

**JAVA PROGRAMLAMA DİLİNİN BİLGİSAYAR DESTEKLİ
ÖĞRETİMİ**

Cahide ERPOLAT

**YÜKSEK LİSANS TEZİ
ELEKTRONİK VE BİLGİSAYAR EĞİTİMİ**

**GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

HAZİRAN 2006

ANKARA

**JAVA PROGRAMLAMA DİLİNİN BİLGİSAYAR DESTEKLİ
ÖĞRETİMİ**

Cahide ERPOLAT

**YÜKSEK LİSANS TEZİ
ELEKTRONİK VE BİLGİSAYAR EĞİTİMİ**

**GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

HAZİRAN 2006

ANKARA

Cahide ERPOLAT tarafından hazırlanan JAVA PROGRAMLAMA DİLİNİN BİLGİSAYAR DESTEKLİ ÖĞRETİMİ adlı bu tezin yüksek lisans tezi olarak uygun olduğunu onaylarım.

Prof.Dr. Ömer Faruk BAY
Tez Yöneticisi

Bu çalışma, jürimiz tarafından Elektronik ve Bilgisayar Eğitimi Anabilim Dalında Yüksek lisans tezi olarak kabul edilmiştir.

Başkan : Yrd. Doç. Dr. Hasan Şakir BİLGE

Üye : Yrd. Doç. Dr. O. Ayhan ERDEM

Üye : Prof. Dr. Ömer Faruk BAY

Tarih : 06/06/2006

Bu tez, Gazi Üniversitesi Fen Bilimleri Enstitüsü tez yazım kurallarına uygundur.

TEZ BİLDİRİMİ

Tez içindeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, ayrıca tez yazım kurallarına uygun olarak hazırlanan bu çalışmada orijinal olmayan her türlü kaynağa eksiksiz atıf yapıldığını bildiririm.

Cahide ERPOLAT

JAVA PROGRAMLAMA DİLİNİN BİLGİSAYAR DESTEKLİ ÖĞRETİMİ

(Yüksek Lisans Tezi)

Cahide ERPOLAT

**GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

Haziran 2006

ÖZET

Bilgisayar destekli sistemlerin ve bilgi teknolojisinin kullanımı hemen hemen her mesleğin önemli bir parçası olmaktadır ve eğitim kavramlarını değiştirmekte, ayrıca çoğu üniversite ve eğitim kurumunun bilgisayar destekli eğitimle ilgili yeni stratejiler planlamasına neden olmaktadır. “Bilgisayar Destekli Öğretim (BDÖ)” de teknolojinin yeni ve etkin olarak kullanılabildiği bir öğretim tekniği olarak karşımıza çıkmaktadır. Bu çalışmada, bilgisayar destekli öğretim kavramı ve esasları yanında Java dili derinlemesine incelenerek, Java programlama dilinin bilgisayar destekli öğretiminin gerçekleştirilmesine yönelik bir uygulama tasarlanmış ve geliştirilmiştir. Java dilinin öğretimini gerçekleştirmek için 10 bölümden oluşan bir eğitim yazılımı Authorware programı kullanılarak hazırlanmıştır. Her bölüm sonunda bilgi ölçücü testler oluşturulmuştur. Bölümler Flash programıyla hazırlanmış animasyonlarla etkileşimli hale getirilmiştir.

Bilim Kodu : 702.6.004
Anahtar Kelimeler : Bilgisayar Destekli Öğretim, Java, Nesne Yönelimli Programlama
Sayfa Adedi : 80
Tez Yöneticisi : Prof.Dr. Ömer Faruk BAY

COMPUTER AIDED JAVA INSTRUCTION**(M.Sc. Thesis)****Cahide ERPOLAT****GAZI UNIVERSITY****INSTITUTE OF SCIENCE AND TECHNOLOGY****June 2006****ABSTRACT**

The use of computer based system and information technology are becoming an important part of the everyday work on almost any profession, and changed education concepts and many universities and education organizations decided to plan new strategies concerning computer based technologies. As a result of these developments Computer Aided Instruction (CAI) methods came out. In this study, by examining computer based education concept and its basis and with these Java programming language, an application was designed and completed that is aimed at realizing teaching Java. The educational software, consist of ten parts, was developed by Authorware program for teaching Java. Each part has measurement of information test. Using Flash program for developing animation parts was became most efficiently.

Science Code : 702.6.004**Key Words : Computer Aided Instruction, Java, Object Oriented
Programming****Page Number : 80****Adviser : Prof. Dr. Ömer Faruk BAY**

TEŐEKKÜR

Çalıőmalarım süresince yardımlarını esirgemeyen danışmanım Sn. Prof.Dr. Ömer Faruk BAY'a, çalıőma arkadaşlarıma ve hocalarıma yardımlarından dolayı teşekkürü bir borç bilirim.

