geliştirilmiştir. Kullanıcı arabirimi kullanımı kolay ve hata yapmadan görevleri tamamlamak için tasarlanmış ve geliştirilmiştir. Çalışması için web tarayıcı (Browser) yeterlidir ve herhangi bir plugin veya download edilen bir uygulama yüklenmesine gerek olmadan kullanılır. Arabirim HTML ve JavaScript uygulaması olarak geliştirilmiştir. Eğitimler bilgisayara yüklenerek internet bağlantısı olmadan da kullanılabilir. Chrome, Firefox, Safari, Opera, IE. gibi başlıca tarayıcıları destekler. Geliştirilen kodları adım adım çalıştırmak mümkündür (Execute code step-by-step).

Geliştirilen blok kodlar istenildiği anda Python, php, Dart, Javascript. dillerine dönüştürmek ve ilgili derleyici/yorumlayıcılar ile çalıştırıp sonuç elde etmek mümkün olmaktadır. Kullanıcı uygulama arabirimi javascript altyapısı kullanılarak ve web tabanlı olarak gerçekleştirilmiştir.

İzlenice (curriculum) geliştirme çalışmasında oluşturulan program (syllabus) gereğince kazandırılacak yetkinlikler belirlenmiş olan plana uygun bir şekilde eğitime dahil

edilmiştir. Tablo 4.6.'da verilen eğitim kategorileri izlence çalışmasında değerlendirilmiştir.

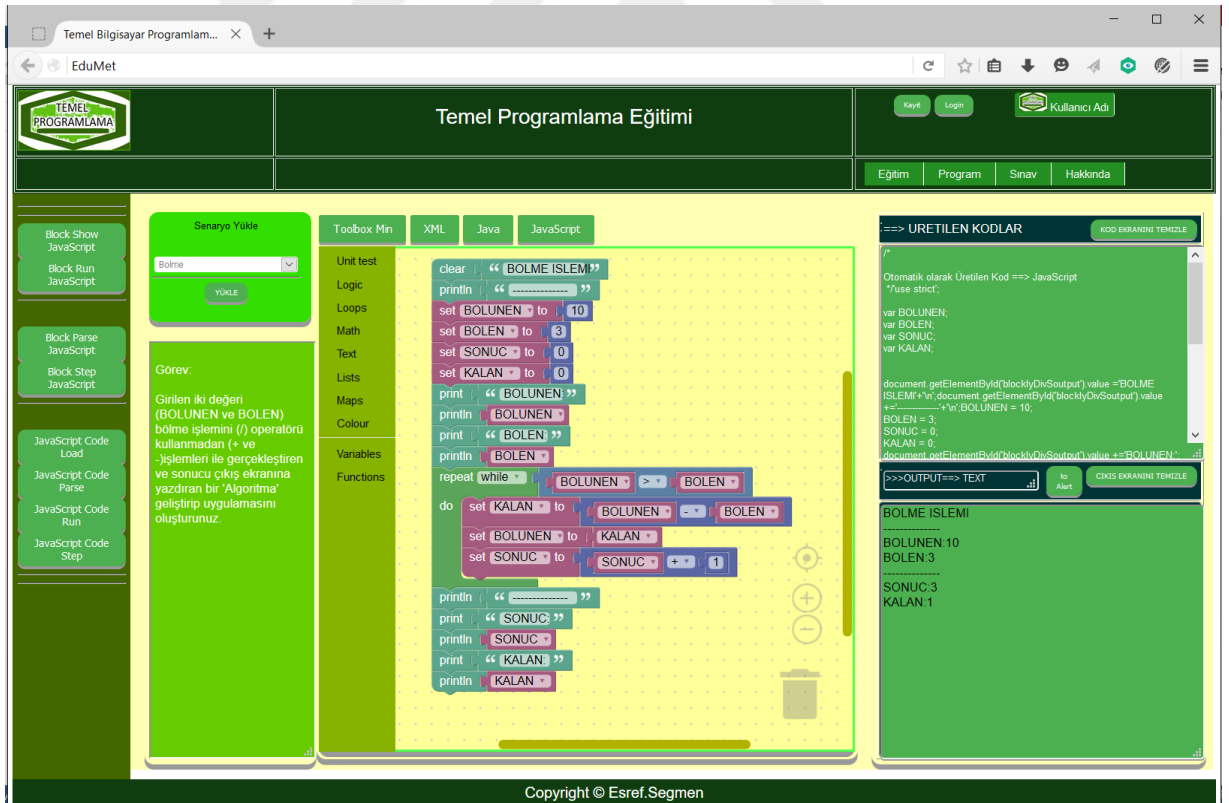
Tablo 4.6. Eğitim Kategorileri

KOD	KONU BAŞLIĞI
01	Soyut düşünme ve Programlama
02	Değişkenler Bellekte saklama(Atama)
03	Karşılaştırma kavramı
04	İteratif (kurgusal döngüsel) düşünme. Ardışıl İşlemler ve Programlama
05	İteratif (kurgusal döngüsel) düşünme. Belirli bir sayıda döndürme
06	İteratif (kurgusal döngüsel) düşünme. Sonsuz döngü kavramı ve Şart oluşana kadar döndürme
07	Ardışıl İşlemler ve Dizi Kavramı
08	Birden fazla boyutta döndürme ve Çok Boyutlu Dizi Kavramı
09	Ardışıl İşlemler ve Arama/Sıralama kavramları

İzlenceye bağlı olarak yapılacak eğitim amaçlı eğitim seti hedefler doğrultusunda temel programlama eğitimini eksiksiz sağlar. Oyun geliştirme yolu ile programlama eğitimi oldukça etkili bir yöntem olduğu düşünülmele birlikte bu konuda yapılmış kapsamlı bir araştırma bulunmamaktadır. Amacımız oyun geliştirmek yolu ile oyun kahramanına

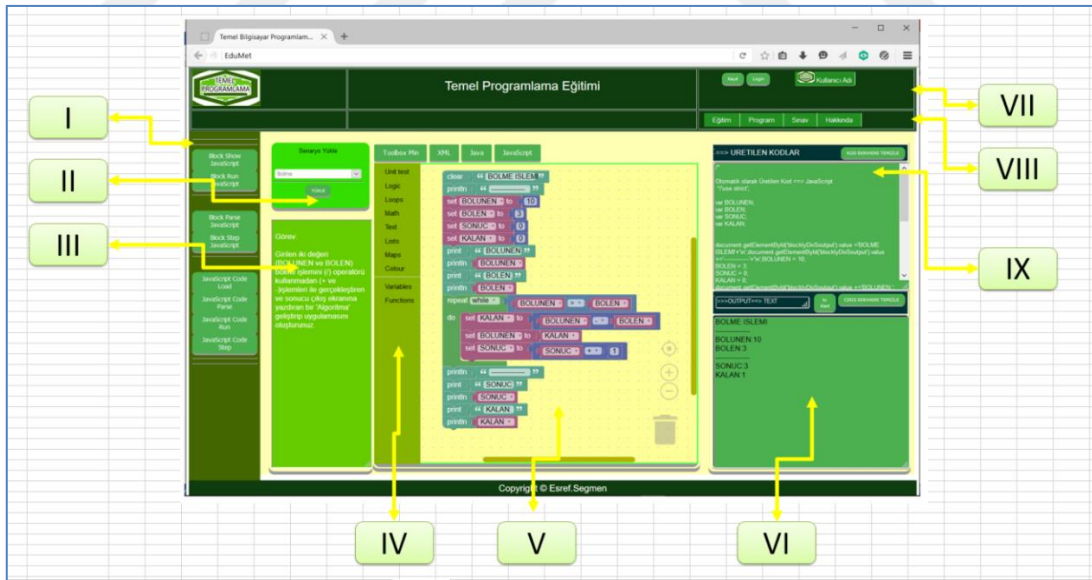
komut verme ve bazı engelleri aşmak için ardışık işlemler yaptırma ve programlama eğitimine katkıda bulunmak yerine doğrudan kod geliştirme ve yazılı kod parçasını gösterme şeklindedir.

Problem ve görev tarif edilir, gerekli kısıtlamalar ile algoritmaların oluşturulması istenir. Bloklar kullanarak kural hatası yapmadan sadece algoritmaya odaklanarak yapılacak çalışmaların yanında doğrudan kod yazarak da algoritma gerçekleştirebilir. Yapılan çalışma geliştirilen algoritma istenildiği anda koşturulabilir (RUN) ve hatta adım adım (step by step) çalıştırılacak hata ayıklama (Debug) özellikleri de vardır.



Şekil 4.11. Geliştirilen Eğitim Aracı İle Örnek Uygulama

Programlama öğrenimi için geliştirilen bu uygulama daha önce belirlediğimiz izlencede 9 kategoride eğitimi kapsamakta; her kategori minimum 4 senaryo ile o kategorinin gereklerini yerine getirmek ve ilgili konuların hedeflerine paralel görevler (problemler) vermek üzere oluşturulmuştur. Her kategori minimum 5 görevden oluşmaktadır. Kayıt olan öğrenci metodun gereği tanımlamalar ve ön işlemler sonrasında 1. Kategori öğrencisi olarak eğitime başlar. Kategorinin eğitim içeriğini oluşturan 5 görevi tamamlaması ile kategori tamamlanmış olur. Her kategoriyi bitirmek için beş görevi başarmış ve pekiştirme sorularını tamamlamış olması gereklidir. Öğrenci 1.kategorinin birinci görevinde verilen değerler ile çözümü oluşturur. Geliştirilen eğitim uygulamasının bölümleri Şekil 4.12.de gösterilmiş ve ilerleyen paragraflarda bu bölümler açıklanmıştır.



Şekil 4.12. Geliştirilen Eğitim Aracı Uygulama Bölümleri



**Bölüm I:**

Aşağıda işlevleri yürüten düğmeler vardır:

Geliştirilen blok uygulamasının Javascript ve diğer dönüşüm yapılan dillerdeki kodlarının görülmesini sağlayan düğme;

-Geliştirilen blok uygulamasını çalıştıran düğme;

-Geliştirilen blok uygulamasının kodlarının adım adım çalıştırmak için uyarlanmasını sağlayan düğme.

-Geliştirilen blok uygulamasının kodlarının adım adım çalıştıran düğme.

**Bölüm II:** Bulunulan kategoride izin verilen görevlerin yüklenmesini sağlayan bölümdür.

**Bölüm III:** Görev tanımının görüldüğü bölümdür.

**Bölüm IV:** Toolbox: Görevin çözümü için gerekli blokların yüklendiği bölümdür.

**Bölüm V:** Çözümün gerçekleşeceği bölümdür.

**Bölüm VI:** Çıkış ekranı: Blok yapısının veya kodların çalıştırıldığında sonuçların oluşacağı bölümdür.

**Bölüm VII:** Kullanıcı kayıt giriş ve profil ile ilgili bölümlerdir.

**Bölüm VIII:** Uygulama arabirimi diğer etkinlikleri ile ilgili bölümdür.

**Bölüm IX:** Oluşturulan blok yapısının XML ve kodlarının otomatik olarak oluşacağı bölümdür.



## 5 UZMANLARLA YAPILAN GEÇERLEME

### ETKİNLİKLERİ

#### 5.1 Uzman Programcılar İle Yapılan Çalışma

Bilişim teknolojileri sektörünün en önemli elemanlarından biri yazılım geliştiricilerdir. Yazılım geliştirme işi ve bu işi seçen bireyler arasında ilişki ve etkileşimi inceleyebilmek adına bir takım çalışma ve analizler yapılmıştır. Yazılım geliştiricilerin ortak özellikleri ve yazılım geliştirme işinin temel gereksinimleri üzerinden yola çıkarak mesleğin icrası üzere tespitler yapmak istedik. Profesyonel yazılım geliştiriciler, BHB bilişsel hazır bulunuşluk ölçeği testi ve programlamaya ilişkin öz yeterlilik ölçeği çalışmalarında uzman olarak yer aldılar.

##### 5.1.1 Uzman Programcılar İle Yapılan Çalışmanın Özeti

Yazılım geliştiricilerin gereksinimlerinden yola çıkarak bireysel özelliklerin etkin bir yazılım geliştirici olabilme yolundaki avantaj ve dezavantajları üzerinde duruldu. Yazılım geliştirme mesleğini icra eden insanların ortak özellikleri ve bireylerin buna uygunluğunu ve bu uygunluğa oranla ne kadar başarılı oldukları incelendi.

Geliştirilen BHB ölçeği, on dokuz, bir ile on yedi yıl arası tecrübeye sahip uzman programcılar ile yapılan bir çalışmada ele alınmıştır. Bu grup kariyerleri boyunca farklı projelerde ve yetkinlik gerektiren işlerde çalışmış farklı düzeyde uygulama geliştirme yetkinliklere haiz, yazılım geliştirme profesyoneli olarak tanımlanabilir. Uzmanlar,

yazılım geliştirme sektörü içerisinde profesyonel olarak çalışan farklı alanlarda uzmanlaşmış, Java, .NET, C++ ve benzeri uygulama geliştirme ortamlarında profesyonel kod geliştirmiş ve hala uygulama geliştirme ekiplerinde, kod geliştirici, analizci, test uzmanı ve dokümantasyon uzmanlarından oluşan ekip üyeleridir.

### 5.1.2 Uzmanların Birbirlerinden Ayıran Faktörler

- Tecrübe
- Mezun Olduğu Okul Bölüm (Programcı-Mühendis)
- Uzman Görüşü - Bitirdiği Projeler
- Uzman Görüşü – Potansiyel
- Bilgisayar Programlama için Öz Yeterlilik ölçeği değerlendirilmesi

Tecrübe: Aktif olarak uygulama geliştirme ekibinde çalışma süresi (yıl)

Uzman Görüşü: İlgili bölüm yöneticisinin programcı hakkındaki kanaati 0- 100 puan arasında değerlendirildi.

Uzman Görüşü Potansiyel: İlgili bölüm yöneticisinin programcı hakkındaki gelecek öngörüsünü oluşturan kanaati 0 – 100 puan arasında değerlendirildi.

Araştırmaya katılan adayların sonuçları incelendiğinde okul ve bölümlerinin test sonuçlarını doğrudan etkileyen bir parametre olmadıkları görüldü. Teste giren adayların Uzman puanları incelendiğinde öz yeterlilik ve anket puanı yüksek olan adayların aynı zamanda potansiyel gelişimi için verilen puanların yüksek olduğu görülmektedir.

Uygulama geliştirme seviyeleri ve yetkinlikler geliştirme hızları bir takım uzman görüşleri ile değerlendirmeye dâhil edilmiştir. Uzman görüşü testlerin sağlamlasını yapacaktır.

Uzmanlar ile çalışmaların yukarıda sıralanan ölçütlere uygun objektif bir sonuç verip veremeyeceği belirlenmelidir. Test sonuçları yukarıdaki sıralanan parametreler ve test sonuçlarına göre değerlendirildi anlamlı bir korelasyon gösterdiği görüldü.

Tablo 5.1. Uzman Programcılar Çalışma Sonuçları

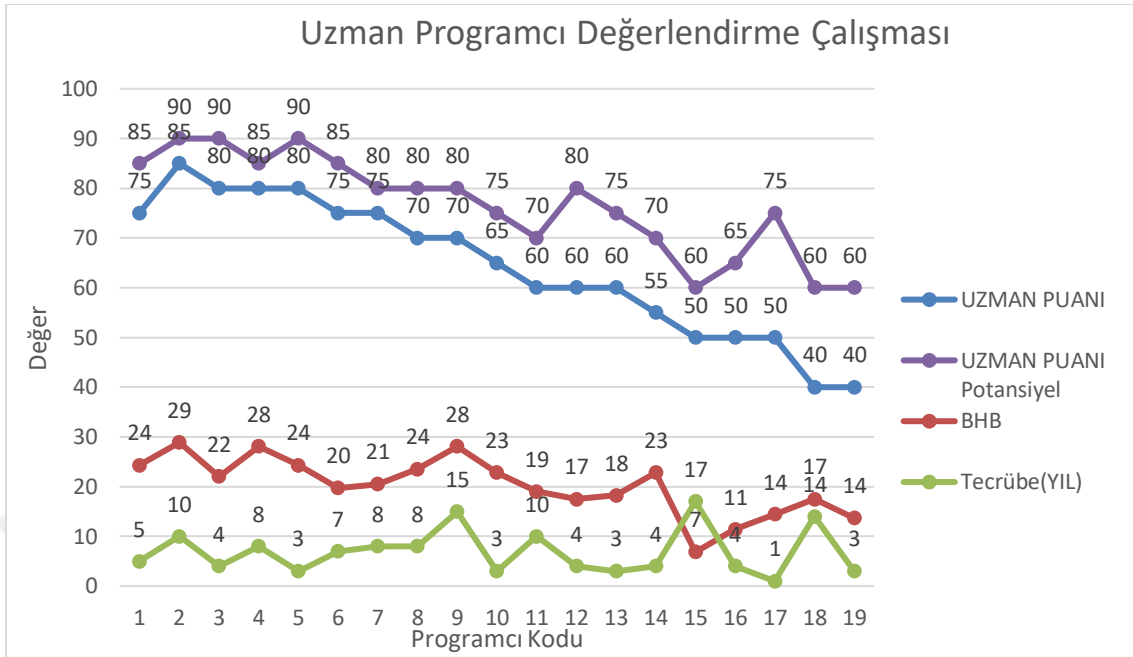
NO	KOD	TECRÜBE (yıl)	UZMAN PUANI	UZMAN PUANI Potansiyel	ÖZ YETERLİLİK PUANI	BİLİŞSEL HAZIR BULUNUŞLUK TEST PUANI
1	AXX7842	5	75	85	59	24
2	SDF744	10	85	90	60	29
3	GRFT345	4	80	90	58	22
4	SDFG746	8	80	85	50	28
5	AERK748	3	80	90	40	24
6	ADRT541	7	75	85	50	20
7	AERT743	8	75	80	40	21
8	AERT747	8	70	80	54	24
9	FRTY749	15	70	80	49	28
10	RTTT759	3	65	75	44	23
11	FFFF855	10	60	70	47	19
12	FTFT954	4	60	80	45	17
13	TYYY052	3	60	75	45	18
14	LLLL051	4	55	70	38	23
15	PIYY556	17	50	60	54	7
16	DYD453	4	50	65	45	11
17	RRTR857	1	50	75	44	14
18	GHJK958	14	40	60	49	17
19	ERTY660	3	40	60	30	14
ORT.		6,89	64,21	76,58	47,42	20,20

Çalışma başında programcılara birer kod atanmıştır. Denekler çalışmaya bu kod ile girmişlerdir. 19 kişilik uzman programcılara, 33 soruluk BHB ölçeği uygulanmış, programlamaya ilişkin öz yeterlilik ölçeği ile değerlendirmeye tabi tutulmuş ve isim verilmeden sonuçlar bu kodlara göre değerlendirilmiştir,(Tablo 5.1.).

Yazılım geliştiriciler bazı durumlarda gerçekten bireysel özelliklerini yansıttığı için bazı durumlarda da gerçekten öyle olmaları gerektiğinin bilincinde olduklarından olsa gerek, bireysel özelliklerini ifade etme konusunda kendilerini bu meslek için yetkin veya kapasitesinin yeterli olduğunu iddia edebilmektedirler. Bu bağlamda yaptığımız öz yeterlilik konusunda kendisini yetkin gören ve tecrübe konusunda da uzun yıllar programcılık yapmış olan kişilerin gerçekte o kadar başarılı olamadığı görüldü.

Yazılım geliştirme mesleği sonradan öğrenebilen bir meslek olmasına karşın, içerisinde büyük bilişsel birikim gereksinimi duyan bir meslek dalıdır. Ne kadar çok eğitim alınsa da probleme yaklaşım tarzı, problem çözme yaklaşımı, soyut düşünme ve ardışık ve döngüsel düşünme kabiliyetleri gibi özellikler kişiler arasında farklılıklar gösterebilmektedir.

Yazılım geliştirme mesleğine yatkın olan kişilerin veya bu meslek dalına uygun olduğu düşünülen kişilerin eğitim durumu, yeniliklere açık olma, öğrenmeye çaba gösteren bireyler olması da bu meslek dalında başarıyı getiren diğer bir etkidir. Yetenek okulu olmayan tek şeydir ancak bu yazılım geliştirme mesleğinin icrası için tek başına yeterli midir? Bu soruya cevap aramak için farklı eğitim seviyesindeki bireyler üzerinde testler yapıldı. Bu bireylerin öz-yeterlilikleri de göz önünde bulundurarak değerlendirmeler yapılmıştır.



Şekil 5.1. Uzman Programcı Değerlendirme Çalışması

Birey belirli bir görev ile ilgili yeteneği bulursa dahi, sahip olduğu bilgi, beceri ve yeteneklerine sahip olduğunun bilincinde olmayabilir. Bu sebeple yapabileceği bir göreve teşebbüs etmeyebilir. Bunun için bireyin yeteneğinin farkında olması gerekmektedir. Öz-yeterlilik inancı yüksek olan kişiler karmaşık problemlere karşı daha ısrarlı olmakta, problemi çözme konusunda daha istekli çaba göstermekte ve kendini her konuda geliştirmeye açıktır.

Bireyler için şuan ve gelecekteki potansiyel durumları için uzman görüşleri ve yorumları alınarak sonuç değerlendirme tablosuna aktarıldı. Her iki puan ise 0-100 arasında verildi. Bu puanlamalar verilirken sınav veya anket sonuçları göz önüne alınmadan, uzman değerlendirmesi sonucu verildi.

Tüm kriterler toplanıp puanlar bazında istatistikler yorumlanarak çıkarımlar yapmaya çalışıldı. Bu ölçümlerde bireylerin okudukları okullar, bölümleri, tecrübeleri, öz yeterlilik puanları uzman puanları ve test puanları ele alınarak gerçekleştirildi.

Sınavlardan yüksek puan alan bireylerin kişisel özellikleri incelendiğinde okudukları okul bakımından anlamlı bir fark görülmezken, sadece sınav puanı olarak ele alındığında bilgisayar mühendisliği bölümü diğer bölümlere göre daha fazla olduğu görülüyor. Ancak test grubumuzda da bilgisayar mühendisliği bölümü mezun adayların fazlalığı göz önüne alındığında bu tablo normal karşılanabilir. Bunun yanında bilgisayar ile birinci dereceden alakalı olmayan sayısal bölüm mezunu adayların bilgisayar mezunu adayları geçebildikleri görülmektedir.

Aynı zamanda yukarıdaki tablo incelendiğinde daha az tecrübe sahibi adaylar daha fazla gelişim potansiyeline sahip oldukları çıkarılabiliyor. Uzman puanları yine incelendiğinde yüksek puanlı adayların öz yeterlilik puanlarının da diğer adaylara göre yüksek oldukları görülmektedir. Bu da kendine güvenen adayların geleceklerinin daha parlak olacağı anlamını çıkarabilmemizi sağlıyor.

Değerlendirme sonuçlarına göre, yazılım geliştirme mesleğinde başarı ve gelecek vaat eden bir birey olabilmek öncelikle yetkinlikler bakımından yeterli olmak gerekliliği ortaya çıkıyor. Bu yetkinlikleri ancak iyi bir eğitimle pekiştirdikten sonra, etkin sonuç alınabilir.



## 6 ÖĞRENCİLERLE YAPILAN GEÇERLEME

### ETKİNLİKLERİ

2015-2016 öğretim yılı ilkbahar dönemi öğrencilerinden 44 kişilik iki gruba toplam 5 hafta saat sürecek önerilen yöntem ile hazırlanmış bir bilgisayar programlama eğitimi gerçekleştirilmiştir. Deneysel, Boğaziçi Üniversitesi Bilgisayar ve Öğretim Teknolojileri Bölümü birinci sınıf öğrencilerinin zorunlu CET 102 dersinin içerisinde yapılmıştır.

Çalışma dokuz seviyeden oluşmakta ve İzlence Tablo 4.1. tam uyumludur. Seviyeler ve içerikleri Tablo 4.6.'da sıralanmıştır. Her seviye bir veya daha fazla örnek ile çalışılmış tam öğrenme gerçekleşene kadar serbest ve planlı çalışma yapılmıştır. Yapısı önceki bölümlerde anlatılan eğitim metodunun laboratuvar ortamında yapılan deneyin adımları şunlardır:

-Geçerleme Bilgisayar ve Öğretim Teknolojileri bölümü birinci sınıf öğrencileri zorunlu CET 102 dersi içerisinde ilk beş haftayı kapsayacak şekilde gerçekleştirildi.

-Ders laboratuvar ortamında ve her öğrenciye bir bilgisayar tahsis edilerek yapıldı.

-Tasarlanan eğitim metodunu görevleri, ek çalışmaları, pekiştirme soruları, serbest çalışmalarını tek başlarına ve eğitmen dışında yardım almadan gerçekleştirmeleri esas koşul olarak belirlendi.

-Deney, CET 102 zorunlu dersinin saatleri içerisinde ve zaman ve kısıtlamalara uygun yürütüldü. Haftada 3 saat olarak tasarlanan ders Cuma saat 10:00-13:00 ve 14:00-15:00

saatleri arasında iki ayrı grup olarak gerçekleştirildi. Çalışmaya katılımın, derse katılım disiplinine uygun olması için devam zorunluluğu ve etkinliklere katılım gerekli kılındı. Bunların yanında ilk hafta eğitim çalışması sonunda neler kazanabilecekleri ve bilgisayar programlama eğitimleri açısından, öğrenciler bilgilendirildi.

-İzlenecede detaylandırıldığı üzere 9 ana bölümde gerçekleşen eğitim çalışmamız her hafta 3 bölüm işlenerek toplam 9 bölüm tamamlandı. Giriş haftası ve sonuç haftası ile çalışma toplam 5 hafta sürdü.

### **Deneysel Çalışmanın 1.Haftası**

Öğrencilere çalışma hakkında detaylı bilgi verildi. Eğitim çalışmasının metodolojisi, adımları içeriği anlatıldı. Yapılan eğitimin CET 102 dersinin içeriği açısından önemi vurgulandı (Süre 30 dakika).

Laboratuvar çalışması için geliştirilen eğitim aracı kullanımı ve özellikleri anlatıldı. Erişim kullanım laboratuvar saati dışında erişim gibi detaylar sunuldu. Kullanım kitapçığı tanıtıldı (pdf) (Süre 20 dakika).

BHB Ön Testi gerçekleştirildi. (Süre 60 dakika).

Programlamaya ilişkin Öz Yeterlilik ölçeği uygulandı. (Süre 20 dakika.).

Öğrenci Soru ve cevap bölümü (süre 20 dakika.).

Toplam 150 dakikalık süre tamamlandı.

## **Deneysel Çalışmanın 2. Haftası**

Her ders saati bir bölüm olmak üzere eğitimin aşağıdaki 1, 2 ve 3 bölümleri tamamlandı.

- a. Soyut düşünme ve Programlama
- b. Değişkenler Bellekte saklama (Atama)
- c. Karar Verme ve Karşılaştırma kavramı

Her bölüm kolaydan zora gidecek şekilde 5 ayrı alt bölüme ayrılmıştır. Her bölüm başlangıcında, modelin önerdiği ve gerekli teorik eğitimi vermek üzere video anlatımının yerine her alt bölüm hakkında bilgi ve teorik gerekli altyapı sunuldu. Görevlerin çözümü için kısa ön bilgiler içeren açıklama dokümanı verildi. Birinci görevler rehber öğretim elemanı yardımı ile çözüldü. İkinci, Üçüncü ve Beşinci görevlerin laboratuvar ortamında sadece rehber öğretmen tarafından gerektiğinde desteklenerek tamamlanması gerçekleştirildi. Laboratuvar saatleri dışında serbest çalışma için çözüm içermeyen eğitim aracı kullanımı ile dördüncü görevler laboratuvar saati dışında yapılmak üzere öğrencilerin ödev olarak verildi.

## **Deneysel Çalışmanın 3.Haftası**

Her ders saati bir bölüm olmak üzere eğitimin aşağıdaki 4, 5 ve 6 bölümleri tamamlandı.

4-İteratif (kurgusal döngüsel) düşünme. Ardışık işlemler ve programlama

5-İteratif (kurgusal döngüsel) düşünme. Belirli bir sayıda döndürme

6-İteratif (kurgusal döngüsel) düşünme. Sonsuz döngü kavramı ve şart oluşana kadar döndürme.

Her bölüm kolaydan zora gidecek şekilde 5 ayrı alt bölüme ayrılmıştır. Bölüm hakkında kısa bilgi verildi. Görevlerin çözümü için kısa ön bilgiler verildi. Birinci görevin rehber öğretim elemanı tarafından çözümü tamamlandı. İkinci, üçüncü ve beşinci görevlerin laboratuvar ortamında sadece rehber öğretmen tarafından gerektiğinde desteklenerek tamamlandı. Laboratuvar saatleri dışında serbest çalışma için çözüm içermeyen eğitim aracı kullanımı ile dördüncü görevler laboratuvar saati dışında yapılmak üzere öğrencilerin ödev olarak verildi.

#### **Deneysel Çalışmanın 4.Haftası**

Her ders saati bir bölüm olmak üzere eğitimin aşağıdaki 7, 8 ve 9 bölümleri tamamlandı.

- a. Ardışık İşlemler ve diziler.
- b. Birden fazla boyutta döndürme ve çok boyutlu diziler.
- c. Ardışık İşlemler ve arama/sıralama kavramları.

Her bölüm kolaydan zora gidecek şekilde 5 ayrı alt bölüme ayrılmıştır.

Bölüm hakkında kısa bilgi verildi. Görevlerin çözümü için kısa ön bilgiler verildi. Birinci görevin rehber öğretim elemanı tarafından çözümü tamamlandı. İkinci, Üçüncü ve Beşinci görevlerin laboratuvar ortamında sadece rehber öğretmen tarafından gerektiğinde desteklenerek tamamlandı. Laboratuvar saatleri dışında serbest çalışma

için çözüm içermeyen eğitim aracı kullanımını ile dördüncü görevler laboratuvar saati dışında yapılmak üzere öğrencilerin ödev olarak verildi.

### **Deneysel Çalışmanın 5.Haftası**

Eğitim Sonu değerlendirme sınavı yapıldı. Eğitim süresince verilmeye çalışılan eğitimin ne kadar başarılıabildiğine dair dönüt alınabilmesi için sorulara cevap verilmesi istendi ve sınav kuralları uygulandı. (Süre 45 dakika). BHB Son Testi gerçekleştirildi. (Süre 60 dakika).

Programlamaya ilişkin Öz Yeterlilik ölçeği tekrar uygulandı (Süre 20 dakika).

Eğitim çalışmasına karşı öğrencilerin görüşlerini almak üzere anket çalışması gerçekleştirildi (Süre 10 dakika).

Soru ve cevap ve teşekkür bölümü 15 dakika sürdü.

Toplam 150 dakikalık süre tamamlandı.

### **6.1 Deneysel Çalışma Sonuçlarının Değerlendirilmesi**

Deneysel çalışmanın sonuçların bazı bölümlerini gösteren veriler Tablolar 6.1. ve 6.2'de verilmiştir.

Tablo 6.1. Deneysel Çalışma Sonuçlarının Örnek Değerlendirme Tabloları-A

No	Student Number	Name	BHB EĞİTİM ÖNCESİ (ÖN TEST)						ÖN ÖLÇEK ÇALIŞMASI	BHB EĞİTİM SONRASI (SON TEST)						SON ÖLÇEK ÇALIŞMASI	EĞİTİM SONU SINAVI
			Algoritmik Düşünce	Ardışık ve Döngüsel Düşünme	Örüntü Tanıma	Problem Çözme	Soyut Düşünme Soyutlama ve Genelleme	Mantıksal çıkarım, Muhakeme, akıl yürütme	Öz yeterlilik Giriş	Algoritmik Düşünce	Ardışık ve Döngüsel Düşünme	Örüntü Tanıma	Problem Çözme	Soyut Düşünme Soyutlama ve Genelleme	Mantıksal çıkarım, Muhakeme, akıl yürütme	Öz yeterlilik Çıkış	EĞİTİM SONU BAŞARI SINAVI
1	2014100093	ACAkgün	7	8	15	15	17	18	49	9	10	18	18	20	22	58	10
2	2014100018	AKırıklı	8	9	15	16	19	20	13	10	12	20	22	25	26	33	9
3	2014451150	AEKara	5	5	10	10	11	12	31	9	10	17	17	18	19	29	9
4	2014100177	BParlak	8	9	15	16	19	20	36	8	10	20	21	23	24	36	7
5	2013100117	CTurna	9	10	16	18	20	21	32	10	11	18	18	20	21	31	8
6	2014100051	DTemel	6	7	13	13	14	15	13	9	10	16	16	17	18	36	10

Tablo 6.2. Deneysel Çalışma Sonuçlarının Örnek Değerlendirme Tabloları-B

YETKİNLİKLERE GÖRE PUANLAR BHO FARK							YETKİNLİKLERE GÖRE PUANLAR BHO FARK YÜZDE						
Algoritmik Düşünce	Ardışık ve Döngüsel Düşünme	Örüntü Tanıma	Problem Çözme	Soyut Düşünme Soyutlama ve Genelleme	Mantıksal çıkarım, Muhakeme, akıl yürütme	Puan Fark	Algoritmik Düşünce	Ardışık ve Döngüsel Düşünme	Örüntü Tanıma	Problem Çözme	Soyut Düşünme Soyutlama ve Genelleme	Mantıksal çıkarım, Muhakeme, akıl yürütme	Puan Fark %
2	2	3	3	3	4	4	28,57	25,00	20,00	20,00	17,65	22,22	22,22
2	3	5	6	6	6	6	25,00	33,33	33,33	37,50	31,58	30,00	30,00
4	5	7	7	7	7	7	80,00	100,00	70,00	70,00	63,64	58,33	58,33
0	1	5	5	4	4	4	0,00	11,11	33,33	31,25	21,05	20,00	20,00
1	1	2	0	0	0	0	11,11	10,00	12,50	0,00	0,00	0,00	0,00
3	3	3	3	3	3	3	50,00	42,86	23,08	23,08	21,43	20,00	20,00
3	3	7	7	7	7	7	50,00	42,86	58,33	53,85	43,75	41,18	41,18
1	1	2	3	3	3	3	14,29	12,50	14,29	20,00	17,65	16,67	16,67

Daha önceki bölümlerde yer alan yetkinliklerin bilgisayar programlama eğitimine katkıları olacağı üzerinde durulmuştu. Bu sonuçlarla yanında bilgisayar programlama eğitimi sürecinde alınan bilgi ve edinilen becerilerin zihinsel yetenek ve yetkinliklere olan katkısını da ölçmüş olacaktır.

Sonuçlardan genel olarak beklenti: Eğitim öncesi hazır bulunuşluk ölçeği testi sonuçları eğitim aracı vasıtası ile öğrenilen ve geliştirilen uygulama geliştirme yetkinliklerinin son teste yapılan hazır bulunuşluk testinin başka bir sürümü olan durum testi ile yapılacak karşılaştırma yapılır. Verilen eğitimin hangi yetkinlikler üzerinde pozitif veya negatif farklılıklar yarattığını tespit edilir. Bu sonuçlar, bize özelde kurulan altyapının

genelde ise programlama eğitiminin yetkinlikler üzerindeki pozitif/negatif etkisini ortaya çıkaracaktır.

Sonuçların beklendiği gibi olması durumunda tezin konusu olan önerilen ve geliştirilen modelin bilgisayar programlama eğitimine pozitif katkıları olduğunun iddiasını destekleyecektir. Ayrıca bilgisayar programlama eğitimi amaçlı metodun ve geliştirilen uygulamanın temel kavramlar olarak belirlenen problem çözme, soyut düşünme vb. yetkinliklerin özelde önerilen metot ve gerçekleştirilen altyapı genelde ise programlama eğitimine pozitif katkıları olduğu tezini destekleyecektir.

## 7 BULGULAR VE ÖNERİNİN DOĞRULANMASI

Yaptığımız deneysel çalışmanın sonuçlarını değerlendirmek üzere elde edilen tüm verilerin analizleri yapılarak sonuçların ne anlama geldiği üzerinde durulacaktır.

Yapılan çalışmanın istatistiksel sonuçları yorumlanacaktır.

Yapılan ön test ve son test eğitim sonrası sınav sonuçları bize deneysel çalışmamızda geliştirdiğimiz eğitim metodu ve eğitim aracı ile yaptığımız eğitimin hangi yetkinlikleri ne kadar geliştirdiği ve ön ve son test alt ve üst grupları arasındaki anlamlı fark olup olmadığı başka bir değiş ile yapılan ölçümün iyi ve iyi olmayan öğrencileri ayırıp ayıramadığı değerlendirilecektir.

### 7.1 Bilişsel Hazır Bulunuşluk Test Madde Analizi Bulguları

Madde analizi: Testin ölçüm yapılan gruba uygulanmasından sonra, test kapsamındaki maddelerin analiz edilmesi olarak tanımlanabilir. Yapılan analizler ile maddenin güçlük düzeyi (doğru cevaplandırılma sıklığı) ve maddenin ayırt ediciliği (bilenle bilmeyeni ne kadar iyi ayırdığı) üzerine test istatistiksel sonuçları yardımı ile yapılan çalışmadır. Madde analizi sonrasında ortaya çıkan sonuçlar ile kusurlu maddelerin belirlenmesinde, öğrencilerin yanlış kavramları hakkında bilgi edinilmesinde, testin iyileştirme gerektiren kısımları hakkında bilgi sahibi olunur. Madde analizi ile ilgili veriler, test sonuçlarının etkili bir şekilde incelenmesine yardımcı olur. Madde analizi sonuçları, teste giren öğrencilerin eksikliklerini ve sorular üzerinde analiz yapılarak eğitimin iyileştirilmesi çalışmalarına yardımcı olur. İşlem adımları şunlardır:



1. Test uygulandıktan sonra öğrencilerin test sonuçları incelenir.
2. Test doğru cevapları değerlendirilir ve doğru cevaplar puanlanır.
3. Test sonuçları yüksekten düşüğe doğru puan bazında sıralanır.
4. Üst grup olarak sonuçların yarısı, üçte biri veya %27'si alınır.
5. Alt grup olarak üst gruptan alınan kadar sonuç en düşük puanlar olarak alınır.
6. Testin her maddesi için ayrı ayrı güçlük ve ayıricılık indeksleri oluşturulur ve bir tablo hazırlanır ve üst ve alt gruplardaki öğrencilerin her madde cevaplama sıklıkları belirtilir.