İÇİNDEKİLER

	Sayfa
ÖZET.....	iv
ABSTRACT.....	v
TEŞEKKÜR.....	vi
İÇİNDEKİLER	vii
ÇİZELGELERİN LİSTESİ.....	xv
RESİMLERİN LİSTESİ	xvi
SİMGELER VE KISALTMALAR.....	xviii
1. GİRİŞ	1
2. BİLGİSAYAR DESTEKLİ ÖĞRETİM	4
2.1. Bilgisayar Destekli Eğitimin Tarihçesi.....	4
2.2. Bilgisayar Destekli Öğretim ve Öğrenme Teorileri.....	5
2.2.1. Davranışçı öğretim anlayışı.....	6
2.2.2. Bilişsel öğretim anlayışı.....	6
2.2.3. Oluşturmacı (constructivist) öğretme anlayışı	6
2.3. Bilgisayar Destekli Öğretimin Olumlu Yönleri.....	7
2.3.1. Öğrenme hızı.....	7
2.3.2. Katılımcı öğrenme.....	7
2.3.3. Öğretimsel etkinliklerin çeşitliliği.....	8
2.3.4. Öğrenci etkinliklerinin ve performansının izlenebilmesi.....	8
2.3.5. Zamandan ve ortamdaki bağımsızlık	8
2.4. Bilgisayar Destekli Öğretimin Olumsuz Yönleri.....	8

Sayfa

2.4.1. Öğrencilerin sosyo-psikolojik gelişimlerini engellemesi.....	9
2.4.2. Özel donanım ve beceri gerektirmesi.....	9
2.4.3. Eğitim programını desteklememesi	9
2.4.4. Öğretimsel niteliğin zayıf olması	9
2.5. Bilgisayar Destekli Öğretimde Öğretmenin Rolü.....	10
2.6. Bilgisayar Destekli Öğretimde Öğrencinin Rolü.....	10
2.7. Bilgisayar Destekli Öğretimin Öğrencinin Gelişimine Etkisi	11
2.8. Bilgisayar Destekli Öğretimin Öğretmene Faydaları	12
2.9. Bilgisayar Destekli Öğretim İçin Temel Gereksinimler	13
2.9.1. Bilgisayar donanımı	13
2.9.2. Bilgisayar yazılımları	13
2.9.3. İnsan kaynakları	14
2.9.4. Eğitim programında düzenlemeler	15
2.10. Bilgisayar Destekli Öğretimde Yöntemler	15
2.10.1. Alıştırma ve tekrar.....	15
2.10.2. Bire-bir öğretim.....	16
2.10.3. Problem çözme.....	16
2.10.4. Deney ve laboratuvar çalışması (benzetim)	16
2.10.5. Eğitici oyunlar	16
2.10.6. Bilgi deposu	17
2.11. Bilgisayar Destekli Öğretim Uygulamaları	17
2.11.1. Laboratuvar yöntemi	17

Sayfa

2.11.2. Her sınıfa PC yöntemi.....	18
2.11.3. Kişisel PC yöntemi.....	18
2.11.4. İnternet yoluyla eğitim yöntemi	18
2.12. Bilgisayar Destekli Öğretimdeki Sorunlar.....	19
2.13. Farklı Okul Kademelerinde Bilgisayar Destekli Öğretim	19
2.14. Dünya’da Bilgisayar Destekli Öğretim.....	20
2.15. Türkiye’de Bilgisayar Destekli Öğretim.....	21
2.16. Eğitim Yazılımı Geliştirme.....	24
2.16.1. Bdö yazılımı geliştirmede temel aşamalar	24
3. JAVA PROGRAMLAMA DİLİ	27
3.1. Nesne Yönelimli Programlama.....	27
3.1.1. Sarmalama (encapsulation).....	27
3.1.2. Kalıtım (inheritance).....	27
3.1.3. Çok biçimlilik (polymorphism)	278
3.2. Java’ya Giriş	28
3.2.1. Java’nın tarihçesi.....	28
3.2.2. Java’ nın uygulama alanları	29
3.2.3. Java’nın özellikleri ve avantajları	29
3.2.4. Java versiyonları.....	29
3.2.5. Java programlamak için neler gerekli	29
3.2.6. Programın derlenmesi	30
3.2.7. Java sanal makinesi	30
3.2.8. Java programlarının yapısı	31

Sayfa

3.2.9. Sınıfların(Class) yapısı.....	31
3.2.10. Metotların yapısı	31
3.2.11. Tanımlayıcılar	31
3.2.12. Açıklamalar	31
3.3. Veri Tipleri, Değişkenler ve Diziler	32
3.3.1. Veri tipleri	32
3.3.2. Değişkenler	32
3.3.3. Diziler.....	33
3.3.4. Karakter katarları	33
3.4. Operatörler	33
3.4.1. Aritmetik operatörler.....	34
3.4.2. Bir artırma bir azaltma operatörleri.....	34
3.4.3. İlişkisel operatörler.....	34
3.4.4. Mantıksal operatörler	35
3.4.5. Atama operatörleri	35
3.4.6. Soru işareti operatörü	36
3.4.7. String + Operatörü	36
3.5. Kontrol İfadeleri	36
3.5.1. Java'nın seçim ifadeleri.....	36
3.5.2. Java'nın döngü İfadeleri.....	37
3.5.3. Sıçrama ifadeleri	38
3.6.Sınıfların Tanıtımı.....	39
3.6.1. Sınıfın genel biçimi	39