Ölçme ve Değerlendirme Objektif Testler İçin Madde Güçlük İndeksi (p):

**Objektif Testler İçin Madde Güçlük İndeksi (p):**

$$\text{Güçlük (p) = İndeksi} = \frac{\text{Üst grupta doğru cevaplayanların sayısı + alt grupta doğru cevaplayanların sayısı}}{\text{Üst gruptaki öğrenci sayısı + Alt gruptaki öğrenci sayısı}}$$

Madde güçlük indeksi, 0 ile 1 arasında değerler alabilir.

Şekil 7.1. Ölçme ve Değerlendirme Madde Güçlük İndeksi (p)

Objektif Testler İçin Madde Ayırıcılık İndeksi (r) ayırıcılık indeksi, -1 ile +1 arasında değişecektir.

**Objektif Testler İçin Madde Ayırıcılık İndeksi (D):**

$$\text{Ayırıcılık (D) = İndeksi} = \frac{\text{Üst grupta doğru cevaplayanların sayısı} - \text{Alt grupta doğru cevaplayanların sayısı}}{\text{Grupların herhangi birindeki öğrenci sayısı}}$$

Madde ayırıcılık indeksi, -1 ile +1 arasında değerler alabilir

Şekil 7.2. Ölçme ve Değerlendirme Madde Ayırıcılık İndeksi (r)

Madde Analizi sonuçlarının yorumlanması: Madde analizi sonuçlarına bakılarak güçlük indekslerinin 0,2 ile 0,8 arasında gerçekleşmesi aranan ve olması gereken durumdur. Testin tümü için ortalama güçlük indeksi 0,5 olması ideal durumdur. Testte sonuçlarına göre maddelerin ayırıcılık indekslerinin 0,3 den fazla olması tercih nedenidir. Testin değerlendirmesinin doğru şekilde yapılabilmesi için güçlük ve ayırıcılık indekslerinin birlikte düşünülmesi gereklidir. Maddenin şartları sağlayamaması durumunda tadil edilmesi veya iptal edilmesi gereklidir.

Yapılan eğitim öncesi BHB ön test sonuçları madde analizi Tablo 7.1. de ayrıntıları ile listelenmiştir.

Tablo 7.1. Eğitim Öncesi BHB Ön Test Sonuçları Madde Analizi

SORU NO	Üst Grup (%27) Doğru Cevap	Alt Grup (%27) Doğru Cevap	Toplam Doğru Cevap	Madde Güçlük İndeksi (p)	Madde Ayırıcılık İndeksi (r)
1	7	4	19	0,46	0,25
2	11	10	39	0,88	0,08
3	12	8	39	0,83	0,33
4	6	0	7	0,25	0,50
5	11	8	34	0,79	0,25
6	11	7	32	0,75	0,33
7	11	6	31	0,71	0,42
8	3	0	9	0,13	0,25
9	3	3	8	0,25	0,00
10	11	7	36	0,75	0,33
11	12	7	37	0,79	0,42
12	10	8	33	0,75	0,17
13	0	0	0	0,00	0,00
14	10	10	39	0,83	0,00
15	1	0	2	0,04	0,08
16	11	7	34	0,75	0,33
17	1	0	2	0,04	0,08
18	5	3	19	0,33	0,17
19	3	0	8	0,13	0,25
20	12	10	40	0,92	0,17
21	7	3	14	0,42	0,33
22	10	1	22	0,46	0,75
23	12	11	43	0,96	0,08
24	9	7	31	0,67	0,17
25	9	3	21	0,50	0,50
26	8	5	22	0,54	0,25
27	12	10	42	0,92	0,17
28	12	9	40	0,88	0,25
29	3	4	8	0,29	-0,08
30	12	10	42	0,92	0,17
31	5	2	12	0,29	0,25
32	2	0	6	0,08	0,17
33	9	4	22	0,54	0,42
<b>Ortalama</b>	7,91	5,06	24,03	0,54	0,24

Tablo 7.2. Eğitim Sonrası BHB Son Test Sonuçları Madde Analizi

SORU NO	Üst Grup (%27) Doğru Cevap	Alt Grup (%27) Doğru Cevap	Toplam Doğru Cevap	Madde Güçlük İndeksi (p)	Madde Ayırıcılık İndeksi (D)
1	11	4	27	0,63	0,58
2	12	11	42	0,96	0,08
3	12	9	38	0,88	0,25
4	9	4	16	0,54	0,42
5	12	10	41	0,92	0,17
6	12	10	37	0,92	0,17
7	8	8	30	0,67	0,00
8	10	2	21	0,50	0,67
9	1	1	6	0,08	0,00
10	12	10	42	0,92	0,17
11	10	10	37	0,83	0,00
12	11	6	34	0,71	0,42
13	2	2	5	0,17	0,00
14	11	10	41	0,88	0,08
15	2	0	2	0,08	0,17
16	12	10	41	0,92	0,17
17	0	0	1	0,00	0,00
18	6	4	19	0,42	0,17
19	7	0	14	0,29	0,58
20	11	8	38	0,79	0,25
21	9	3	22	0,50	0,50
22	7	4	21	0,46	0,25
23	11	10	41	0,88	0,08
24	10	7	33	0,71	0,25
25	11	7	34	0,75	0,33
26	11	6	35	0,71	0,42
27	11	8	36	0,79	0,25
28	11	8	39	0,79	0,25
29	4	2	8	0,25	0,17
30	12	10	42	0,92	0,17
31	12	8	38	0,83	0,33
32	12	7	35	0,79	0,42
33	12	5	33	0,71	0,58
<b>Ortalama</b>	9,21	6,18	28,76	0,64	0,25

Yapılan Eğitim sonrası BHB son test sonuçları madde analizi Tablo 7.1. de ayrıntıları ile listelenmiştir.

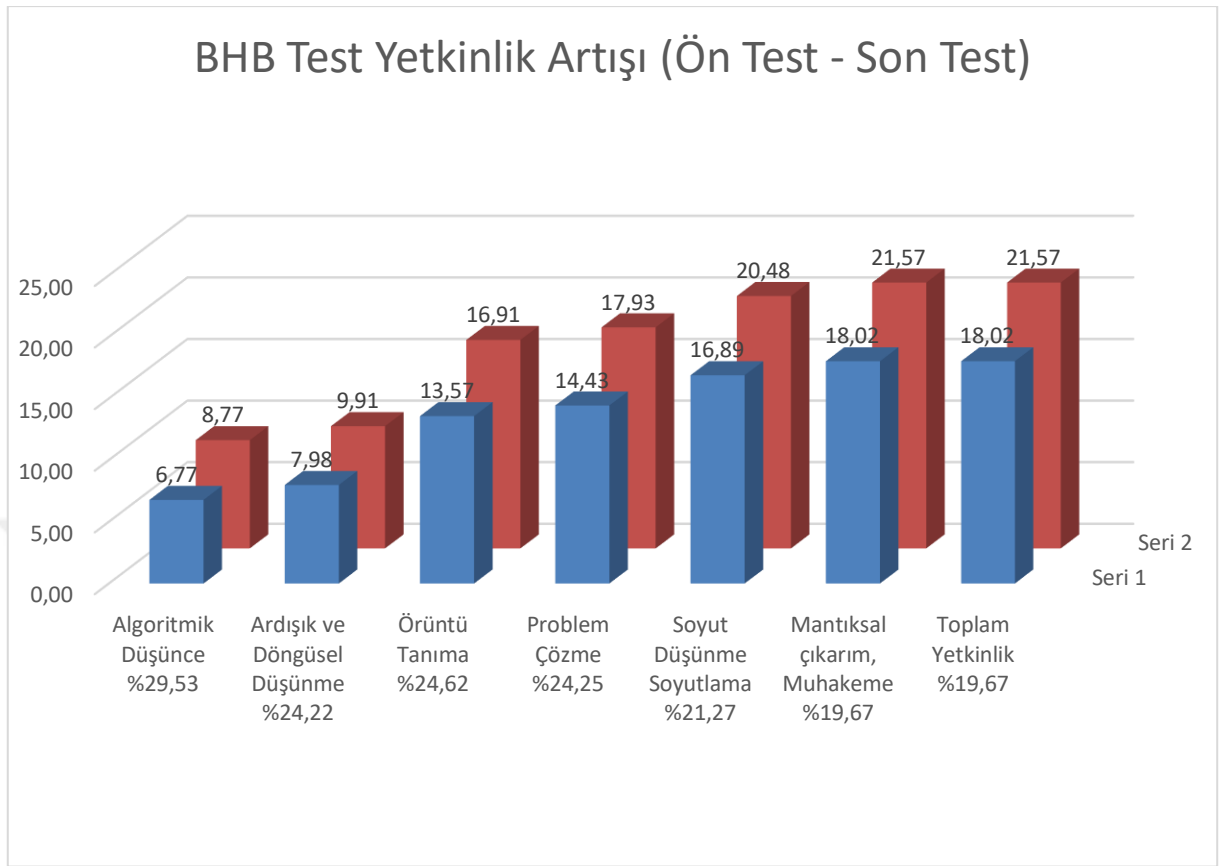
Yapılan madde analizi güçlük indeksi ortalaması ön test: 0.54 son test: 0.64 olarak çıkmıştır. Her iki sonuçta normal sınırlar içerisinde. Madde analizi ayırıcılık indeksleri ön ve son test için Tablolar 7.1. ve 7.2.'de görüldüğü şekildedir. Maddelerin ayırıcılık indeksleri klasik madde analizine göre hesaplanmıştır, çoğu madde için elde edilen ayırıcılık indekslerinin çoğu literatürde kabul gören sınırlar içinde değildir, bunun temel sebebi maddelere yanıt veren denek sayısının oldukça küçük olması ve denek grubunun görece homojen bir grup olması olarak açıklanabilir.

## **7.2 Geçerleme Çalışması Bulguları**

Gerçekleştirilen programlama eğitimi laboratuvar çalışmasının sonuçlarının değerlendirmesi yapılmıştır: Değerlendirmeye beş haftanın tamamına katılan 44 öğrenci dâhil edilmiştir. Gerçekleştirilen beş ölçme sınav ve değerlendirme sonuçları Tablo 7.3.'de gösterilmektedir.

Tablo 7.3. Deneysel Çalışma Sonuçlar

YETKİNLİKLERE GÖRE PUANLAR - BHB GİRİŞ							
	Algoritmik Düşünme	Ardışık ve Döngüsel Düşünme	Örüntü Tanıma	Problem Çözme	Soyut Düşünme Soyutlama ve Genelleme	Mantıksal çıkarım, Muhakeme, akıl yürütme	Toplam Yetkinlik
ORTALAMA	6,77	7,98	13,57	14,43	16,89	18,02	18,02
ORTANCA	7	8	14	14	17	18	18
YETKİNLİKLERE GÖRE PUANLAR - BHB ÇIKIŞ							
	Algoritmik Düşünme	Ardışık ve Döngüsel Düşünme	Örüntü Tanıma	Problem Çözme	Soyut Düşünme Soyutlama ve Genelleme	Mantıksal çıkarım, Muhakeme, akıl yürütme	Toplam Yetkinlik
ORTALAMA	8,77	9,91	16,91	17,93	20,48	21,57	21,57
ORTANCA	9	10	18	18	21	22	22
YETKİNLİKLERE GÖRE PUANLAR - BHB FARK							
	Algoritmik Düşünme	Ardışık ve Döngüsel Düşünme	Örüntü Tanıma	Problem Çözme	Soyut Düşünme Soyutlama ve Genelleme	Mantıksal çıkarım, Muhakeme, akıl yürütme	Toplam Yetkinlik
ORTALAMA	2,00	1,93	3,34	3,50	3,59	3,55	3,55
ORTANCA	2	2	3	3	3	3	3
YETKİNLİKLERE GÖRE PUANLAR - BHB FARK YÜZDE							
	Algoritmik Düşünme	Ardışık ve Döngüsel Düşünme	Örüntü Tanıma	Problem Çözme	Soyut Düşünme Soyutlama ve Genelleme	Mantıksal çıkarım, Muhakeme, akıl yürütme	Toplam Yetkinlik
Yüzde Değişim	29,53	24,22	24,62	24,25	21,27	19,67	19,67



Şekil 7.3. BHB Test Yetkinlik Artışı (Ön Test - Son Test)

### 7.2.1 Bulgu 1: Yetkinlikler Eğitim Sonrası Yetkinlik Artış Oranları

Eğitimin, BHB ön testi ortalaması 18,02 doğru cevap sayısını BHB son testi 21.57 ye yükselterek sonucu pozitif yönde %19,67 arttırdığı ortaya çıkmıştır.



Şekil 7.4. BHB Yetkinlikler Eğitim Sonrası Yetkinlik Artış Oranları

Öğrencinin yetkinliklerini ölçtüğümüz BHB ön ve son sınavları sonuçlarına göre öğrencilerin eğitim sonunda; Algoritmik Düşünme 29,53 Ardışık ve Döngüsel Düşünme: 24,22, Örüntü Tanıma: 24,62, Problem Çözme: 24,25, Soyut Düşünme Soyutlama ve Genelleme: 21,27, Mantıksal çıkarım, Muhakeme, akıl yürütme: 19,67 yetkinlikler bazında arttığı ortaya çıkmıştır. Son yetkinlik tüm yetkinlikleri kapsadığı için toplam sonuç da ortalama 19,67 artış göstermiştir.

Tablo 7.4. BHB Eğitim Sonrası Yetkinlik Artış Oranları

YETKİNLİKLERE GÖRE PUANLAR - BHB FARK YÜZDE							
	Algoritmik Düşünme	Ardışık ve Döngüsel Düşünme	Örüntü Tanıma	Problem Çözme	Soyut Düşünme Soyutlama ve Genelleme	Mantıksal çıkarım, Muhakeme, akıl yürütme	Toplam Yetkinlik
Yüzde Değişim	29,53	24,22	24,62	24,25	21,27	19,67	19,67



### 7.2.2 Bulgu 2: Programlamaya İlişkin Öz Yeterlilik Ölçeği Değişimi

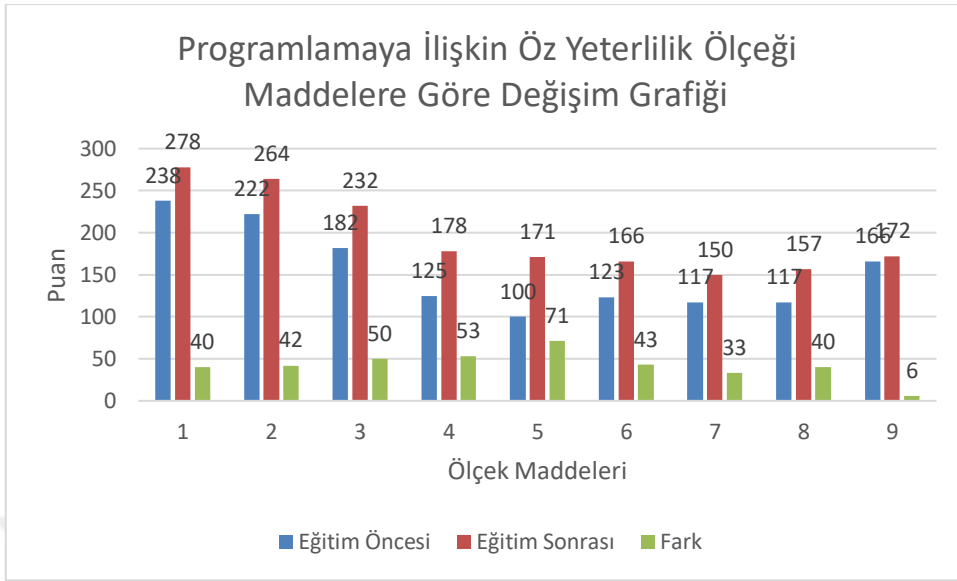
Eğitimin Programlamaya ilişkin Öz Yeterlilik ölçeği (Tablo 7.5) sonuçları: Eğitim başı yapılan ve eğitim sonunda yapılan değerlendirmelerde 33,1 den 42,1 a çıkarak ortalama pozitif yönde %27,19 artmıştır.

Tüm ölçeğin değişimine bakıldığında öğrencilerin programlamaya ilişkin özgüveni eğitim sonunda %27.19 artarak oldukça başarılı olmuştur.

Tablo 7.5. Programlamaya İlişkin Öz Yeterlilik Ölçeği Değişim

Madde	1	2	3	4	5	6	7	8	9	Toplam Puan
Eğitim Öncesi	238	222	182	125	100	123	117	117	166	1390
Eğitim Sonrası	278	264	232	178	171	166	150	157	172	1768
Fark	40	42	50	53	71	43	33	40	6	378
Yüzde	16,81	18,92	27,47	42,40	71,00	34,96	28,21	34,19	3,61	27,19

Ölçek dokuz maddede öz yeterlilik düzeyi üzerinde durmaktadır ve bütün maddelerde ayrı ayrı pozitif yönde artış gözlenmektedir. En çok artış 5 maddede %71 oranında ikinci sırada ise %53 artış ile 4. maddede artış gözlenmiştir.



Şekil 7.5. Programlamaya İlişkin Öz Yeterlilik Ölçeği Madde Değişim Grafiği

### 7.2.3 Bulgu 3: Değerlendirme Sınavı Başarım Sonuçları

Önerilen eğitim metodu ile gerçekleştirilen, yetkinlik temelli eğitim platformunun başarıya ulaşip ulaşmadığını belirlemek için “Eğitim sonrası programlama değerlendirme (Bilgi Ölçme) sınavı” sonuçları incelenmiştir. Eğitim sonunda gerçekleştirilen “Eğitim sonrası Programlama Değerlendirme Sınavı” sonuçları alt ve üst grup öğrenciler (sınav puanlarının ortancası altındaki puanlara sahip olanlar alt grup, üstündekiler üst grup olarak nitelenmiştir) arasında anlamlı fark olduğunu göstermektedir. Eğitim sonrası programlama eğitiminin değerlendirme sınavı sonuçlarına göre öğrenciler ortalama 10 üzerinden 8,25 puan ile başarılı olmuşlardır. Standart sapma 1,67 olarak gerçekleşmiştir. Sadece bir öğrenci 10 üzerinden 3 puan olarak başarısız olmuştur. Gerçekleştirilen eğitimin başarımını gösteren eğitim sonu

değerlendirme sınavı ortalaması %82.5 olarak ortaya çıkmaktadır (Tablo 7.6). Eğitimin katkısı açıkça görülmektedir

Tablo 7.6. Değerlendirme Sınavı İstatiksel Değerleri

<b>Eğitim sonrası Programlama Değerlendirme Sınavı Sonuçları</b>	
ÖĞRENCİ SAYISI	44
TOPLAM	363,00
ORTALAMA	8,25
ORTANCA	8,50
STANDART SAPMA	1,67
BASIKLIK	0,94
ÇARPIKLIK	-0,96
VARYANS	2,82

#### 7.2.4 Bulgu 4: BHB t-test Analizleri Alt Grup – Üst Grup

BHB sonuçları ortanca altında kalanlara alt grup üstünde puan alanları üst grup olarak iki gruba ayrılarak üst ve alt grubun ortalama farkları bize bazı bulgular sunacaktır.