Sayfa

3.6.2. Basit bir sınıf	39
3.6.3. Nesnelerin bildirimi	40
3.6.4. Metotlara giriş	41
3.6.5. Yapılandırıcılar	42
3.6.6. Artık toplama	43
3.6.7. Finalize() metodu	43
3.7. Metot ve Sınıflar	43
3.7.1. Metotları aşırı yüklemek	43
3.7.2. Nesneleri parametre olarak kullanmak.....	44
3.7.4. Yinelenme	45
3.7.5. Erişim kontrolüne giriş.....	45
3.7.6. Static anahtar sözcüğü.....	45
3.7.7. Final anahtar sözcüğü.....	45
3.7.8. İç içe yerleşmiş ve iç sınıflar.....	46
3.7.9. String sınıfına giriş.....	46
3.8. Kalıtım	46
3.8.1. Kalıtımın temelleri	46
3.8.2. Üye erişimi ve kalıtım.....	47
3.8.3. Super kullanımı	47
3.8.4. Yapılandırıcıların çağırılması.....	47
3.8.5. Metodu geçersiz kılmak (Overriding).....	47
3.8.6. Dinamik metot göndermesi	48
3.8.7. Metotları geçersiz kılmanın nedeni.....	48

	Sayfa
3.8.8. Özet sınıfları kullanmak.....	48
3.8.9. Final' in kalıtımla kullanılması	49
3.8.10. Object sınıfı.....	50
3.9. Paketler ve Arabirimler.....	50
3.9.1. Paketler.....	50
3.9.2. Paketleri tanımlamak.....	51
3.9.3. Paketleri bulmak ve classpath	51
3.9.4. Erişim koruması	52
3.9.5. Paketlerin alınması	52
3.9.6. Arabirimleri tanımlama	53
3.9.7. Arabirimlerin uygulanması	53
3.9.8. Kısmi uygulamalar	54
3.9.9. Arabirimlerde değişkenler.....	54
3.9.10. Arabirimler genişletilmesi.....	54
3.10. İstisna Yönetimi.....	55
3.10.1. İstisna tipleri.....	55
3.10.2. Try ve catch kullanımı	56
3.10.3. Throw	56
3.10.4. Throws	56
3.10.5. Finally	57
3.10.6. Özel istisna alt sınıfları oluşturmak.....	57
3.10.7. Zincirleme İstisnalar	57
4. EĞİTİM YAZILIMI UYGULAMASININ TASARIMI	58

Sayfa

4.1. Uygulamanın Amacı	58
4.2. Uygulamada Kullanılan Teknolojiler	58
4.2.1. Authorware.....	58
4.2.2. Flash	59
4.3. Uygulamanın Bölümleri ve Tasarımı	59
4.3.1. Konular bölümü	59
4.3.2. Test bölümü.....	60
4.3.3. Animasyonlar	60
4.3.4. Etkileşimli örnek sorular	60
4.3.5. Seslendirme	60
4.4.Eğitim Yazılımının Kullanımı	61
4.4.1. Eğitim yazılımının başlangıç ekranı	61
4.4.2. Konular bölümü	62
4.4.3. Konu ekranı	62
4.4.4.Testler bölümü	64
4.4.5. Animasyonlar	66
4.4.6. Etkileşimli örnek sorular	66
4.5. Uygulamada Kullanılan Düğmeler	67
4.5.1. Konular düğmesi	67
4.5.2. Hedef düğmesi	67
4.5.3. Sözlük düğmesi	69
4.5.4. Ara düğmesi	69
4.5.5. Ses düğmesi.....	70

	Sayfa
4.5.6. Yardım düğmesi	70
4.5.7. Kapat düğmesi.....	70
4.5.8. Giriş düğmesi	72
4.5.9. Sonraki sayfa düğmesi	72
4.5.10. Önceki sayfa düğmesi	72
4.5.11. Sonraki bölüm düğmesi	72
4.5.12. Önceki bölüm düğmesi	73
4.5.13. Animasyon düğmesi.....	73
4.5.14. Sonraki komut düğmesi	73
4.5.15. Önceki komut düğmesi	73
5. SONUÇ VE ÖNERİLER.....	74
KAYNAKLAR	77
ÖZGEÇMİŞ	80

ÇİZELGELERİN LİSTESİ

Çizelge	Sayfa
Çizelge 3.1. Veri tipleri.....	32
Çizelge 3.2. Aritmetik operatörler	34
Çizelge 3.3. Bir artırma bir azaltma operatörleri	34
Çizelge 3.4. İlişkisel operatörler	34
Çizelge 3.5. Mantıksal operatörler	35
Çizelge 3.6. Atama operatörleri	35

RESİMLERİN LİSTESİ

Resim	Sayfa
Resim 4.1. Eğitim yazılımının başlangıç ekranı	61
Resim 4.2. Eğitim yazılımının ikinci ana ekranı.....	62
Resim 4.3. Eğitim yazılımının bölüm giriş ekranı	63
Resim 4.4. Eğitim yazılımının konu ekranı	63
Resim 4.5. Eğitim yazılımının özet ekranı.....	64
Resim 4.6. Test ekranı.....	65
Resim 4.7. Test sonucu ekranı	65
Resim 4.8. Animasyon düğmesi	66
Resim 4.9. Örnek animasyon ekranı	66
Resim 4.10. Etkileşimli örnek soru	67
Resim 4.11. Konular düğmesi ve penceresi	68
Resim 4.12. Hedefler düğmesi ve penceresi	68
Resim 4.13. Sözlük düğmesi ve penceresi	69
Resim 4.14. Ara düğmesi ve penceresi	70
Resim 4.15. Ses düğmesi	70
Resim 4.16. Yardım düğmesi ve penceresi	71
Resim 4.17. Kapat düğmesi ve penceresi.....	71
Resim 4.18. Giriş düğmesi	72
Resim 4.19. Sonraki sayfa düğmesi	72
Resim 4.20. Önceki sayfa düğmesi	72
Resim 4.21. Sonraki bölüm düğmesi	72

Resim	Sayfa
Resim 4.22. Önceki bölüm düğmesi	73
Resim 4.23. Animasyon düğmesi	73
Resim 4.24. Sonraki komut düğmesi	73
Resim 4.25. Önceki komut düğmesi	73

SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış bazı kısaltmalar, açıklamaları ile birlikte aşağıda sunulmuştur.

Kısaltmalar	Açıklama
BDE	Bilgisayar Destekli Eğitim
BDÖ	Bilgisayar Destekli Öğretim
GUI	Grafiksel Kullanıcı arayüzü (Graphical User Interface)
JRE	Java Runtime Engine
JSP	Java Server Sayfaları (Java Server Page)
JSDK	Java Yazılım Geliştirme Aracı (Java Software Development Kit)
JVM	Java Sanal Makinesi (Java Virtual Machine)
MEB	Milli Eğitim Bakanlığı
OOP	Nesne Yönelimli Programlama (Object Oriented Programming)
PLATO	Programmed Logic for Automatic Teaching Operation
SDK	Software Development Kit
TICCIT	Time-Shared Interactive Computer Controlled Information Tv
TÜBİTAK	Türkiye Bilimsel Teknik Araştırma Kurumu

1. GİRİŞ

Bilim ve teknoloji birikimi günümüzde hızla artmakta, bu hız sonucunda bilgi ve teknolojinin geçerlilik süresi kısalmaktadır. Bu durum ise eğitim programlarının yeni ihtiyaçlara göre çeşitlendirilmesini, öğretim programları ve eğitim araçlarının sürekli olarak yenilenmesini zorunlu kılmaktadır.

Bilim ve teknolojinin günümüzde ulaştığı boyutları ve kazandığı gelişme hızını, geleneksel eğitim sistemi artık yakından takip edemez hale gelmiştir. Bu durum ileri ülkelerde yeni eğitim teknolojilerinin geliştirilmesini ve örgün ve yaygın eğitime sokulmasını zorunlu kılmıştır.

Teknoloji, öğreticinin görevini azaltıp onun yerini alamaz. Teknoloji ne kadar ileri götürülürse götürülsün, dersi yine öğretmenin hazırlamasına ve sınıfta yine öğretmenin yorumuna gerek vardır [1].

Teknolojinin ve bilgisayar tabanlı eğitim sistemlerinin hızla gelişimi eğitim kurumlarında uygulanabilecek yepyeni öğretim tekniklerinin oluşturulabilmesine olanak sağlamıştır. “Bilgisayar Destekli Öğretim” de teknolojinin yeni ve etkin olarak kullanılabilirdiği bir öğretim tekniği olarak karşımıza çıkmaktadır [2].

Bilgisayar Destekli Öğretim (BDÖ): Bilgisayarların eğitim sistemine dahil edilerek etkili birer eğitim ve öğrenim aracı olarak kullanılmalarıdır.

Bilgi ve iletişim alanlarındaki gelişmelerin çağdaş eğitim düzeyini yakalayabilmek için eğitim programlarıyla bütünleştirilmesi kaçınılmazdır. Bu alanda yapılan çalışmalarda bilgisayar destekli öğretim geleneksel eğitim yöntemiyle karşılaştırıldığında başarının daha yüksek olduğunu bulunmuştur. Bunun yanında bilgisayar teknolojisi bireyin oluşturacağı bilgileri belleğinde hem grafiksel hem de sembolik temsil biçimleri dâhilinde depolamasına olanak sağlayarak bilgiyi çok boyutlu olarak depolatarak hem öğrenmeyi daha anlamlı hem de bilgi depolamasını uzun vadeli kılabilir [2].

1926'da Pressley'in geliřtirdiđi ve ilköđretim makinesi olan "řeker makinesi" eđitim-öđretim ortamına direkt olarak giren ilk mekanik araç oldu. Makine bir dizi soru kartı; bu kartların sırayla gösterimini sađlayan bir kol; sunulan soru için yanıtı seçtiren bir dizi buton ve bir "ödöl"(řeker) sunma biriminden oluřmaktadır [3].

İlk bilgisayar destekli öđretim geliřtirme çalıřmaları 1970'lerin bařında, geleneksel öđretime destek mahiyetinde ortaya çıktı. Bilgisayarın eđitimde kullanılmasına iliřkin ilk geniř kapsamlı proje sayılabilen PLATO(Programmed Logic for Automatic Teaching Operation) ise üniversitelerde deđiřik disiplin alanında öđrencilerin bilgisayar destekli öđretim gereksinimini karřılamak amacı ile geliřtirilmiřtir. TICCIT (Time-Shared Interactive Computer Controlled Information Television) sistemi ise, 1977'de Texas ve Brigham Üniversitelerince ortaklařa geliřtirilen ve özellikle Matematik ve İngilizce derslerine yer veren bir projedir [4,5].