İstatistikte t-testi nicel araştırmalarda en çok kullanılan testlerden biridir. Aralıklı ya da oranlı ölçüm düzeyinde olup normal dağılım gösteren iki değer birbiri ile karşılaştırıldığı zaman çoğunlukla t-testinden yararlanır (Akbulut, 2016). Deneysel araştırmamıza katılan 44 öğrenci için gerçekleştirdiğimiz eğitimin başında ve sonunda gerçekleştirilen sınavlarda elde edilen veriler farklı sonuç tablosunun ortancaya göre iki gruba ayırıldı, Bu durumda sınavlardan yüksek puan alanlar ile düşük puanlar alanlar arasında yapılacak analizler ile sınavın veya testin ayırıcılığı üzerinde de aydınlatıcı sonuçlar sundu.

Tablo 7.7. Deney Sonuçları t-testi Toplu Sonuçlar

NO	ÖLÇÜM	t	sd	Sig. (2yönlü)	Ortalama Fark
1	Programlamaya İlişkin Öz Yeterlilik Ölçeği (Ön Test)	-10,57	42	0,000	-22,147
2	Programlamaya İlişkin Öz Yeterlilik Ölçeği (Son Test)	-11,2	42	0,000	-23,4369
3	Eğitim sonrası Programlama Değerlendirme Sınavı	-9,977	42	0,000	-2,7727
4	Algoritmik Düşünme (Ön Test)	-11,25	42	0,000	-2,1408
5	Ardışık ve Döngüsel Düşünme (Ön Test)	-9,223	42	0,000	-3,0124
6	Örüntü Tanıma (Ön Test)	-9,329	42	0,000	-3,6108
7	Problem Çözme (Ön Test)	-8,418	42	0,000	-3,9091
8	Soyut Düşünme Soyutlama ve Genelleme (Ön Test)	-9,155	42	0,000	-4,4833
9	Mantıksal Çıkarım, Muhakeme, Akıl Yürütme (Ön Test)	-9,235	42	0,000	-4,8447
10	TOPLAM PUAN (Ön Test)	-9,235	42	0,000	-4,8447

	Tablo 7.7.'den Devam				
11	Algoritmik Düşünme (Son Test)	-8,751	42	0,000	-2,9296
12	Ardışık ve Döngüsel Düşünme (Son Test)	-8,573	42	0,000	-3,793
13	Örüntü Tanıma (Son Test)	-10,01	42	0,000	-5,2589
14	Problem Çözme (Son Test)	-9,292	42	0,000	-5,5893
15	Soyut Düşünme Soyutlama ve Genelleme (Son Test)	-9,34	42	0,000	-5,5263
16	Mantıksal Çıkarım, Muhakeme, Akıl Yürütme (Son Test)	-9,686	42	0,000	-5,8083
17	TOPLAM PUAN (Son Test)	-9,686	42	0,000	-5,8083

Yapılan Deneysel çalışmanın sonuçlarının istatiksel olarak t-testi incelemesi, tüm sonuçlar ortancanın altında kalan ve üstünde olan olarak iki gruba ayrılmıştır. Uygulanan t-testi sonuçları tüm parametrelerde anlamlı farklar ortaya çıkarmıştır. Yaptığımız deney sonuçları ayırıcılık açısından alt ve üst gruplar arasında farklar göstermektedir. Yapılan sınavların yüksek ve düşük seviye yetkinliklere sahip öğrencileri ayırabildiğini göstermektedir.

## 1- Ön Test Programlamaya İlişkin Öz yeterlilik Ölçeği

Programlamaya İlişkin Öz Yeterlilik Ölçeği (Ön Test) üst grup ve alt grup arasındaki fark t-testi uygulanarak ölçümlenmiş ve iki grup sonuçları arasında anlamlı bir fark ortaya çıkmıştır.

Programlamaya ilişkin öz yeterlilik ölçeği eğitim öncesi uygulamasından elde edilen veriler çözümlenmiştir. Programlamaya ilişkin öz yeterlilik ölçeği puanlarına göre alt ve üst grup olarak ayrılan grupların puan ortalamaları arasında üst grup lehine anlamlı fark vardır: ( $t(42) = -10,57$   $p = 0,000$ ), alt (ort: 23,043; ss: 7,2581) ve üst (ort:45,190; ss: 6,5774). Grupların programlamaya ilişkin öz yeterlilik ölçeği puan ortalamalarının anlamlı olarak farklı olması sonucunda programlamaya ilişkin öz yeterlilik ölçeği ön testinin ölçtüğü özellikler bağlamında öğrencileri ayırabildiği söylenebilir.

## 2- Son Test Programlamaya İlişkin Öz yeterlilik Ölçeği

Eğitim sonrasında gerçekleştirilen, Programlamaya İlişkin Öz yeterlilik Ölçeği (Son Test) üst grup ve alt grup arasındaki ilişki t-testi ile uygulanarak ve iki grup sonuçları arasında anlamlı bir fark olduğu ortaya çıkmıştır.

Programlamaya ilişkin öz yeterlilik ölçeği eğitim sonrası uygulamasından elde edilen veriler çözümlenmiştir: Programlamaya ilişkin öz yeterlilik ölçeği puanlarına göre alt ve üst grup olarak ayrılan grupların puan ortalamaları arasında üst grup lehine anlamlı fark vardır: ( $t(42) = -11,2$   $p = 0,000$ ), alt (ort: 29,476; ss: 8,6233) ve üst (ort: 52,913; ss:

4,9167). Grupların programlamaya ilişkin öz yeterlilik ölçeği puan ortalamalarının anlamlı olarak farklı olması sonucunda programlamaya ilişkin öz yeterlilik ölçeği testinin ölçtüğü özellikler bağlamında öğrencileri ayırabildiği söylenebilir.

Ortalamalar arasındaki fark 23,4369 ve anlamlılık değeri  $0,000 < 0,05$  den küçüktür ve iki grup arasında ortalamalar açısından anlamlı bir fark vardır. Eğitim öncesi ortalamaların farkı 22,1470 iken eğitim sonrası alt ve üst grup ortalama farkı 23,4369 olarak ortaya çıkmıştır. Bu fark eğitim sonrası özgüveni yüksek öğrencilerin öz güven seviyesini, özgüveni düşük öğrencilere göre daha çok arttırdığı görülmektedir.

### **3-Eğitim Sonrası Programlama Eğitimi Değerlendirme (Bilgi Ölçme) Sınavı**

“Eğitim Sonrası Programlama Değerlendirme Sınavı” eğitim sonrası öğrencilerin altı haftalık temel bilgisayar programlama eğitimi deneysel çalışmasından elde ettikleri faydayı ölçmek için düzenlenmiş bir sınavdır. Bu sınavın sonuçlarından elde edilen veriler çözümlenmiştir. Eğitim sonrası programlama değerlendirme sınavı puanlarına göre alt ve üst grup olarak ayrılan grupların puan ortalamaları arasında üst grup lehine anlamlı fark vardır: ( $t(42) = -9,997$   $p = 0,000$ ), alt (ort.: 6,864; ss: 1,2069) ve üst (ort: 9,636; ss:0,4924). Grupların Eğitim Sonrası Programlama Değerlendirme Sınavı puan ortalamalarının anlamlı olarak farklı olması sonucunda Eğitim Sonrası Programlama Değerlendirme Sınavı testinin ölçtüğü özellikler bağlamında öğrencileri ayırabildiği söylenebilir.

Deneysel çalışmaya katılan ve programlama eğitimi alan öğrencilerin eğitimden aldıkları faydanın ölçümlendiği sınav sonuçlarına göre ortalamaya göre yüksek puan

alan (eđitimden yüksek fayda sađlamıř) ođrenciler ile eđitimden dűřuk fayda sađlamıř ođrenciler arasında anlamlı bir fark vardır.

#### **4-Algorithmik Dűřünme (Ön Test)**

Algorithmik Dűřünme (Ön Test) sonuçları: eđitim öncesi BHB testi sonuçlarında ilgili soruların ođrencilerin altı haftalık temel bilgisayar programlama eđitimi öncesi seviyelerini tespit etmek için BHB testi sonuçlarında elde edilen verilerden oluşmaktadır. Bu sınavın sonuçlarından elde edilen veriler çözümlenmiştir. Eđitim öncesi BHB Algorithmik Dűřünme puanlarına göre alt ve üst grup olarak ayrılan grupların puan ortalamaları arasında üst grup lehine anlamlı fark vardır: ( $t(42) = -11,25$   $p = 0,000$ ), alt (ort.: 4,478; ss: 0,5931) ve üst (ort: 6,619; ss: 0,6690). Grupların BHB Algorithmik Dűřünme puan ortalamalarının anlamlı olarak farklı olması sonucunda BHB Algorithmik Dűřünme testinin ölçtüđü özellikler bağlamında ođrencileri ayırabildiđi söylenebilir.

#### **5-Ardışık ve Döngüsel Dűřünme (Ön Test)**

Ardışık ve Döngüsel Dűřünme (Ön Test) sonuçları: eđitim öncesi BHB testi sonuçlarında ilgili soruların ođrencilerin altı haftalık temel bilgisayar programlama eđitimi öncesi seviyelerini tespit etmek için BHB testi sonuçlarında elde edilen verilerden oluşmaktadır. Bu sınavın sonuçlarından elde edilen veriler çözümlenmiştir. Eđitim öncesi BHB Ardışık ve Döngüsel Dűřünme puanlarına göre alt ve üst grup olarak ayrılan grupların puan ortalamaları arasında üst grup lehine anlamlı fark vardır: ( $t(42) = -9,223$   $p = 0,000$ ), alt (ort.: 9,857; ss: 1,2762) ve üst (ort: 12,870; ss: 0,8689). Grupların BHB Ardışık ve Döngüsel Dűřünme puan ortalamalarının anlamlı olarak



farklı olması sonucunda BHB Ardışık ve Döngüsel Düşünme testinin ölçtüğü özellikler bağlamında öğrencileri ayırabildiği söylenebilir.

### **6-Örüntü Tanıma (Ön Test)**

Örüntü Tanıma (Ön Test) sonuçları: eğitim öncesi BHB testi sonuçlarında ilgili soruların öğrencilerin altı haftalık temel bilgisayar programlama eğitimi öncesi seviyelerini tespit etmek için BHB testi sonuçlarında elde edilen verilerden oluşmaktadır. Bu sınavın sonuçlarından elde edilen veriler çözümlenmiştir. Eğitim öncesi BHB Örüntü Tanıma puanlarına göre alt ve üst grup olarak ayrılan grupların puan ortalamaları arasında üst grup lehine anlamlı fark vardır: ( $t(42) = -9,329$   $p= 0,000$ ), alt (ort : 11,476; ss: 1,5368) ve üst (ort: 15,087; ss: 0,9960). Grupların BHB Örüntü Tanıma puan ortalamalarının anlamlı olarak farklı sonucunda olması BHB Örüntü Tanıma testinin ölçtüğü özellikler bağlamında öğrencileri ayırabildiği söylenebilir.

### **7-Problem Çözme (Ön Test)**

Problem Çözme (Ön Test) sonuçları: eğitim öncesi BHB testi sonuçlarında ilgili soruların öğrencilerin altı haftalık temel bilgisayar programlama eğitimi öncesi seviyelerini tespit etmek için BHB testi sonuçlarında elde edilen verilerden oluşmaktadır. Bu sınavın sonuçlarından elde edilen veriler çözümlenmiştir. Eğitim öncesi BHB Problem Çözme puanlarına göre alt ve üst grup olarak ayrılan grupların puan ortalamaları arasında üst grup lehine anlamlı fark vardır: ( $t(42) = -8,418$   $p= 0,000$ ), alt (ort.: 12,273; ss: 1,9069) ve üst (ort: 16,182; ss: 1,0527). Grupların BHB Problem Çözme puan ortalamalarının anlamlı olarak farklı olması sonucunda BHB Problem Çözme testinin ölçtüğü özellikler bağlamında öğrencileri ayırabildiği söylenebilir.

### **8-Soyut Düşünme Soyutlama ve Genelleme (Ön Test)**

Soyut Düşünme Soyutlama ve Genelleme (Ön Test) sonuçları: eğitim öncesi BHB testi sonuçlarında ilgili soruların öğrencilerin altı haftalık temel bilgisayar programlama eğitimi öncesi seviyelerini tespit etmek için BHB testi sonuçlarında elde edilen verilerden oluşmaktadır. Bu sınavın sonuçlarından elde edilen veriler çözümlenmiştir. Eğitim öncesi BHB Soyut Düşünme Soyutlama ve Genelleme puanlarına göre alt ve üst grup olarak ayrılan grupların puan ortalamaları arasında üst grup lehine anlamlı fark vardır: ( $t(42) = -9,155$   $p= 0,000$ ), alt (ort.: 13,350; ss: 2,0072) ve üst (ort: 17,833; ss: 1,2039). Grupların BHB Soyut Düşünme Soyutlama ve Genelleme puan ortalamalarının anlamlı olarak farklı olması sonucunda BHB Soyut Düşünme Soyutlama ve Genelleme testinin ölçtüğü özellikler bağlamında öğrencileri ayırabildiği söylenebilir.

### **9-Mantıksal Çıkarım, Muhakeme, Akıl Yürütme (Ön Test)**

Mantıksal Çıkarım, Muhakeme, Akıl Yürütme (Ön Test) sonuçları: eğitim öncesi BHB testi sonuçlarında ilgili soruların öğrencilerin altı haftalık temel bilgisayar programlama eğitimi öncesi seviyelerini tespit etmek için BHB testi sonuçlarında elde edilen verilerden oluşmaktadır. Bu sınavın sonuçlarından elde edilen veriler çözümlenmiştir. Eğitim öncesi BHB Mantıksal Çıkarım, Muhakeme, Akıl Yürütme puanlarına göre alt ve üst grup olarak ayrılan grupların puan ortalamaları arasında üst grup lehine anlamlı fark vardır: ( $t(42) = -9,235$   $p= 0,000$ ), alt (ort.: 15,286; ss: 2,2393) ve üst (ort: 20,130; ss: 1,0998). Grupların BHB Mantıksal çıkarım, Muhakeme, akıl yürütme puan ortalamalarının anlamlı olarak farklı olması sonucunda BHB Mantıksal Çıkarım,

Muhakeme, Akıl Yürütme testinin ölçtüğü özellikler bağlamında öğrencileri ayırabildiği söylenebilir.

### **10-Toplam Puan (Ön Test)**

BHB testi toplam puan (Ön Test) sonuçları: eğitim öncesi BHB testi sonuçlarında ilgili soruların öğrencilerin altı haftalık temel bilgisayar programlama eğitimi öncesi seviyelerini tespit etmek için BHB testi sonuçlarında elde edilen verilerden oluşmaktadır. Bu sınavın sonuçlarından elde edilen veriler çözümlenmiştir. Eğitim öncesi BHB testi toplam puanlarına göre alt ve üst grup olarak ayrılan grupların puan ortalamaları arasında üst grup lehine anlamlı fark vardır: ( $t(42) = -9,235$   $p= 0,000$ ), alt (ort.: 15,286; ss: 2,2393) ve üst (ort: 20,130; ss: 1,0998). Grupların BHB testi toplam puan ortalamalarının anlamlı olarak farklı olması sonucunda BHB testi toplam testinin ölçtüğü özellikler bağlamında öğrencileri ayırabildiği söylenebilir.

### **11-Algorithmik Düşünme (Son Test)**

Algorithmik Düşünme (Son Test) sonuçları: eğitim öncesi BHB testi sonuçlarında ilgili soruların öğrencilerin altı haftalık temel bilgisayar programlama eğitimi sonrası seviyelerini tespit etmek ve artış oranları ölçmek için BHB testi sonuçlarında elde edilen verilerden oluşmaktadır. Bu sınavın sonuçlarından elde edilen veriler çözümlenmiştir. Eğitim öncesi BHB Algorithmik Düşünme puanlarına göre alt ve üst grup olarak ayrılan grupların puan ortalamaları arasında üst grup lehine anlamlı fark vardır: ( $t(42) = -8,751$   $p= 0,000$ ), alt (ort.: 6,810; ss:1,4007) ve üst (ort: 9,739; ss: 0,7518). Grupların BHB Algorithmik Düşünme puan ortalamalarının anlamlı olarak farklı olması sonucunda BHB

Algoritmik Düşünme testinin ölçtüğü özellikler bağlamında öğrencileri ayırabildiği söylenebilir.

### **12-Ardışık ve Döngüsel Düşünme (Son Test)**

Ardışık ve Döngüsel Düşünme (Son Test) sonuçları: eğitim öncesi BHB testi sonuçlarında ilgili soruların öğrencilerin altı haftalık temel bilgisayar programlama eğitimi sonrası seviyelerini tespit etmek ve artış oranları ölçmek için BHB testi sonuçlarında elde edilen verilerden oluşmaktadır. Bu sınavın sonuçlarından elde edilen veriler çözümlenmiştir. Eğitim öncesi BHB Ardışık ve Döngüsel Düşünme puanlarına göre alt ve üst grup olarak ayrılan grupların puan ortalamaları arasında üst grup lehine anlamlı fark vardır: ( $t(42) = -8,573$   $p= 0,000$ ), alt (ort.: 12,381; ss:1,8568) ve üst (ort: 16,174; ss: 0,9841). Grupların BHB Ardışık ve Döngüsel Düşünme puanlarına puan ortalamalarının anlamlı olarak farklı olması sonucunda BHB Ardışık ve Döngüsel Düşünme testinin ölçtüğü özellikler bağlamında öğrencileri ayırabildiği söylenebilir.

### **13-Örüntü Tanıma (Son Test)**

Örüntü Tanıma (Son Test) sonuçları: eğitim öncesi BHB testi sonuçlarında ilgili soruların öğrencilerin altı haftalık temel bilgisayar programlama eğitimi sonrası seviyelerini tespit etmek ve artış oranları ölçmek için BHB testi sonuçlarında elde edilen verilerden oluşmaktadır. Bu sınavın sonuçlarından elde edilen veriler çözümlenmiştir. Eğitim öncesi BHB Örüntü Tanıma puanlarına göre alt ve üst grup olarak ayrılan grupların BHB Örüntü Tanıma puanlarına göre puan ortalamaları arasında üst grup lehine anlamlı fark vardır: ( $t(42) = -10,01$   $p= 0,000$ ), alt (ort.: 13,563; ss:2,2202) ve üst (ort: 18,821 ; ss: 1,2781). Grupların BHB Örüntü Tanıma puanlarına puan

ortalamlarının anlamlı olarak farklı olması sonucunda BHB Örüntü Tanıma testinin ölçtüğü özellikler bağlamında öğrencileri ayırabildiği söylenebilir.