Keser'in 1988'de "Bilgisayar Destekli Öđretim İçin Bir Model Önerisi" konulu doktora tezinin amacı bilgisayar destekli öđretilimi ana hatlarıyla tanıtmak ve Türk Eđitim Sistemi'nde mevcut kořulları dikkate alarak ortaöđretim kurumları için bilgisayar destekli öđretim model önerisi geliřtirmeye yöneliktir [6].

řimřek'in 1990'daki "Programlama dilleri öđretilimi için anlayıřlı bir eđitim sisteminin tasarımı ve gerçekteřirimi" adlı çalıřmasında, anlayıřlı eđitim sistemlerinde olması gereken uzman modölü, öđrenci modeli, öđretici modölü ve bunlar arasındaki iliřkilerle C programlama dilinin öđretilimi gerçekteřirilmiřtir. Bu çalıřma ayrıca diđer programlama dillerinin öđretilimi için de kullanılabilmektedir [7].

Çomonöđlu 1990'daki "Pascal için bir bilgisayar destekli öđretim sistemi tasarımı ve gerçekteřirimi" adlı çalıřmasında kullanıcının program adı, izleyeceđi yordamları ve deđiřkenleri bařlangıçta ekrandan alınarak hangi ekler hangi noktalara yapılacađı belirlenip, yeni kaynak program oluřturulup, bu programın çalıřtırılması yöntemiyle Pascal programlama dilinin öđretilimi sađlanmıřtır [8].

Mutlu 2001' deki "Java tabanlı bir eğitsel yazılım tasarımı" adlı çalışmasında Linux ortamında ve Java programlama dili kullanılarak sanal ders programı geliştirilmiştir. Sanal ortamda eğitim merkezinin hazırlanmış olduğu 56 modüllük Calculus dersi içerikleri, sistemin gerektirdiği formata çevrilmiştir [9].

Bu çalışmanın amacı Java Programlama Dilinin Bilgisayar Destekli Öğretimi için bir uygulama yazılımı geliştirmektir. Java 1995 yılında ortaya çıkmış olan yeni bir programlama dilidir. Yeni olmasına rağmen çok kısa bir süre içinde yaygın bir şekilde kullanılmıştır. Java oldukça basit, nesne yönelimli programlamaya dayalı, güvenli, dinamik ve yüksek performansa sahip bir programlama dilidir.

Geliştirilen eğitim yazılımıyla Java programının en etkili bir şekilde öğretimi hedeflenmektedir. Bu çalışmada Authorware eğitim materyali hazırlama programı kullanılarak Java programlama dilinin bilgisayar destekli öğretimi tasarlanmakta ve geliştirilmektedir. Konuların animasyonlu anlatımını sağlamak ve konuların daha etkili öğretimini gerçekleştirmek için Flash gibi yardımcı programlar kullanılmaktadır.

Bu çalışma beş bölümden meydana gelmektedir. Birinci bölümde bilgisayar destekli öğretim ile ilgili daha önceden yapılan çalışmalar, tezin amacı ve kapsamı üzerinde durulmaktadır. Çalışmanın ikinci bölümünde bilgisayar destekli öğretim hakkında kapsamlı bilgi verilmektedir. Üçüncü bölümde, Java programlama dili ile ilgili bilgi verilmektedir. Dördüncü bölümde, Java programlama dilinin bilgisayar destekli öğretimi için geliştirilen eğitim yazılımı hakkında bilgi verilmektedir. Beşinci bölümde geliştirilen eğitim yazılımı ile ilgili değerlendirme yapılmış ve önerilerde bulunulmuştur.

2. BİLGİSAYAR DESTEKLİ ÖĞRETİM

Bilgisayar Destekli Öğretim (BDÖ); bilgisayarların eğitim sistemine dahil edilerek etkili birer eğitim ve öğrenim aracı olarak kullanılmalarıdır. Bir eğitim aracı olarak bilgisayarlar, öğretmenin ders işleme metotlarını güçlendirip zenginleştirirken, bir öğrenim aracı olarak da öğrencinin bilgiye ulaşma, kavrama ve kullanma verimini arttırır [10].

BDÖ öğrencinin bilgisayar kullanmayı öğrenmesi değildir. Bu, bilgisayar eğitiminin tanımına girer ve yalnızca öğrenciyle ilgilidir. BDÖ, bilgisayarın bir amaç değil bir araç olarak kullanılmasıdır ve öğretmen-öğrenci-teknoloji üçgeni içinde gerçekleşir. BDÖ uygulamalarında öğretmen, ders işleme yöntemlerini teknoloji sayesinde ciddi boyutlarda zenginleştirebilir. Geleneksel yöntemlerde aktarılmak istenen bilgi, öğrenciye, çağdaş eğitim yazılımları sayesinde bugüne kadar hayal bile edilemeyen biçimlerde aktarılabilir. Bilgisayar öğretmen için yalnızca bir eğitim aracı olarak değildir. Bu teknoloji, öğretmenin kendi bilgisini geliştirmesine, yeni öğretim yöntemleri oluşturmaya ve bu yöntemleri öğrenciler üzerinde uygulamasına da olanak sağlar.

Bilgisayar teknolojisi sayesinde eğitim öğrenci-merkezli hale gelebilir, yani öğrencinin araştırarak, deneyerek ve paylaşarak öğrenmesine uygun ortam yaratılabilir. Böylece, her öğrenci kendi hızında öğrenme şansına da sahip olabilir. BDÖ, öğrenmeyi pasif değil aktif bir süreç haline getirir. Bu teknoloji aracılığıyla, bilgiyi bizzat aramak, bulmak, analiz etmek ve uygulamak, çağdaş eğitimin en önemli yaklaşımı haline gelmektedir [11].

2.1. Bilgisayar Destekli Eğitimin Tarihçesi

1950'li yılların sonlarında ABD'nde gelişmiş bazı üniversitelerde, bilgisayar yönetsel amaçlarla kullanılıyordu. 1970'li yıllarda bilgisayarlı eğitim uygulamalarıyla ilgili projeler de geliştirilmeye başlanmıştır. Bu projelerden en önemlileri IBM 1500, PLATO (Programmed Logic for Automatic Teaching

Operation) ve TICCIT (Time-Shared Interactive Computer Controlled Information Television) sistemlerdir.

IBM 1500 projesi ile önceleri üniversite düzeyinde bilgisayar destekli fizik ve istatistik öğretimi, 1960'ların ortasında ise okuma ve matematik becerilerinin yükseltilmesine ilişkin öğretim yapılmıştır. Bilgisayarın eğitimde kullanılmasına ilişkin ilk geniş kapsamlı proje sayılabilen PLATO ise üniversitelerde değişik disiplin alanında öğrencilerin bilgisayar destekli öğretim gereksinimini karşılamak amacı ile geliştirilmiştir. TICCIT sistemi ise, 1977'de Texas ve Brigham Üniversitelerince ortaklaşa geliştirilen ve özellikle Matematik ve İngilizce derslerine yer veren bir projedir [4].

Fransa'da 1983'te "100.000 Bilgisayar" hedefinin belirlenmesi ve bu hedefe kısa sürede varılması üzerine 1985'te "Herkes için İnfomatik" programının başlatılması; Federal Almanya'da 1975'te orta öğretimin üst kademelerine bilgisayar eğitimi verilmesi ve daha sonra alt kademelerine de yaygınlaştırılması bu gelişmelere örnek olarak verilebilir [4].

Türkiye'deki gelişmeler ise ortaöğretim kurumlarına 1100 mikrobilgisayar alınmakla başlamıştır. Daha sonraları ise bilgisayar eğitimi yerine bilgisayarın bir eğitim aracı olarak kullanıldığı bilgisayar destekli eğitim uygulamalarının başlatılması uygun görülmüştür. Milli Eğitim Bakanlığı Dünya Bankası katılımı ile 53 bilgisayar deneme okuluna 1666 adet bilgisayar alınmıştır ve bu okullarda bilgisayar laboratuvarları kurulmuştur. Ayrıca bu okullara denemek üzere Bilim ve Teknik ansiklopedisi, İngilizce, matematik, fizik, kimya ve biyoloji konularında ders yazılımları temin edilmiştir. Donanım ve alt yapı çalışmalarına ek olarak 1996 yılı içerisinde 256 yeni formatör öğretmeninin eğitimi yapılmıştır [11].

2.2. Bilgisayar Destekli Öğretim ve Öğrenme Teorileri

Öğrenme, insanların yaşamları boyunca karşılaştıkları çeşitli durumlarla etkileşimleri sonucu oluşan kalıcı izli davranış değişmeleridir.

2.2.1. Davranışçı öğretim anlayışı

Bazı psikologlar için zihinsel süreçler önemli değildir. İnsan zihninde neler olduğunu bilinemez. Ayrıca buna gerek de yoktur. Önemli olan herkes tarafından gözlenebilen, ölçülebilen davranışlardır. Böylece çalışmalarını nesnel bir temele oturtan ve davranışlar üzerine odaklaştıran psikologlara davranışçılar, geliştirdikleri kuramlara da davranışçı kuramlar denmiştir [3].

Davranışçılar öğrenmeyi uyarıcılarla davranışlar arasında bir bağ kurma süreci olarak açıklamaktadırlar. Uyarıcılarla davranışlar arasında oluşan bağ güçlendiği ve bir alışkanlık durumuna geldiği zaman “*öğrenme*” oluşmaktadır. Alışkanlık durumuna gelmeyen ve yapılmayan davranışlar öğrenilmiş sayılmaz [3].

2.2.2. Bilişsel öğretim anlayışı

Bilişsel öğrenme kuramcıları davranışçıların aksine, öğrencilerin sunulan bilgileri alan durağan bireyler olmadığı, ancak bilgiyi alan, bunu kodlayan, hafızaya kaydeden ve gerektiğinde hafızadan geri çağırıp kullanan bireyler olduğunu savunmuşlardır [3].

Bilişsel kuramcılara göre de öğrenme bireylerin belirlenmiş hedefler doğrultusunda gözlenebilir ve ölçülebilir davranışlar sergileyebilmeleri olarak yorumlanmıştır [3].

2.2.3. Oluşturmacı (Constructivist) öğretme anlayışı

Öğrenme konusunda oluşturmacı (constructivist) görüş, üretici öğrenme, keşfederek öğrenme ve duruma bağlı gibi teorilerin bir araya gelmesiyle oluşan oldukça yeni bir görüştür. Bütün bu görüşler arasında ortak nokta, bireylerin aktif olarak gerçek durumlar karşısında ve gerçek problemi çözerken kendi bilgilerini oluşturmaları ilkesine dayanmaktadır. Bu görüşe göre, ayrıca, bireylerin kendi bilgilerini etraflarındaki bireylerin yardımıyla oluşturdukları kabul edilmektedir [3].