BHB testinin eğitim sonrası uygulaması sonuçları alt ve üst grup ortalamaları arasında üst grup lehine 5,2589 fark vardır. Anlamlılık değeri  $0,000 < 0,0005$  dir. Her iki grup ortalamaları arasında anlamlı bir fark olduğu söylenebilir.

#### **14-Problem Çözme (Son Test)**

Problem Çözme (Son Test) sonuçları: eğitim öncesi BHB testi sonuçlarında ilgili soruların öğrencilerin altı haftalık temel bilgisayar programlama eğitimi sonrası seviyelerini tespit etmek ve artış oranları ölçmek için BHB testi sonuçlarında elde edilen verilerden oluşmaktadır. Bu sınavın sonuçlarından elde edilen veriler çözümlenmiştir. Eğitim öncesi BHB Problem Çözme puanlarına göre alt ve üst grup olarak ayrılan grupların puan ortalamaları arasında üst grup lehine anlamlı fark vardır:  $(t(42) = -9,292$   $p = 0,000)$ , alt (ort.: 14,375; ss:2,2767) ve üst (ort: 19,964; ss: 1,6884) grupların BHB Problem Çözme puanlarına puan ortalamalarının anlamlı olarak farklı olması sonucunda BHB Problem Çözme testinin ölçtüğü özellikler bağlamında öğrencileri ayırabildiği söylenebilir.

#### **15-Soyut Düşünme Soyutlama ve Genelleme (Son Test)**

Soyut Düşünme Soyutlama ve Genelleme (Son Test) sonuçları: eğitim öncesi BHB testi sonuçlarında ilgili soruların öğrencilerin altı haftalık temel bilgisayar programlama eğitimi sonrası seviyelerini tespit etmek ve artış oranları ölçmek için BHB testi sonuçlarında elde edilen verilerden oluşmaktadır. Bu sınavın sonuçlarından elde edilen

veriler çözümlenmiştir. Eğitim öncesi BHB Soyut Düşünme Soyutlama ve Genelleme puanlarına göre alt ve üst grup olarak ayrılan grupların puan ortalamaları arasında üst grup lehine anlamlı fark vardır: ( $t(42) = -9,34$   $p= 0,000$ ), alt (ort.: 16,474; ss:2,2452) ve üst (ort: 22,000; ss: 1,6833). Grupların BHB Soyut Düşünme Soyutlama ve Genelleme puan ortalamalarının anlamlı olarak farklı olması sonucunda BHB Soyut Düşünme Soyutlama ve Genelleme testinin ölçtüğü özellikler bağlamında öğrencileri ayırabildiği söylenebilir.

#### **16-Mantıksal çıkarım, Muhakeme, akıl yürütme (Son Test)**

Mantıksal Çıkarım, Muhakeme, Akıl Yürütme (Son Test) sonuçları: eğitim öncesi BHB testi sonuçlarında ilgili soruların öğrencilerin altı haftalık temel bilgisayar programlama eğitimi sonrası seviyelerini tespit etmek ve artış oranları ölçmek için BHB testi sonuçlarında elde edilen verilerden oluşmaktadır. Bu sınavın sonuçlarından elde edilen veriler çözümlenmiştir. Eğitim öncesi BHB Mantıksal Çıkarım, Muhakeme, Akıl Yürütme puanlarına göre alt ve üst grup olarak ayrılan grupların puan ortalamaları arasında üst grup lehine anlamlı fark vardır: ( $t(42) = -9,686$   $p= 0,000$ ), alt (ort.: 18,400; ss:2,3033) ve üst (ort: 24,208; ss: 1,6676) grupların BHB Mantıksal çıkarım, Muhakeme, akıl yürütme puan ortalamalarının anlamlı olarak farklı olması sonucunda BHB Mantıksal Çıkarım, Muhakeme, Akıl Yürütme testinin ölçtüğü özellikler bağlamında öğrencileri ayırabildiği söylenebilir.

### **17-Toplam Puan (Son Test)**

BHB Toplam puan (Son Test) sonuçları: eğitim öncesi BHB testi sonuçlarında ilgili soruların öğrencilerin altı haftalık temel bilgisayar programlama eğitimi sonrası seviyelerini tespit etmek ve artış oranları ölçmek için BHB testi sonuçlarında elde edilen verilerden oluşmaktadır. Bu sınavın sonuçlarından elde edilen veriler çözümlenmiştir. Eğitim öncesi BHB Toplam puanlarına göre alt ve üst grup olarak ayrılan grupların puan ortalamaları arasında üst grup lehine anlamlı fark vardır: ( $t(42) = -9,686$   $p = 0,000$ ), alt (ort.: 18,400; ss:2,3033) ve üst (ort: 24,208; ss: 1,6676). Grupların BHB toplam puan ortalamalarının anlamlı olarak farklı olması sonucunda BHB testinin ölçtüğü özellikler bağlamında öğrencileri ayırabildiği söylenebilir.

DeneySEL eğitim çalışması sürecinde elde edilen 44 öğrenci için 17 farklı sonuç listesi değerlendirmeleri sonucu üst ve alt gruplar arasında anlamlı bir fark görülmüştür. BHB testi için hazırlanan ve yetkinlikleri ölçümlemek için oluşturulan test sorularının ayırıcılık bakımından öğrencileri ayırabildiği sonucuna ulaşabiliriz.

### **7.2.5 Bulgu 5: BHB Ön Test ve Son Test Analizleri**

SPSS kullanılarak BHB Ön Test ve Son Test karşılaştırmalı analizleri yapılarak Tablo 7.8. oluşturuldu.

Tablo 7.8. Deney t-testi Sonuçları

		Ort.	N	Std. Sapma	Std. Hata	t	sd	Sig. (2-yönlü)
1	Programlamaya İlişkin Özyeterlilik Ölçeği (Son Test)	41,727	44	13,6814	2,0625	3,899	43	0,000
	Programlamaya İlişkin Özyeterlilik Ölçeği (Ön Test)	33,614	44	13,1259	1,9788			
2	Algoritmik Düşünme (Ön Test)	5,5	44	1,2484	0,1881	-10,843	43	0,000
	Algoritmik Düşünme (Son Test)	8,341	44	1,8419	0,2777			
3	Ardışık ve Döngüsel Düşünme (Ön Test)	11,432	44	1,8602	0,2804	-8,911	43	0,000
	Ardışık ve Döngüsel Düşünme (Son Test)	14,364	44	2,4024	0,3622			
4	Örüntü Tanıma (Ön Test)	13,364	44	2,2214	0,3349	-9,169	43	0,000
	Örüntü Tanıma (Son Test)	16,909	44	3,0486	0,4596			
5	Problem Çözme (Ön Test)	14,227	44	2,4952	0,3762	-9,255	43	0,000
	Problem Çözme (Son Test)	17,932	44	3,3159	0,4999			
6	Soyut Düşünme Soyutlama ve Genelleme (Ön Test)	15,795	44	2,7667	0,4171	-8,698	43	0,000
	Soyut Düşünme Soyutlama ve Genelleme (Son Test)	19,614	44	3,3703	0,5081			
7	Mantıksal Çıkarım, Muhakeme, Akıl Yürütme (Ön Test)	17,818	44	2,9905	0,4508	-8,321	43	0,000
	Mantıksal Çıkarım, Muhakeme, Akıl Yürütme (Son Test)	21,568	44	3,52	0,5307			
8	BHB Toplam Puan (Ön Test)	17,818	44	2,9905	0,4508	-8,321	43	0,000
	BHB Toplam Puan (Son Test)	21,568	44	3,52	0,5307			

Sigma (2-yönlü) 0,05 den küçük olması karşılaştırılan çiftler arasında anlamlı bir fark olduğunu göstermektedir. Ön test ve son test ile ilgili elde edilen sonuçlar son test lehine farklılık göstermiştir. Bu sonuç geliştirilen eğitim sisteminin kendisinden beklenen yararı sağladığını göstermektedir.

### 1- Programlamaya İlişkin Öz Yeterlilik Ön ve Son Test Karşılaştırması

Yapılan deneysel çalışma Programlamaya İlişkin Öz Yeterlilik Ölçeği boyutuyla ilgili maddelerinden eğitim başında ve eğitim sonunda elde edilen puan ortalamaları



karşılaştırılmıştır. Öğrencilerin Programlamaya İlişkin Öz Yeterlilik Ölçeği ön ve son testlerinden elde ettikleri puanların ortalamaları arasında anlamlı bir fark bulunmuştur: ( $t(43) = 3,899$   $p= 0,000$ ). Eğitim sonrasında, öğrencilerin eğitim öncesine göre Programlamaya İlişkin Öz Yeterlilik seviyelerinde anlamlı bir artış vardır. Öğrenciler; eğitim sonrasında, öncesine göre Programlamaya İlişkin Öz Yeterlilik Ölçeğinde anlamlı bir artış göstermiştir. Verilen eğitimin öğrencilerin programlamaya ilişkin özgüvenlerini arttırdığı söylenebilir.

## **2- Algoritmik Düşünme Ön ve Son Test Karşılaştırması**

BHB test sonuçlarına göre Algoritmik Düşünme boyutuyla ilgili maddelerinden ön testte ve son testte elde edilen puanların ortalamaları karşılaştırılmıştır. Öğrencilerin Algoritmik Düşünme ön ve son testlerinden elde ettikleri puanların ortalamaları arasında anlamlı bir fark bulunmuştur ( $t(43) = -10,84$   $p= 0,000$ ). Eğitim sonrasında öğrencilerin eğitim öncesine göre Algoritmik Düşünme yetkinliklerinde anlamlı bir artış vardır .

## **3-Ardışık ve Döngüsel Düşünme Ön ve Son Test Karşılaştırması**

BHB test sonuçlarına göre Ardışık ve Döngüsel Düşünme boyutuyla ilgili maddelerinden ön testte ve son testte elde edilen puanların ortalamaları karşılaştırılmıştır. Öğrencilerin Ardışık ve Döngüsel Düşünme ön ve son testlerinden elde ettikleri puanların ortalamaları arasında anlamlı bir fark bulunmuştur: ( $t(43) = -8,911$   $p= 0,000$ ). Eğitim sonrasında öğrencilerin eğitim öncesine göre Ardışık ve Döngüsel Düşünme yetkinliklerinde anlamlı bir artış vardır .

#### **4- Örüntü Tanıma Ön ve Son Test Karşılaştırması**

BHB test sonuçlarına göre “Örüntü Tanıma” boyutuyla ilgili maddelerinden ön testte ve son testte elde edilen puanların ortalamaları karşılaştırılmıştır. Öğrencilerin “Örüntü Tanıma” ön ve son testlerinden elde ettikleri puanların ortalamaları arasında anlamlı bir fark bulunmuştur: ( $t(43) = -9,169$   $p= 0,000$ ). Öğrencilerin, eğitim sonrasında, eğitim öncesine göre Örüntü Tanıma yetkinliğinde anlamlı bir artış vardır

#### **5- Problem Çözme Ön ve Son Test Karşılaştırması**

BHB test sonuçlarına göre “Problem Çözme” boyutuyla ilgili maddelerinden ön testte ve son testte elde edilen puanların ortalamaları karşılaştırılmıştır. Öğrencilerin “Problem Çözme” ön ve son testlerinden elde ettikleri puanların ortalamaları arasında anlamlı bir fark bulunmuştur ( $t(43) = -9,255$   $p= 0,000$ ). Öğrencilerin, eğitim sonrasında eğitim öncesine göre Problem Çözme yetkinliğinde anlamlı bir artış vardır.

#### **6- Soyut Düşünme Soyutlama ve Genelleme Ön ve Son Test Karşılaştırması.**

BHB test sonuçlarına göre “Soyut Düşünme Soyutlama ve Genelleme” boyutuyla ilgili maddelerinden ön testte ve son testte elde edilen puanların ortalamaları karşılaştırılmıştır. Öğrencilerin “Soyut Düşünme Soyutlama ve Genelleme” ön ve son testlerinden elde ettikleri puanların ortalamaları arasında anlamlı bir fark bulunmuştur ( $t(43) = -8,698$   $p= 0,000$ ). Öğrencilerin, eğitim sonrasında eğitim öncesine göre “Soyut Düşünme Soyutlama ve Genelleme” yetkinliğinde anlamlı bir artış vardır

### **7- Mantıksal çıkarım, Muhakeme, akıl yürütme Ön ve Son Test Karşılaştırması.**

BHB test sonuçlarına göre “Mantıksal çıkarım, Muhakeme, akıl yürütme” boyutuyla ilgili maddelerinden ön testte ve son testte elde edilen puanların ortalamaları karşılaştırılmıştır. Öğrencilerin “Mantıksal çıkarım, Muhakeme, akıl yürütme” ön ve son testlerinden elde ettikleri puanların ortalamaları arasında anlamlı bir fark bulunmuştur ( $t(43) = -8,321$   $p = 0,000$ ). Öğrencilerin, eğitim sonrasında eğitim öncesine göre Mantıksal çıkarım, Muhakeme, akıl yürütme” yetkinliklerinde anlamlı bir artış vardır

### **8- BHB Toplam puan Ön ve Son Test Karşılaştırması**

BHB test sonuçlarına göre “Toplam Puan” alt boyutuyla ilgili maddelerinden ön testte ve son testte elde edilen puanların ortalamaları karşılaştırılmıştır. Öğrencilerin “Toplam Puan” ön ve son testlerinden elde ettikleri puanların ortalamaları arasında anlamlı bir fark bulunmuştur ( $t(43) = -8,321$   $p = 0,000$ ). Öğrencilerin, eğitim sonrasında eğitim öncesine göre Toplam Puan Toplam Puan yetkinliklerinde anlamlı bir artış vardır .

Eğitim Sonrasında BHB ön ve son test sonuçlarının farkları yetkinlik hiyerarşisine uygun artan bir görüntü sergilemiştir

Tablo 7.9. Eğitim Sonrası Yetkinlik Artış Oranları.

YETKİNLİKLERE GÖRE PUANLAR - BHB FARK DEĞİŞİMLER(%)						
Algoritmik Düşünme	Ardışık ve Döngüsel Düşünme	Örüntü Tanıma	Problem Çözme	Soyut Düşünme Soyutlama ve Genelleme	Mantıksal çıkarım, Muhakeme, akıl yürütme	Toplam Yetkinlik
29,53	24,22	24,62	24,25	21,27	19,67	19,67

### 7.2.6 Bulgu 6: Çalışma Sonuçları Yetkinlikler Arası Korelasyonlar

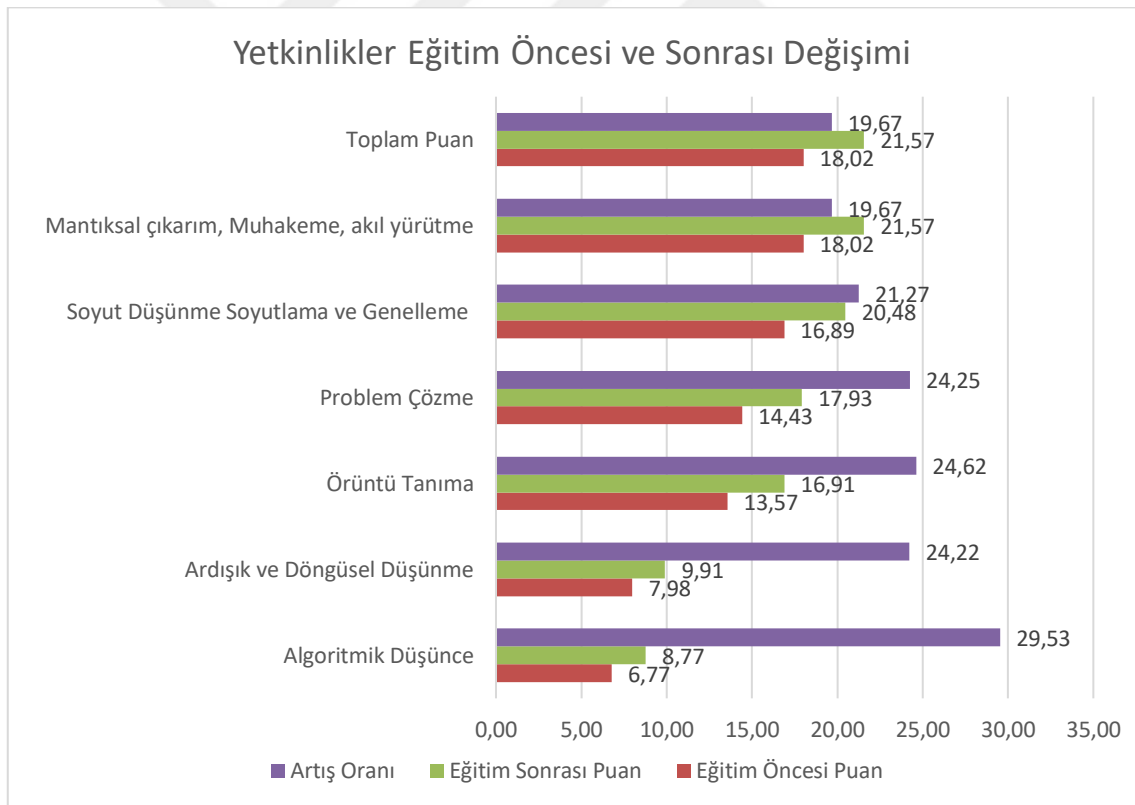
SPSS analizlerinde çalışmanın sonuçları üzerinde yapılan değerlendirmelerde ilgili 6 alt boyut (yetkinlikler) arası korelasyonlar Tablo 7.10. da görüleceği üzere her bir alt boyutun diğer boyutlarla istatistiksel olarak anlamlı ilişkisi olduğu görülmektedir. En yüksek korelasyon 0.988\*\* ile “Mantıksal Çıkarım, Muhakeme, Akıl Yürütme” ve “Soyut Düşünme Soyutlama ve Genelleme” yetkinliği arasında görülmektedir. En düşük korelasyon 0.758\*\* ile “Algoritmik Düşünme” ve “Problem Çözme” yetkinlikleri arasında olduğu görülmüştür.