Bu görüşe göre, yeni bilgilerin oluşturulması sırasında olayı düşünmek ve eski tecrübelerimizi yorumlamak çok önemli bir yer tutmaktadır. Her kişinin kendi bilgi, beceri ve tecrübelerinin olduğu göz önüne alındığında, herkes kendi bilgisini oluşturur demek mümkün olmaktadır. Öğrenmeyi bu durumda yeniden tanımlamak gerekecek olursa, öğrenme eski bilgilerimizin tecrübelerimiz ışığında yeniden yorumlanarak yeni bir hale getirilmesidir diyebiliriz [3].

2.3. Bilgisayar Destekli Öğretimin Olumlu Yönleri

BDÖ' nin olumlu yönleri aşağıda sıralanmıştır.

2.3.1. Öğrenme hızı

BDÖ'nün sunduğu en önemli fayda, belki de, öğrencilerin kendi öğrenme hızlarına uygun olarak konuyu işlemeleri ve gerek duyduklarında aynı konuyu tekrar çalışma olanağı bulabilmeleridir. Özellikle yavaş öğrenen öğrenciler için BDÖ uygulamaları, bu öğrencilerin düzeyine ulaşmasında önemli bir avantaj sağlamaktadır. Diğer taraftan, öğrenilecek konuyu hali hazırda bilen ya da sınıftaki diğer öğrencilere oranla daha hızlı öğrenen öğrenciler ise, diğer öğrencileri beklemeden bir sonraki konuya geçebilirler [12-14].

2.3.2. Katılımcı öğrenme

Her ne kadar BDÖ uygulamaları öğrencilerin kendi kendilerine ve kendi öğrenme hızlarına uygun öğrenme ortamları sunsa da, grup çalışmasını destekleyebilmesi açısından da etkin materyallerdir. Birçok BDÖ yazılımı, öğrencinin verdiği cevaplar doğrultusunda dersi sunar ya da öğrenciye belli aralıklarla dönüt sağlar. Bu yüzden, BDÖ ortamındaki her öğrenci aktif şekilde derse katılır ve dersteki performansını gösterebilme imkanı vermesi ve öğrenciye dönüt sağlayabilmesi nedeniyle, BDÖ ortamları öğrencinin derse katılımını sürekli hale getirir [12-14].

2.3.3. Öğretimsel etkinliklerin çeşitliliği

Diğer materyallerle karşılaştırıldığında, görsel–işitsel öğelerin en etkin kullanılabilirdiği ortam BDÖ ortamıdır. Öğretim ortamının farklı etkinliklerle zenginleştirilmesi, öğrencinin başarıya ulaşmasında önemli bir etkidir. İşte bu bakımdandır ki, BDÖ ortamları, sağladıkları öğretimsel etkinliklerin niteliği ve niceliği açısından en etkin ortamlardır [12-14].

2.3.4. Öğrenci etkinliklerinin ve performansının izlenebilmesi

BDÖ ortamındaki bir öğrencinin bir konu üzerinde harcadığı zaman ve gösterdiği performans, bilgisayar tarafından kayıt edilebilir ve istendiği zaman öğretmenin kullanımına sunulabilir. Geleneksel öğrenme ortamlarında, öğretmenin her öğrencinin performansını gözlemlemesi ve buna bağlı olarak öğrenciyi yönlendirmesi oldukça zordur. Özellikle kalabalık sınıflarda öğretmenin bu etkinlikleri başarması neredeyse imkânsızdır. Bu bakımdan, BDÖ ortamının sunduğu bu özellik, öğretim etkinliğinin geliştirilmesi için önemli bir unsurdur [12-14].

2.3.5. Zamandan ve ortamdan bağımsızlık

Geleneksel öğretim ortamlarındaki öğrenciler, belli konuları belli zaman dilimleri içinde öğrenmeli ve belirlenmiş öğretimsel etkinlikler yine belirli zaman dilimleri içinde gerçekleştirilmelidir. Diğer taraftan, BDÖ ortamındaki bir öğrenci istediği öğretimsel etkinlikleri istediği zaman, ders saati dışında kalan zamanlarda da, uygulayabilir ya da tekrar edilebilir. Hatta bu etkinlikleri evinde, bilgisayar başında uygulama şansı bulabilir [12-14].

2.4. Bilgisayar Destekli Öğretimin Olumsuz Yönleri

BDÖ' in olumlu yönlerinin yanı sıra birtakım olumsuz yönleri de vardır.

2.4.1. Öğrencilerin sosyo-psikolojik gelişmelerini engellemesi

Bazı uzmanlara göre, bilgisayarların öğretimi bireyselleştirebilmesi, öğrencinin sınıf içinde arkadaşları ve öğretmenleriyle olan etkileşimini azaltmaktadır. Başka bir deyişle, yazılımların görsel-işitsel özelliklerinden dolayı çocuğun ilgisini çekmesi ve özellikle de eğitimsel oyunlarda çocuğun saatlerce bilgisayar başında kalması gibi özellikler nedeniyle, çocuğun yaşlılarıyla ve diğer bireylerle olan etkileşimini azaltmakta ve bu durum çocuğun sosyo-psikolojik gelişimini olumsuz yönde etkilemektedir [12-14].