Tablo 7.10. Çalışma Sonuçları Yetkinlikler Arası Korelasyonlar

	Algoritmik Düşünme	Ardışık ve Döngüsel Düşünme	Örüntü Tanıma	Problem Çözme	Soyut Düşünme Soyutlama ve Genelleme	Mantıksal Çıkarım, Muhakeme, Akıl Yürütme
1	1	,836**	,788**	,791**	,758**	,766**
2	,836**	1	,940**	,915**	,890**	,892**
3	,788**	,940**	1	,954**	,928**	,920**
4	,791**	,915**	,954**	1	,977**	,966**
5	,758**	,890**	,928**	,977**	1	,988**
6	,766**	,892**	,920**	,966**	,988**	1

### 7.2.7 Eğitim Öncesi ve Sonrası Yetkinlik Değişimleri

Algoritmik düşünmenin %29,53 artarken mantıksal çıkarım, muhakeme ve akıl yürütme yetkinliği %19,67 artmaktadır. Bu sonuçlara göre önerilen eğitim metodunun algoritmik düşünmeyi yetkinliğine daha fazla katkı sağladığı sonucu ortaya çıkar. Bu bulgunun önemli bir sonucu da “Mantıksal Çıkarım, Muhakeme ve Akıl Yürütme Yetkinliğinin daha az yükselmesini zaten gelişmiş olmasına da bağlanabilir. Bu durumda deneklerimizde bulunan yetkinlik seviyelerinin yüksek olanlarının daha düşük artış göstermesi doğaldır.



Şekil 7.6. Eğitim Sonrası Yetkinlik Artış Oranları

## 8 SONUÇLAR

Günümüzde bilgi ve iletişim teknolojilerinde yaşanan hızlı gelişmeler, içinden geçtiğimiz süreci bilgi ve iletişim çağı olarak tanımlamamızı zorunlu kılmıştır. En önemli değer olan bilgiye ulaşmak artık 10 yıl öncesine hatta 5 yıl öncesine nazaran çok kolaylaşmıştır. Artık bilmek veya biliyor olmak önemini yitirmekte ve bilinenin nasıl yeni bilgiler inşa edeceği ve bilgiden nasıl fayda ve sonuç üreteceğimiz önem kazanmaktadır. Ulaşabileceğimiz bilgi artık faydadan çok karışıklığa neden olacak kadar çoktur. Veriden bilgi çıkarma sürecini oluşturan eylemler tezde yer alan altı bilişsel yetkinliğinin kapsamına girer. Bilgisayar programlama eğitimi sözü edilen yetkinlikler ile paralel bir başarı eğrisi oluşturması nedeni ile eğitim sonrası bu yetkinliklerin seviyesi de artmaktadır. Sonuç olarak bilgisayar programlama eğitiminin salt program geliştirme hedefinin ötesinde gelecekte her seviyede eğitimin zorunlu bir ögesi olacaktır.

Bu çalışmada elde edilen verilerin analiz edildiği istatistiksel testlerin toplam istatistiksel hata konusundaki zafiyetleri dikkate alınmalı ve çalışma büyük gruplarla yinelenerek teyit edilmelidir. Ancak küçük bir grupta yapılan öğretim çalışması ve ilgili ölçümler programlama öğretimine dönük olumlu ipuçları sunmuştur. Programcılık konusundaki uzmanlarla geliştirilen ölçekle elde edilen altı boyutun kapsamındaki bilişsel yapıların burada önerilen bilgisayar öğretimi etkinlikleriyle geliştirilebileceği gösterilmiştir. Tespit edilen altı alt boyutun incelenmesi sonraki çalışmalarda daha derinlemesine yapılmalı ve alt boyutlar arası etkileşimler farklı istatistiklerle yeniden incelenmelidir. Programlama eğitiminde kullanılan etkinliklerin çeşitleri ve sayısı da

artırılarak farklı öğrencilere dönük etkinlik setleri üretilerek test edilmelidir. Çalışmanın kavramsal, uygulamalı ve istatistiksel sınırlılıklarına rağmen, Tez çalışmasında aşağıda sıralanan beş ayrı hipotezi destekleyen sonuçlar bulunmuştur.

**Hipotez 1:** Belirlenen altı bilişsel yetkinlik seviyesi ölçümlenebilir.

Önerilen altı bilişsel yetkinlik seviyesi eğitime başlayan ve bitiren her öğrenci için ölçümlenebilir. Geliştirilen BHB testi bu yetkinliklerin ölçümlenebildiğini destekleyen sonuçlar ortaya koymuştur.

**Hipotez 2:** Belirlenen altı bilişsel yetkinlik bilgisayar programlama eğitimi başarımları ile doğrudan ilintilidir.

Yapılan eğitim çalışmasında altı bilişsel yetkinlik eğitim öncesi ve eğitim sonrası ölçümlerde öğrenciler için anlamlı fark göstermiştir. Bilişsel yetkinliklerin seviyesi bilgisayar programlama eğitim sürecine ve eğitim sonunda elde edilen başarımları ile doğrudan ilişkilidir ve yetkinlik seviyeleri alt ve üst gruplar başarımları ile anlamlı bir fark göstermektedir.

**Hipotez 3:** Altı bilişsel yetkinlik önerilen eğitim metodu ile geliştirilebilir.

Önerilen bilgisayar programlama eğitimi metodu ile programlama eğitimi almış kişilerde belirlenen altı bilişsel yetkinlik eğitim almadan öncesine göre anlamlı bir farklılık göstermektedir. Tüm öğrenciler belirlenen altı bilişsel yetkinlik düzeyinde artış vardır. Yapılan eğitim çalışmasında altı bilişsel yetkinlik eğitim öncesi ve eğitim sonrası ölçümlerde öğrenciler için anlamlı fark göstermiştir.

**Hipotez 4:** Önerilen eğitim metodu bilgisayar programlama öz yeterliliğini arttırır.

Geliştirilen izlence ile yapılan programlama etkinliklerine katılan öğrencilerin eğitim sonrası ölçümleri eğitim öncesine göre “Bilgisayar programlama öz yeterlilik” %27.19 oranında artış gözlenmiştir. Önerdiğimiz bilgisayar programlama eğitimi metodu ile programlama eğitimi almış öğrenciler öz yeterliliğini arttığı deney sonucu tarafından desteklenmektedir.

**Hipotez 5:** Önerilen bilgisayar programlama eğitimi metodu ile öğrenciler temel bilgisayar programlamayı etkin bir şekilde öğrenir.

Önerilen eğitim metodu ile gerçekleştirilen, yetkinlik temelli eğitim platformunun başarıya ulaşp ulaşmadığını belirlemek için “Eğitim sonrası programlama değerlendirme (Bilgi Ölçme) Sınavı” sonuçlarına göre öğrenciler ortalama 10 üzerinden 8,25 puan ile başarılı olmuşlardır. Standart sapma 1,67 olarak gerçekleşmiştir. Bir öğrenci 10 üzerinden 3 puan alarak başarısız olmuştur (Tablo 7.6.). Eğitim sonunda gerçekleştirilen “Eğitim sonrası programlama değerlendirme sınavı” sonuçları alt ve üst grup öğrenciler arasında anlamlı fark olduğunu göstermektedir. Önerdiğimiz bilgisayar programlama eğitimi metodu ile programlama eğitimi almış öğrencilerin, etkin bilgisayar programlama öğrendikleri deney sonucunda desteklenmektedir.

**Sonuç:** Deney sonuçlarının analizi hipotezleri doğrulamaktadır.

Bilişsel yetkinliklerin seviyesi bilgisayar programlama eğitim sürecine ve eğitim sonunda elde edilen başarı ile doğrudan ilişkilidir ve yetkinlik seviyeleri alt ve üst gruplar başarımla anlamlı bir fark göstermektedir. BHB testimizin altı bilişsel



yetkinlik altındaki değerlendirme ve analizlerde iyi not alan öğrenci ile kötü not alan öğrencileri ayırt edebildiğine dönük bulgular elde edilmiştir.

Solmaz çalışmasında programlama ders başarısı değişkenine ait son test verileri bağımsız örneklem için t-testi kullanılarak analiz etmiş öğrencilerin başarı sınavı puanlarının deney ve kontrol grubuna göre anlamlı bir farklılık göstermediği belirlenmiştir. Alice programının PHP programlama dili öğretiminde ders başarısı üzerinde çok fazla etkili olmadığı, ancak Java, C++ gibi nesne yönelimli dillerin öğretiminde ders başarısı üzerinde olumlu bir etkiye sahip olduğu ortaya çıkmıştır (Solmaz, 2014). Bizim çalışmamızda hem yetkinliklerin seviyesi artmış hem de programlama başarı puanları yüksek bulunmuştur. Bu bulgular ele aldığımız altı bilişsel yapı programlama bağlamında geliştirildiğinde programlama bilgisinin de geliştirileceğine işaret etmektedir.

Önerilen bilgisayar programlama eğitimi metodu ile programlama eğitimi almış kişilerde belirlenen altı bilişsel yetkinlik düzeyleri eğitim almadan öncesine göre anlamlı bir farklılık göstermektedir. Geliştirilen eğitim metodu ile programlama etkinliklerine katılan bu öğrencilerin bilgisayar programlama öz yeterlilik seviyeleri %27.19 artış gözlenmiştir. Altun ve arkadaşlarının çalışmasında ortalama özyeterlilik seviyesi 42 olarak ortaya çıkmış olup, çalışma grubumuzla uyum sağlamaktadır.

Diğer bir çalışmada Akçay araştırmasının sonucunda öğrencilerin programlamaya ilişkin öz yeterliklerinin orta düzeyde olduğu, sorgulama becerilerine sahip olduğu ve problem çözme becerilerinin yüksek olduğu tespit edilmiştir. Bununla birlikte programlamaya ilişkin öz yeterlik düzeyleri ile problem çözme becerileri arasında

anlamli bir farklılık elde edilemezken, programlamaya ilişkin öz yeterlik düzeyleri ile sorgulama becerileri arasında anlamlı bir farklılık bulunmuştur (Akçay, 2015). Çalışmamızın sonuçları bakımından uyum göstermektedir.

Önerilen eğitim metodu bilgisayar programlama eğitimi verme görevini başarı ile tamamladığını deney sonuçları göstermektedir. Eğitim sonrası Programlama Değerlendirme Sınavı sonuçları ise görevin başarı düzeyini ortaya çıkarmaktadır.

Altun ve arkadaşlarının çalışması sonucunda ortaya çıkan: zihinsel döndürme becerisinin diğer tüm değişkenlerden oldukça farklı olduğu bulunmuştur. Hem yüksek hem de düşük gruba ilişkin ortaya çıkan iki profilin zihinsel döndürme becerisi bakımından oldukça farklı olduğu dikkat çekmiştir. Sonucu bizim çalışmamızda Algoritmik düşünme, Ardışık ve Döngüsel Düşünme bu parametrelere yakın olduğundan çalışmamızın sonuçları bu çalışmayı destekleme yönündedir.

SPSS analizlerinde çalışmanın sonuçları üzerinde yapılan değerlendirmelerde ilgili 6 alt boyut (yetkinlikler) arası korelasyonlar Tablo 7.10. da görüleceği üzere her bir alt boyutun diğer boyutlarla istatistiksel olarak anlamlı ilişkisi olduğu görülmektedir.

Tasarlanan ve geliştirilen bilgisayar programlama öğrenim setinin mühendislik çalışması; en az doksan adam/gün sürmüştür. Merkezi sistem Java teknolojileri, JSF (JavaServer Faces), Hibernate ve MySql veri tabanı kullanılarak geliştirilmiştir. Kullanıcı arabirimi (Uygulama geliştirme modülü) tüm geçerli sistemlerde (bilgisayar, tablet, akıllı telefon vb.) çalışabilecek şekilde HTML/javascript ile geliştirilmiştir. Tasarlanan ve programlanan sistem, öğrencilerin kullanımına açılmıştır. Sistem

yazılımı geliştirme sırasında uzun süreli testler yapılarak hatalardan arındırılmıştır. Öğrencilere açık kullanım sırasında ise her hangi bir hata ile karşılaşılmamıştır.

Bu çalışmanın genişletilmesi; sistemin çok sayıda kullanıcının erişebileceği bir ortam olarak kapasitesinin arttırılması ve topluma açılması ile mümkündür. Elde edilecek faydanın izlenerek ölçülmesi bir ilerideki çalışmalar için itici güç olacaktır.

Önerilen eğitim metodolojisinin adaptif öğrenme ve yapay zekâ kullanımı gibi desteklerle de geliştirilmesinin önü açıktır. Yeteneklerin nasıl ölçüleceği belirlenmiş olması, eğitim içeriğinin bunlara göre ağırlıklandırılması önemli bir aşamadır.

## 9 KAYNAKLAR

**Ağaç, G.**, (2013), 8. Sınıf Öğrencilerinin Matematiğe Yönelik; Problem Çözme, Soyut Düşünme, İnanç, Öğrenilmiş Çaresizlik Puanlarının Bazı Değişkenler Açısından İncelenmesi ve Aralarındaki İlişki, Yüksek Lisans Tezi, Sakarya Üniversitesi, Sakarya, s-88

**Akbulut, Y.**, (2016), Sosyal Bilimlerde SPSS Uygulamaları İdeal Kültür Yayıncılık, İstanbul, s-107

**Akçay, A.**, (2015), Programlama becerisi öz yeterliğinin problem çözme ve sorgulama becerileri bağlamında incelenmesi, Yüksek lisans tezi, Necmettin Erbakan Üniversitesi, Konya.

**Akpınar, Y.**, (2005), Bilgisayar Destekli Eğitimde Uygulamalar, 2.Baskı, Anı Yayıncılık, Ankara.

**Akpınar, Y., Altun, A.**, (2014), Bilgi Toplumu Okullarında Programlama Eğitimi Gereksinimi, Elementary Education Online, 13(1), dy: 1-4.

**Alpaydın, E.**, (2010), Introduction to Machine Learning, Massachusetts Institute of Technology Press, USA.

**Alpaydın, E.**, (2011), Yapay Öğrenme, Birinci Baskı s-1,111 Boğaziçi Üniversitesi Yayınevi, İstanbul.

**Altun, A., Mazman, S., G.**, (2012), Programlamaya İlişkin Öz Yeterlilik Algısı Ölçeğinin Türkçe Formunun Geçerlilik ve Güvenirlik Çalışması, Eğitimde ve Psikolojide Ölçme ve Değerlendirme Dergisi, Kış 2012, 3(2), 297- 308

- Altun, A., Mazman, S. G., Demir, Ö.,** (2013), Üniversite Öğrencilerinin Programlama Performansını Etkileyen Faktörlerin Bilişsel Tabanlı Bireysel Farklılıklar Temelinde Modellenmesi, Aralık 2013 ANKARA, Tübitak 112K615
- Ayhan İ.,** (2007), Öğrenme. Bilim Teknik, Temmuz, (2-15) Ankara.
- Budak, S.,** (2000), “Psikoloji Sözlüğü.”, Bilim ve Sanat Yayınları, Ankara.
- Coşar, M.,** (2013), Problem Temelli Öğrenme Ortamında Bilgisayar Programlama Çalışmalarının Akademik Başarı, Eleştirel Düşünme Eğilimi Ve Bilgisayara Yönelik Tutuma Etkileri, Doktora Tezi, Gazi Üniversitesi, Ankara.
- Çetin, E.,** (2012), Bilgisayar programlama eğitiminin çocukların problem çözme becerileri üzerine etkisi, Yüksek Lisans tezi(349116), Gazi Üniversitesi, Ankara.
- Çölkesen, R.,** (2002), Veri Yapıları ve Algoritmalar, 1.Baskı, Papatya Yayıncılık, İstanbul.
- Demir, F.,** (2015), Programlama Öğretiminde Eğitsel Programlama Dilinin Farklı Kullanımlarının Programlama Başarısı Ve Kaygısına Etkisi, Doktora Tezi, Atatürk Üniversitesi, Erzurum, s-40
- Demirer, B.,** (2012), Çok kriterli karar verme sürecinde dinamik programlama Uygulaması, Yüksek Lisans Tezi, Dokuz Eylül Üniversitesi, İzmir.
- Dilci, D.,** (2014), Gelişim Psikolojisi, İdeal Kültür Yayıncılık, İstanbul, s. 99.
- Ekici, G.,** (2002), Gregorc Öğrenme Stili Ölçeği, Eğitim ve Bilim, Cilt 27, Sayı 123 (42-47) Education and Science.
- Erdoğan, B.,** (2005), Programlama Başarısı İle Akademik Başarı, Genel Yetenek, Bilgisayara Karşı Tutum, Cinsiyet ve Lise Türü Arasındaki İlişkilerin İncelenmesi, Yüksek lisans tezi, Marmara Üniversitesi, İstanbul.

- Eripek, S.**, (1998), Eğitim Biliminde Yenilikler, İlköğretim Çağı Çocuklarının Bilişsel, Bedensel ve Kişilik Özellikleri S.96 Yazar Prof. Dr. Süleyman ERİPEK, T.C. Anadolu Üniversitesi Yayınları No: 1016 ISBN975-492-765-0, s-23
- Ersoy, H., Madran, R. O., Gülbahar, Y.**, (2011), Programlama Dilleri Öğretimine Bir Model Önerisi: Robot Programlama, Akademik Bilişim'11 - XIII. Akademik Bilişim Konferansı, İnönü Üniversitesi, Malatya.
- Flanagan, D.**, (2004) Java Uygulamaları, Birinci Baskı, Pusula yayıncılık, İstanbul.
- Futschek, G.**, (2006), "Algorithmic Thinking: The Key for Understanding Computer Science", Vienna University of Technology, s-73.  
([http://link.springer.com/chapter/10.1007/11915355\\_15#page-1](http://link.springer.com/chapter/10.1007/11915355_15#page-1))
- Gültekin, K.**, (2006), Çokluortamın Bilgisayar Programlama Başarısı Üzerine Etkisi, Yüksek Lisans Tezi, Hacettepe Üniversitesi, Ankara.
- İnanç, B., Y, Bilgin, M. ve Atıcı, M., K.**, (2012), Gelişim psikolojisi: Çocuk ve ergen gelişimi, Pagem Akademi, Ankara.
- Koçak, N., Pınarcık, Ö., Ergin, B.**, (2015), Okul Öncesi Dönem Çocuklarının Bilişsel Gelişim Özellikleri İle Sosyal Becerileri Arasındaki İlişkinin İncelenmesi(Konya Örneği) S3(1), 21-29 Asya Öğretim Dergisi [www.e-aji.com](http://www.e-aji.com)  
[Asian Journal of Instruction] e-ISSN:2148-2659
- Onur, B.**, (1986), Gelişim Psikolojisi, 1. Baskı, V Yayınları Ankara, s-10.
- Pappert, S.**, (1985), Mindstorms Children Computers and Powerful Ideas, Harvester Press, Brighton Sussex, Great Britain.
- Polya, G.**, (1990), How to Solve It, A New Aspect of Mathematical Method s-xxxi Princeton University Press, USA.

**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) 365-379.

**Sağır, C.,** (2006), Karar verme sürecini etkileyen faktörler ve karar verme sürecinde etiğin önemi: uygulamalı bir araştırma, Yüksek Lisans Tezi, Trakya Üniversitesi, Edirne, s-22.

**Scott, P., Nicolson, R.,** (1991), *Cognitive Science Project in Prolog Lawrence Erlbaum Associates East Sussex, Hove, Sussex, UK.*

**Sevim, O.,** (2015), Kuramdan Uygulamaya Bilgisayar Destekli Öğretim Materyali Geliştirme. Nobel akademik Yayıncılık, Ankara, s-81-82

**Sönmez, V.,** (2011), Öğretim İlke ve Yöntemleri, 6.Baskı, Anı Yayıncılık, Ankara.

**Teymur, C.,** (2008), Kural Tabanlı Karar Verme Sistemlerinde Belirsizliğin Modellenmesi, Doktora Tezi, İstanbul Teknik Üniversitesi, İstanbul.

**Topses, G.,** (2006), Gelişim ve Öğrenme Psikolojisi, Nobel Yayın Dağıtım, 2.Baskı, İstanbul, s-109-231

**Tutkun, Ö., F., Okay, S., Şahin, E.,** (2012), Öğretim Süreçleri ve Öğrenme Psikolojisi(Mark, P.Driscoll), Anı Yayıncılık, Ankara.

**Yeşilyaprak ve diğerleri** (2015), Eğitim Psikolojisi: Gelişim, Öğrenme, Öğretim Pagem Yayıncılık, Ankara.

**Yılmaz, S.,** (2005), Bilgi İşleme Modeline Dayalı Bir Dersin Fen Bilgisi Öğretmen Adaylarının Manyetizma Konusundaki Başarılarına Etkisi. Hacettepe Üniversitesi Eğitim Fakültesi Dergisi, 28, 236-243, Ankara.

**Yılmaz, F.**, (2013), Meslek Yüksek Okulu Öğrencilerinin Programlama Başarısını Etkileyen Faktörlerin İncelenmesi, Yüksek Lisans Tezi, Gazi Üniversitesi, Ankara.





## 10 EKLER

### EK-1

#### Boğaziçi Üniversitesi Eğitim Fakültesi

#### Bilgisayar Öğretim Teknolojileri Eğitimi Bölümü

#### Bilgisayar Programlama Eğitimi Çalışması

12.02.2016-11.03.2016

Öğrenci No	
Adı Soyadı	
Mezun olduğunuz lise	
Lisede bilgisayar programlama öğrendiniz mi?	
Yaşınız	
Cinsiyet	
Sınıf	

**Değerlendirmenizi 1-5 arası puan vererek yapınız**

AÇIKLAMA	PUAN
Genel olarak Bilgisayar programlamaya ne kadar meraklısınız?	
Genel olarak bilgisayar oyunları oynamaya ne kadar meraklısınız?	
Bilgisayar Programlama bilginize kaç puan verirsiniz?	
Üç haftalık Eğitimi beğendiniz mi?	
Eğitim öncesi düşünceniz. Bilgisayar Programcısı olmayı düşünüyor musunuz?	
Eğitim sonrası düşünceniz. Bilgisayar Programcısı olmayı düşünüyor musunuz?	
Eğitimin size katkısı ne kadar olduğunu düşünmektесiniz.	
Eğitim aracı ile yaptığınız eğitim katkısının ne kadar olduğunu düşünmektесiniz.	
Eğitim aracını beğendiniz mi? Kaç puan verirsiniz?	
Eğitmeni beğendiniz mi? Kaç puan verirsiniz?	
Yardım dosyaları ve konu anlatımı yapan video ders materyali ile eğitmen	
Eğitimin ileri düzeyini almak ister misiniz?	
Algoritma sorularını amacına yönelik buldunuz mu?	
Öğretmen olsaydınız bu eğitimi öğrencilerinize uygularımsınız?	
Genel olarak bu çalışmanın zorluk derecesi size göre ne düzeydedir?	
Genel olarak bu çalışmaya kaç puan verirsiniz?	
Size göre bu araç ile programlama öğrenilebilir mi?	
Bu eğitimden aldığınız sonucu(gelişim) puan olarak değerlendiriniz.	

### EK-2 Geliştirilen Temel Bilgisayar Programlama Eğitimi Metodunun Görevleri

GÖREV NO	BÖLÜM NO	GÖREVLER
1	1	Çıkış ekranına "Hello World" yazdırılması işleminin algoritmasını tasarlayınız ve blok sistemi ile programını geliştiriniz. Otomatik olarak oluşacak kodları inceleyiniz.,
2	2	Verilen (girilen) bir değerın çıkış ekranına yazdırılması işleminin algoritmasını tasarlayınız ve blok sistemi ile programını geliştiriniz. Otomatik olarak oluşacak kodları inceleyiniz.,
3	3	Verilen (girilen) bir değerın iki katının ve karesinin çıkış ekranına yazdırılması işleminin algoritmasını tasarlayınız ve blok sistemi ile programını geliştiriniz. Otomatik olarak oluşacak kodları inceleyiniz.
4	4	Verilen (girilen) bir değerın bir küçük sayısının çıkış ekranına yazdırılması işleminin algoritmasını tasarlayınız ve blok sistemi ile programını geliştiriniz. Otomatik olarak oluşacak kodları inceleyiniz.
5	5	Verilen (girilen) değerın karesi ile arasındaki farkı çıkış ekranına yazdırılması işleminin algoritmasını tasarlayınız ve blok sistemi ile programını geliştiriniz. Otomatik olarak oluşacak kodları inceleyiniz.
6	1	Verilen (girilen) bir değerın soyut bir değişkene atanması ve saklanması ardından çıkış ekranına yazdırılması işleminin algoritmasını tasarlayınız ve blok sistemi ile programını geliştiriniz. Otomatik olarak oluşacak kodları inceleyiniz.,
7	2	Verilen (girilen) iki pozitif tamsayı değerın iki değişken tanımlanarak atama işlemlerinin yapılması ve çıkış ekranına yazdırılması ardından iki değişken değerlerini yer değiştirime yapılarak tekrar çıkış ekranına yazdırılması işleminin algoritmasını tasarlayınız ve blok

		sistemi ile programını geliştiriniz. Otomatik olarak oluşacak kodları inceleyiniz.,
8	3	Verilen (girilen) iki pozitif integer değerini iki değişken tanımlanarak atama işlemlerinin yapılması ve çıkış ekranına yazdırılması ardından iki değişken değerlerini aritmetik ortalamasını bularak tekrar çıkış ekranına yazdırılması işleminin algoritmasını tasarlayınız ve blok sistemi ile programını geliştiriniz. Otomatik olarak oluşacak kodları inceleyiniz.,
9	4	Verilen (girilen) iki tamsayı ve bir operatör (+ - * /) yapılacak işlemin sonucunu çıkış ekranına yazdırılması işleminin algoritmasını tasarlayınız ve blok sistemi ile programını geliştiriniz. Otomatik olarak oluşacak kodları inceleyiniz.,
10	5	Verilen (girilen) altı pozitif tamsayı değerini altı değişkene atanması. Ortalamasının bulunması ve çıkış ekranına yazdırılması işleminin algoritmasını tasarlayınız ve blok sistemi ile programını geliştiriniz. Otomatik olarak oluşacak kodları inceleyiniz.
11	1	Verilen (girilen) tamsayı değerini pozitif veya negatif sayı olup olmadığına göre sınıflandırmasını yapan ve sonucun çıkış ekranına yazdırılması işleminin algoritmasını tasarlayınız ve blok sistemi ile programını geliştiriniz. Otomatik olarak oluşacak kodları inceleyiniz.
12	2	Verilen (girilen) iki pozitif tamsayı değeri iki değişken tanımlanarak atama işlemlerinin yapılması ve çıkış ekranına yazdırılması ardından iki değişken değerlerini karşılaştırma yapılarak küçük olan değerini "Küçük Değer :-----" şeklinde çıkış ekranına yazdırılması işleminin algoritmasını tasarlayınız ve blok sistemi ile programını geliştiriniz. Otomatik olarak oluşacak kodları inceleyiniz.,
13	3	Yaşları verilen (girilen) üç kardeşin (Ali, Veli, Ayşe) yaş olarak büyük kardeşi bulan işlemi yapan işleminin algoritmasını tasarlayınız ve blok sistemi ile programını geliştiriniz. Otomatik olarak oluşacak kodları inceleyiniz.
14	4	Verilen (girilen) pozitif tam bir sayının çift ve tek olmasına göre sınıflandırma yapan işleminin algoritmasını tasarlayınız ve blok sistemi ile programını geliştiriniz. Otomatik olarak oluşacak kodları inceleyiniz.

15	5	<p>Parametreleri verilen (girilen) "a,b,c" "<math>y=ax^2+bx+c</math>" ikinci derece denklemin köklerini bulan. Kökleri değerlerine(sanal, reel)göre sınıflandırma yapan ve sonuçları çıkış ekranına yazdıran işleminin algoritmasını tasarlayınız ve blok sistemi ile programını geliştiriniz. Otomatik olarak oluşacak kodları inceleyiniz.</p> <p><math>ax^2+bx+c=0</math></p> <p><math>\Delta=b^2-4ac</math></p> <p>Eğer <math>\Delta&gt;0</math> ise denklemin iki farklı reel kökü vardır.</p> <p>Eğer <math>\Delta=0</math> ise denklemin iki reel kökü vardır fakat bunlar aynıdır.</p> <p>Eğer <math>\Delta&lt;0</math> ise denklemin reel kökü yoktur, yani çözüm kümesi boş kümedir.</p>
16	1	<p>1 den 7 e kadar sayıların karelerini çıkış ekranına alt alta yazdıran işleminin algoritmasını tasarlayınız ve blok sistemi ile programını geliştiriniz. Otomatik olarak oluşacak kodları inceleyiniz.,</p>
17	2	<p>1 den 7 e kadar sayıların karelerini döngü ile çıkış ekranına alt alta yazdıran işleminin algoritmasını tasarlayınız ve blok sistemi ile programını geliştiriniz. Otomatik olarak oluşacak kodları inceleyiniz.</p> <p>Sayı: 1 Karesi: 1</p>
18	3	<p>"0" girilene kadar sayıların toplamını hesaplayan ve çıkış ekranına yazdıran işleminin algoritmasını tasarlayınız ve blok sistemi ile programını geliştiriniz. Otomatik olarak oluşacak kodları inceleyiniz.</p> <p>Sayı: 1 Karesi: 1,</p>
19	4	<p>Girilen sayı bir önceki sayıdan küçükse işlemi tamamlayan ve sayıların toplamını hesaplayan ve çıkış ekranına yazdıran işleminin algoritmasını tasarlayınız ve</p>

		blok sistemi ile programını geliştiriniz. Otomatik olarak oluşacak kodları inceleyiniz.
20	5	Girilen sayı adedi 7 oluncaya kadar sayıların toplamını hesaplayan ve çıkış ekranına yazdıran işleminin algoritmasını tasarlayınız ve blok sistemi ile programını geliştiriniz. Otomatik olarak oluşacak kodları inceleyiniz.
21	1	Sıfırdan verilen değere kadar sayıları çıkış ekranına alt alta yazdıran işleminin algoritmasını tasarlayınız ve blok sistemi ile programını geliştiriniz. Otomatik olarak oluşacak kodları inceleyiniz.
22	2	Sıfırdan verilen değere kadar çift sayıları çıkış ekranına alt alta yazdıran işleminin algoritmasını tasarlayınız ve blok sistemi ile programını geliştiriniz. Otomatik olarak oluşacak kodları inceleyiniz.
23	3	Verilen bir sayının faktöriyelini hesaplayan sonucu çıkış ekranına yazdıran işleminin algoritmasını tasarlayınız ve blok sistemi ile programını geliştiriniz. Otomatik olarak oluşacak kodları inceleyiniz,
24	4	Verilen(girilen) n değeri için $f(x)=\sum x^2-1$ değerini bulan ve çıkış ekranına yazdıran işlemin algoritmasını tasarlayınız ve blok sistemi ile programını geliştiriniz. Otomatik olarak oluşacak kodları inceleyiniz.
25	5	Verilen sayıda taban genişliği olan yıldızlardan oluşan bir piramit oluşturan ve çıkış ekranına yazdıran işlemin algoritmasını tasarlayınız ve blok sistemi ile programını geliştiriniz. Otomatik olarak oluşacak kodları inceleyiniz.  *  * * *  * * * * *  * * * * * * *  * * * * * * * * *  Piramit
26	1	Sizden istenen karakterlerin kaç tanesinin rakam kaç tanesinin harf olduğunu (-) girilinceye kadar hesaplayan ve çıkış ekranına yazdıran işlemin

		algoritmasını tasarlayınız ve blok sistemi ile programını geliştiriniz. Otomatik olarak oluşacak kodları inceleyiniz.,
27	2	Sizden istenen pozitif tam sayıları (-1) değeri girilinceye kadar alıp toplamını ve ortalamasını çıkış ekranına yazdıran işlemin algoritmasını tasarlayınız ve blok sistemi ile programını geliştiriniz. Otomatik olarak oluşacak kodları inceleyiniz.,
28	3	Verilen 10 tabanında pozitif tamsayı değerin ikili sayı sistemine dönüşüm yapan işlemin algoritmasını tasarlayınız ve blok sistemi ile programını geliştiriniz. Otomatik olarak oluşacak kodları inceleyiniz.,
29	4	Verilen(girilen) BOLUNEN ve BOLEN gibi iki sayının sadece toplama ve çıkarma işlemleri yaparak kalanlı bölme yapan ve çıkış ekranına yazdıran işlemin algoritmasını tasarlayınız ve blok sistemi ile programını geliştiriniz. Otomatik olarak oluşacak kodları inceleyiniz.,
30	5	Verilen(girilen) X ve Y gibi iki sayının sadece toplama ve çıkarma işlemleri yaparak Çarpımını yapan ve çıkış ekranına yazdıran işlemin algoritmasını tasarlayınız ve blok sistemi ile programını geliştiriniz. Otomatik olarak oluşacak kodları inceleyiniz.,
31	1	Üç elemanlı bir diziye elemanlarını atayalım ve çıkış ekranına yazdıran işleminin algoritmasını tasarlayınız ve blok sistemi ile programını geliştiriniz. Otomatik olarak oluşacak kodları inceleyiniz.,
32	2	Rasgele üretilmiş n elemanlı bir diziyi başka bir diziye kopyalayan ve çıkış ekranına yazdıran işleminin algoritmasını tasarlayınız ve blok sistemi ile programını geliştiriniz. Otomatik olarak oluşacak kodları inceleyiniz.,
33	3	Rasgele üretilmiş n elemanlı bir dizinin elemanlarının toplamını bulan ve sonucu çıkış ekranına yazdıran işleminin algoritmasını tasarlayınız ve blok sistemi ile programını geliştiriniz. Otomatik olarak oluşacak kodları inceleyiniz.,
34	4	Rasgele üretilmiş n elemanlı bir dizinin en küçük elemanını bulan ve sonucu (dizi index'i ve değeri) çıkış ekranına yazdıran işleminin algoritmasını tasarlayınız ve blok sistemi ile programını geliştiriniz. Otomatik olarak oluşacak kodları inceleyiniz.,

35	5	Rasgele üretilmiş n elemanlı bir dizinin ortalamasını bulan ve sonucu çıkış ekranına yazdıran işleminin algoritmasını tasarlayınız ve blok sistemi ile programını geliştiriniz. Otomatik olarak oluşacak kodları inceleyiniz.,
36	1	Değerleri aşağıda verilen 3x3 elemanlı iki boyutlu bir dizinin Matrisin sonucu çıkış ekranına yazdıran işleminin algoritmasını tasarlayınız ve blok sistemi ile programını geliştiriniz. Otomatik olarak oluşacak kodları inceleyiniz.  $\begin{matrix} 1 & 2 & 1 \\ 2 & 4 & 6 \\ 3 & 5 & 1 \end{matrix}$
37	2	Rasgele üretilmiş nxn elemanlı iki boyutlu bir dizinin Matrisin sonucu çıkış ekranına yazdıran işleminin algoritmasını tasarlayınız ve blok sistemi ile programını geliştiriniz. Otomatik olarak oluşacak kodları inceleyiniz.
38	3	Rasgele üretilmiş nxn elemanlı iki boyutlu bir dizinin Matrisin en büyük elemanının satır – sütun numarasını çıkış ekranına yazdıran işleminin algoritmasını tasarlayınız ve blok sistemi ile programını geliştiriniz. Otomatik olarak oluşacak kodları inceleyiniz.,
39	4	Rasgele üretilmiş 3x3 elemanlı iki boyutlu bir kare Matrisin determinantını bulan ve sonucu çıkış ekranına yazdıran işleminin algoritmasını tasarlayınız ve blok sistemi ile programını geliştiriniz. Otomatik olarak oluşacak kodları inceleyiniz.  3 x 3 türünde bir kare matriste ise determinant aşağıdaki şekilde hesaplanır:  $\begin{matrix} a & b & c \\ d & e & f \\ g & h & i \end{matrix} = aei+bfh+cdg-(afh+bdi+ceg)$

40	5	Rasgele üretilmiş $n \times n$ elemanlı iki boyutlu bir dizinin Matrisin Devriğini (Transpose) bulan ve sonucu çıkış ekranına yazdıran işleminin algoritmasını tasarlayınız ve blok sistemi ile programını geliştiriniz. Otomatik olarak oluşacak kodları inceleyiniz.
41	1	Değerleri verilen bir dizi elemanlarının içerisinde yine değeri verilen bir sayıyı bulan buldu ise dizinin hangi sırasında bulunduğunu (dizinin index'ini) çıkış ekranına yazdıran bulunamadı ise bulunamadı yazdıran işleminin algoritmasını tasarlayınız ve blok sistemi ile programını geliştiriniz. Otomatik olarak oluşacak kodları inceleyiniz.,
42	2	Değerleri verilen bir dizi elemanlarının içerisinde en büyük değeri bulan dizinin hangi sırasında bulunduğunu (dizinin index'ini) çıkış ekranına yazdıran işleminin algoritmasını tasarlayınız ve blok sistemi ile programını geliştiriniz. Otomatik olarak oluşacak kodları inceleyiniz.
43	3	Değerleri verilen bir dizi elemanlarının içerisinde en çok tekrarlanan değeri bulan dizinin hangi sırasında bulunduğunu (dizinin index'ini) çıkış ekranına yazdıran işleminin algoritmasını tasarlayınız ve blok sistemi ile programını geliştiriniz. Otomatik olarak oluşacak kodları inceleyiniz.
44	4	Değerleri verilen bir dizi elemanlarını Seçerek Sıralama Algoritması (Selection Sort) kullanarak sıralayan ve sonucu çıkış ekranına yazdıran bulunamadı ise bulunamadı yazdıran işleminin algoritmasını tasarlayınız ve blok sistemi ile programını geliştiriniz. Otomatik olarak oluşacak kodları inceleyiniz.
45	5	Değerleri verilen bir dizi elemanlarını Kabarcık Sıralama Algoritması (Bubble Sort) kullanarak sıralayan ve sonucu çıkış ekranına yazdıran bulunamadı ise bulunamadı yazdıran işleminin algoritmasını tasarlayınız ve blok sistemi ile programını geliştiriniz. Otomatik olarak oluşacak kodları inceleyiniz.



**EK -3**

# Temel Programlama Eğitimi

Eğitim Sonu

Değerlendirme Sınavı

Hazırlayan: Eşref Seğmen

Soru -1: Siyah olarak görülen kutucuğa hangi seçenek gelmelidir?

**BASIC PROGRAMING EDUCATION**

Register | Login/Pop | Kullanıcı Adı | English | Türkçe

Education | Program | Exam | Help

**LOAD ALGORITHM**

Algorithm 1.5

LOAD

**Görev: 5**

5. Verilen (girilen) degerin karesi ile ekrandaki farklı çıkış ekranına yazdırılması işleminin algoritmasını tasarlayınız ve blok sistemi ile programını geliştiriniz. Otomatik olarak oluşacak kodları inceleyiniz.

**==>CODE MONITOR**

```

set deger to 1 prompt for number with message " Bir Deger Giriniz "
print " Girilen Deger "
print deger
print " Girilen degerin iki kati "
print deger * 2
print " Girilen Degerin Karesi "
print deger * deger

```

A)  \*

B)  \*

C)  \*

D)  \*

**==>CODE MONITOR**

```

<!--?php?www.w3.org/1999/xhtml?
<!--?type="textiles_get">110?>110?
<!--?name="Val">deger?>
<!--?name="Val">2?
<!--?type="text_prompt_end">
<!--?name="NUMBER">deger?
<!--?name="TEXT">
<!--?name="TEXT">Bir Sayı Giriniz?>
<!--?type="text">
<!--?>
<!--?>
<!--?>
<!--?type="text_prompt">
<!--?name="TEXT">
<!--?type="text">

```

**==>OUTPUT==> TEXT**

Start | CLEAR OUTPUT MONITOR

Copyright © Esref.Segmen

Soru -2: Siyah olarak görülen kutucuğa hangi seçenek gelmelidir?

**BASIC PROGRAMING EDUCATION**

Education | Program | Exam | Help

LOAD ALGORITHM

Algorithm 2.2

LOAD

Görev. 7

2. Verilen (girilen) iki pozitif tam sayı değerini iki değişken tanımlanarak atama işleminin yapılması ve çıkış ekranına yazdırılması ardından iki değişken değerlerini yer değiştirme yapılarak tekrar çıkış ekranına yazdırılması işleminin algoritmasını tasarlayınız ve blok sistemi ile programını geliştiriniz. Otomatik olarak oluşacak kodları inceleyiniz.

Logic

Loops

Math

Text

Lists

Variables

Functions

```

clear // 44 Değişken Yer Değiştirme //
print // 44 //
set deger1 0 to prompt for (number //) with message // 44 Birinci Değeri Giriniz //
print // 44 Girilen Değer 1 //
print // deger1 //
set deger2 0 to prompt for (number //) with message // 44 İkinci Değeri Giriniz //
print // 44 Girilen Değer 2 //
print // deger2 //
set temp to deger2 //
set deger1 to temp //
print // 44 Girilen Değer 1 //
print // deger1 //
print // 44 Girilen Değer 2 //
print // deger2 //

```

A) `set deger1 to deger2` C) `set temp to deger1`

B) `set deger1 to temp` D) `set deger2 to deger1`

==>CODE MONITOR

```

<code></code>
</pre>


>>>OUTPUT==>> TEXT



Copyright © Esref.Segmen


```

Soru -3: Program kořturulduęunda ıkıř ekranı hangi seenekte doęru olarak gsterilmiřtir?

The screenshot shows the BASIC PROGRAMMING EDUCATION software interface. The main window displays a code editor with a program that prompts for three names (Ali, Veli, Ayşe) and their ages (11, 5, 7). The program then prints the names and ages, and finally prints the name of the oldest person (Ali). The code is as follows:

```
set (Ali) to prompt for number with message (( Ali ))
set (Veli) to prompt for number with message (( Veli ))
set (Ayse) to prompt for number with message (( Ayse ))
print (( Ali'nin Yařı ))
print (( Veli'nin Yařı ))
print (( Ayse'nin Yařı ))
set (Buyuk_ism) to (( ))
if (Ali) > (Veli)
do
set (Buyuk_ism) to (( Ali ))
else
set (Buyuk_ism) to (( Veli ))
else if (Ayse) > (Veli)
do
set (Buyuk_ism) to (( Ayse ))
else
set (Buyuk_ism) to (( Veli ))
print (( Buyuk Kardař ))
print (( Buyuk_ism ))
```

A)

```
Ali'nin Yařı :11
Veli'nin Yařı :5
Aygeenin nin Yařı :7
Buyuk Kardař :Veli
```

B)

```
Ali'nin Yařı :11
Veli'nin Yařı :5
Aygeenin nin Yařı :7
Buyuk Kardař :Ali
```

C)

```
Ali'nin Yařı :11
Veli'nin Yařı :5
Aygeenin nin Yařı :7
Buyuk Kardař :Ali
```

D)

```
>>>OUTPUT=>> TEXT
Buyuk Kardař :Veli
```

E)

```
>>>OUTPUT=>> TEXT
Buyuk Kardař :Ayse
Ali'nin Yařı :11
Veli'nin Yařı :5
Aygeenin nin Yařı :7
Buyuk Kardař :Ali
```

Soru -4: Siyah olarak görülen kutucuğa hangi seçenek gelmelidir?

**BASIC PROGRAMMING EDUCATION**

Education | Program | Exam | Help

LOAD ALGORITHM

Algoritma 1.3

Görev: 13

3 Yaşları verilen (gülsen) üç kardeşin (Ali, Veli, Ayşe) yaş olarak büyük kardeşi bulan işlemi yapan işlemin algoritmasını tasarlayınız. Ya da sistemi ile programı geliştiriniz. Önemli olarak aşağıdaki kodları inceleyiniz.

```

def Ali'nin Yaşı to prompt for (number) with message 44 Ali'nin Yaşı ??
set Veli'nin Yaşı to prompt for (number) with message 44 Veli'nin Yaşı ??
set Ayşe'nin Yaşı to prompt for (number) with message 44 Ayşe'nin Yaşı ??
print 44 Ali'nin Yaşı ??
print 44 Veli'nin Yaşı ??
print 44 Veli'nin Yaşı ??
print 44 Ayşe'nin Yaşı ??
print 44 Ayşe'nin Yaşı ??
set Büyük Kardeş to 0
if
  if Ali'nin Yaşı > Veli'nin Yaşı
  do
    set Büyük Kardeş to Ali'nin Yaşı
  else
    set Büyük Kardeş to Veli'nin Yaşı
  else
    if Ayşe'nin Yaşı > Veli'nin Yaşı
    do
      set Büyük Kardeş to Ayşe'nin Yaşı
    else
      set Büyük Kardeş to Veli'nin Yaşı
  print 44 Büyük Kardeş ??
print 44 Büyük Kardeş ??
print 44 Büyük Kardeş ??

```

==>CODE MONITOR

```

set name="Ali" name="Veli" name="Ayşe"
set Ali'nin Yaşı to prompt for (number) with message 44 Ali'nin Yaşı ??
set Veli'nin Yaşı to prompt for (number) with message 44 Veli'nin Yaşı ??
set Ayşe'nin Yaşı to prompt for (number) with message 44 Ayşe'nin Yaşı ??
print 44 Ali'nin Yaşı ??
print 44 Veli'nin Yaşı ??
print 44 Veli'nin Yaşı ??
print 44 Ayşe'nin Yaşı ??
print 44 Ayşe'nin Yaşı ??
set Büyük Kardeş to 0
if
  if Ali'nin Yaşı > Veli'nin Yaşı
  do
    set Büyük Kardeş to Ali'nin Yaşı
  else
    set Büyük Kardeş to Veli'nin Yaşı
  else
    if Ayşe'nin Yaşı > Veli'nin Yaşı
    do
      set Büyük Kardeş to Ayşe'nin Yaşı
    else
      set Büyük Kardeş to Veli'nin Yaşı
  print 44 Büyük Kardeş ??
print 44 Büyük Kardeş ??
print 44 Büyük Kardeş ??

```

==>OUTPUT==>TEXT

```

Ali'nin Yaşı :11
Veli'nin Yaşı :6
Ayşe'nin Yaşı :7
Büyük Kardeş :Ali

```

Copyright © Esref Segmen

- A) set Büyük Kardeş to 44 Ayşe ??
- B) set Büyük Kardeş to 44 Veli ??
- C) print 44 Büyük Kardeş ??
- D) print 44 Büyük Kardeş
- E) set Büyük Kardeş to 44 Ali ??

Soru -5: Siyah olarak görülen kutucuğa hangi seçenek gelmelidir?

The screenshot shows a BASIC programming education interface. The main window is titled "BASIC PROGRAMMING EDUCATION". It features a "LOAD ALGORITHM" panel on the left with a "LOAD" button. The central area is a code editor with a yellow background, displaying a BASIC program for calculating a factorial. The program includes a loop and a message box. A black rectangular box obscures a line of code in the loop. Below the code editor, five options (A, B, C, D, E) are provided, each showing a different block of BASIC code. The right side of the interface includes a "CODE MONITOR" panel showing the execution of the program, and an "OUTPUT" panel displaying the results: "Faktoriyel Hesabi", "Girilen Değer: 5", and "Faktoriyel: 120".

Copyright © Esref Segmen

Soru -6: Siyah olarak görülen kutucuğa hangi seçenek gelmelidir?

The screenshot shows a BASIC programming education software interface. The main window is titled "BASIC PROGRAMING EDUCATION". On the left, there is a "LOAD ALGORITHM" section with a "LOAD" button and a "Görev: 24" section. The central area is a code editor with a yellow background, showing a BASIC program with a loop and a conditional statement. The code is as follows:

```

100 print "Merhaba!" with message: "Görüş"
110 print "Hi (Görüş_Organ)"
120 print "Hi (Görüş_Organ)"
130 print with "Hi form"
140 do
150   [ ]
160   [ ]
170   [ ]
180   [ ]
190 print "Hi (Görüş_Organ) on Screen!"
200 print

```

Below the code editor, there are five options (A, B, C, D, E) for the missing code blocks in the loop:

- A) `set [ ] to [ ]`
- B) `set [ ] to [ ]`
- C) `set [ ] to [ ]`
- D) `set [ ] to [ ]`
- E) `set [ ] to [ ]`

On the right, there is a "CODE MONITOR" section showing the execution of the program. The output is as follows:

```

100 print "Merhaba!" with message: "Görüş"
110 print "Hi (Görüş_Organ)"
120 print "Hi (Görüş_Organ)"
130 print with "Hi form"
140 do
150   [ ]
160   [ ]
170   [ ]
180   [ ]
190 print "Hi (Görüş_Organ) on Screen!"
200 print

```

At the bottom of the interface, there is a copyright notice: "Copyright © Esref Segmen".

Soru -7: Siyah olarak görülen kutucuğa hangi seçenek gelmelidir?

The screenshot displays a web-based programming environment titled "BASIC PROGRAMMING EDUCATION". The interface is divided into several sections:

- Top Bar:** Contains the logo "TEMEL PROGRAMLAMA", the title "BASIC PROGRAMMING EDUCATION", and buttons for "Register", "LogInApp", "Kullanıcı Adı", "English", and "Türkçe".
- Navigation:** A menu with "Education", "Program", "Exam", and "Help".
- Left Panel:**
  - LOAD ALGORITHM:** A dropdown menu showing "Algorithm 5.4" and a "LOAD" button.
  - Görev 24:** A task description in Turkish: "4. Verilen (girilen) n değeri için  $f(x) = 2x^2 - 1$  değerini bulan ve çıkış ekranına yazdırılan işlemin algoritmasını tasarlayınız ve blok sistemi ile programını geliştiriniz. Otomatik olarak oluşacak kodları inceleyiniz."
- Code Editor:** A central area with a "CODE MONITOR" tab. It shows a block-based code editor with the following code:

```
def C100 in prompt for number with message: 4 Copied in
set Location to 1
print Girilen Değeri
input
input with 100 from 1 to 100 by 1
do
  def A1000 in
    100
  end
end
print Sonuç
```

Below the code, there is a text box containing "Return the sum of the two numbers." and a list of options: A) 100, B) 200, C) 1000, D) 100.
- Right Panel:**
  - CODE MONITOR:** A tab showing the HTML output of the code, including a form with a "Girilen Değeri" input field and a "Sonuç" output field.
  - OUTPUT:** A tab showing the current output: "Girilen Değeri : 4" and "Sonuç : 26".

At the bottom of the interface, there is a copyright notice: "Copyright © Esref.Segmen" and a small number "7252".



Soru -8: Siyah olarak görülen kutucuğa hangi seçenek gelmelidir?

**BASIC PROGRAMING EDUCATION**

Yazdır | LoginApp | Kullanıcı Adı | English | Turkish

Educaion | Program | Exam | Help

LOAD ALGORITHM

Algorithm 2.1

LOAD

Görev 6

1. Verilen (girilen) bir değerin sayıdır bir değişkene atanması ve saklanması ardından çıkış ekranına yazdırılması işleminin algoritmasını yazdırınız ve blok sistemi ile programını geliştiriniz. Otomatik olarak oluşacak kodları inceleyiniz.

Logic  
Loops  
Math  
Text  
Lists  
Variables  
Functions

set deger to prompt for number with message " Bir Sayı Giriniz "

print deger

printin deger

A) print deger

B) print Girilen Deger

C) print " Girilen Deger "

D) print " Girilen Deger = "

E) print " Girilen = "

==>CODE MONITOR

```

<code>
</code>

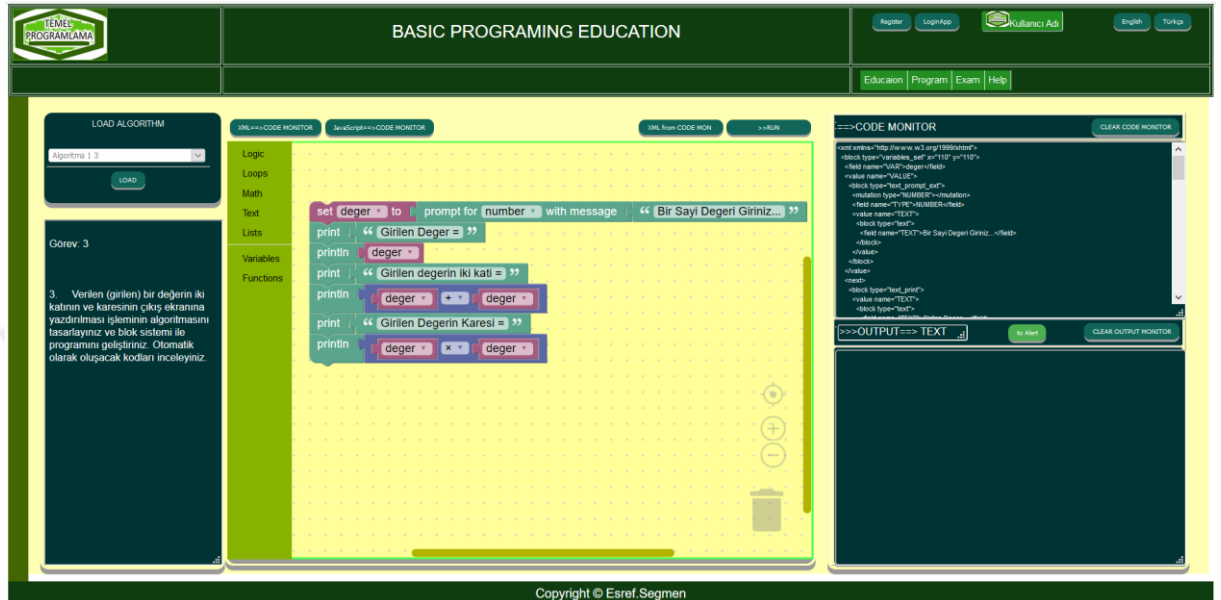
```

>>>OUTPUT==> TEXT

Girilen Deger =22

Copyright © Esref.Segmen

Soru -9: Program kořturulduęunda ıkıř ekranı hangi seenekte doęru olarak gsterilmiřtir?



The screenshot shows the BASIC PROGRAMMING EDUCATION software interface. The main workspace displays a block-based program with the following code:

```

set deger to prompt for number with message "Bir Sayı Degeri Giriniz..."
print "Girilen Deger = "
deger
println "Girilen degerin iki kati = "
deger * 2
println "Girilen Degerin Karesi = "
deger * deger

```

The right panel shows the CODE MONITOR output, which is currently empty. The left panel shows the task description:

3. Verilen (girilen) bir deęerin iki katının ve karesinin ıkıř ekranına yazdırılması iřleminin algoritmasını tasarlayınız ve blok sistemi ile programını geliřtirez. Otomatik olarak oluřacak kodları inceleyiniz.

A)	B)
Girilen Deger =9	Girilen Deger =9
C)	D)
Girilen Deger =3	Girilen Deger =9

Soru -10: Siyah olarak görülen kutucuğa hangi seçenek gelmelidir?

TEMEL PROGRAMCAMA

BASIC PROGRAMING EDUCATION

Register Login/Pop Kullanıcı Adı English Türkçe

Education Program Exam Help

LOAD ALGORITHM

Algorithm 1.4

LOAD

Görev: 4

4. Verilen (girilen) bir değerin bir küçük sayısının çığık ekranına yazdırılması işleminin algoritmasını tasarlayınız ve blok sistemi ile programını geliştiriniz. Otomatik olarak oluşacak kodları inceleyiniz.

Logic  
Loops  
Math  
Text  
Lists

Variables  
Functions

clear

set deger to prompt for number with message " Bir Sayı Giriniz..."

print " Girilen Deger = "

println deger

println "-----"

print " Girilen degerin bir eksigi = "

println deger

println "-----"

A) / B) - C) \*

D) = E) +

==>CODE MONITOR

```

<code>
</code>

```

>>>OUTPUT==> TEXT

Girilen Deger =5

Girilen degerin bir eksigi =4

Copyright © Esref Seçmen

**Student No** :

**Name** :

**Surname** :

SORU	CEVAPLAR				
	A	B	C	D	E
1	0	0	0	0	0
2	0	0	0	0	0
3	0	0	0	0	0
4	0	0	0	0	0
5	0	0	0	0	0
6	0	0	0	0	0
7	0	0	0	0	0
8	0	0	0	0	0
9	0	0	0	0	0
10	0	0	0	0	0

## 11 ÖZGEÇMİŞ

EŞREF SEĞMEN

Kontrol ve Bilgisayar Mühendisi,

Sivas, Suşehri’nde doğdu (1967), 1989 yılında İstanbul Teknik Üniversitesi, Elektrik-Elektronik Fakültesi, Kontrol ve Bilgisayar Mühendisliği Bölümünü tamamladı ve mühendis olarak meslek hayatına başladı. Eğitimine İstanbul Üniversitesi İşletme İktisadi Enstitüsü ve MBA eğitimleri alarak devam etti. Gerek kamu ve gerekse özel sektörde, kontrol ve bilgisayar mühendisi olarak; programcı, uzman, yönetici, teknik üst düzey yönetici, yarı zamanlı öğretim görevlisi ve danışman olarak farklı birçok görevde, ağırlıklı olarak telekomünikasyon alanında yirmi yedi yıl çalıştı. Halen, bilgi ve iletişim sektöründe üst düzey yönetici, danışman ve yarı zamanlı öğretim görevlisi olarak meslek hayatına devam etmektedir. Evli ve bir çocuk babasıdır.