2.4.2. Özel donanım ve beceri gerektirmesi

Her şeyden önce, bir öğretim yazılımının kullanılabilmesi için mutlaka gerekli donanımın bulunması gerekir. Sınıfların ya da okulların BDÖ için gerekli donanımlara erişimi bazen zor ve pahalı bir süreç olabilir. Bunun yanında, öğretimsel yazılımların kullanılabilmesi için bilgisayarlara ek olarak özel donanımlara da ihtiyaç duyulabilir. Bunu yanında, diğer öğretim materyallerinin aksine, BDÖ materyallerinin kullanımı için hem öğrencilerin hem de öğretmenlerin bazı özel bilgi ve becerilere sahip olması gerekir. Bu gereklilikler, okul yönetimine büyük bir maddi yük getirmektedir [12-14].

2.4.3. Eğitim programını desteklememesi

Öğretimsel yazılımlar, diğer öğretim materyalleri ile karşılaştırıldığında, öğretmen tarafından geliştirilmesi zor olan, hazırlanması uzun süren ve geliştirilmesi pahalı olan materyallerdir. Bu yüzden, piyasadaki yazılımların birçoğunun eğitim programıyla bir tutarlılık göstermemesi, BDÖ'min sahip olduğu sınırlılıkların başında yer alır [12-14].

2.4.4. Öğretimsel niteliğin zayıf olması

Eğitim yazılımının türü ne olursa olsun (alıştırma-uygulama, benzetim vb.), her türlü

yazılım öğretim tasarımı ilkelerine uygun olarak geliştirilmelidir. Bu gerçeğe rağmen, piyasadaki yazılımların büyük bir çoğunluğu bu nitelikten yoksundur. Özellikler bazı yazılımlar, yazılı materyallerin elektronik ortama aktarılmış şeklerinden öteye gitmemektedir. Diğer taraftan bazı yazılımlar ise, hedeflenen öğrenci grubunun pedagojik özelliklerine uygun olmayan öğretim tasarımları üzerine geliştirildiği için öğretimsel etkinliği düşük olan yazılımlardır. Piyasada öğretimsel niteliği yüksek olan yazılımların az olması, BDÖ'in sahip olduğu diğer bir sınırlılıktır [12-14].

2.5. Bilgisayar Destekli Öğretimde Öğretmenin Rolü

BDÖ, var olan eğitim sistemini terkedip yepyeni bir sistemine geçmek anlamı taşımamaktadır. BDÖ, öğretmenin geleneksel ders işleme yöntemlerini zenginleştirici ek bir yöntem fırsatıdır. BDÖ her dersin saatinde uygulanması gereken bir metot da değildir. Aslında BDÖ uygulamalarının biçimi, süresi ve amaçları tamamen öğretmenin inisiyatifindedir. Çünkü BDÖ yalnızca bir araçtır. Nasıl, ne zaman ve ne amaçla kullanılacağı öğretmenin ilgi, istek ve hayal gücüne bağlıdır [15,16].

BDÖ uygulamalarında öğretmen, bilgiyi aktaran kişi olmaktan çıkar ve bilgiyi bizzat arayan öğrenciyi yönlendirici ve yol gösterici kişi olur. Hatta öğretmen de zaman zaman öğrencileriyle birlikte öğrenen rolünü üstlenecektir. BDÖ' de öğretmen, öğrencileri için bilgisayar teknolojisini kullanarak araştırma yapacakları konuları belirleyen, onların bu çalışmalarını organize edip yönlendiren, bulgularını analiz edip sonuçlar çıkartmalarını isteyen bu sonuçları sunmalarını teşvik eden kişidir. Yine BDÖ' de öğretmen, bilgisayar teknolojisini sınıfında kabul eden, bu güçlü teknoloji aracılığıyla konu anlatımını, ders işleme biçimlerini zenginleştiren ve böylece öğrencilerine çağdaş eğitim olanaklarının kapısını açan eğitimcidir [15,16].

2.6. Bilgisayar Destekli Öğretimde Öğrencinin Rolü

BDÖ uygulamalarında öğrenci, problem belirleyici, bu problemi çözme yolunda yöntemler geliştirici ve bir bakıma kendi öğretmenliğini yapan öğrenci rolünü alan

aktif öğrencidir. Bilgiye öğretmen ya da kitapların ötesinde bilgi teknolojilerinin sunduğu kaynaklara aracılığıyla da ulaşabileceğini farkedene öğrenci, sorgulamak, araştırmak ve analiz etmek isteği duyar. Çağdaş bilgisayar teknolojisinin kullandığı bir okul ortamında öğrenci, aradığı bilgiye hızla erişebildiği gibi, bu bilgiyi verimli ve güçlü bir şekilde kullanabilir. Bilgisayar teknolojisinin sağladığı son derece renkli, ilginç ve merak uyandıran medya aracılığıyla öğrenmenin keyifli bir aktivite olacağını hisseden öğrenci, belirlenen eğitim hedeflerinin çok daha fazlasına ulaşabilir [15,16].

BDÖ ile öğrenim pasif bir aktivite olmaktan çıkar ve aktif bir deneye dönüşür. Öğrenci değişik kaynaklardan elde ettiği bilgiyi ilişkilendirir ve sonuçlar çıkarır. BDÖ ortamı geleneksel eğitim ortamından çok, gerçek hayata yakın bir ortamdır. Çünkü işlenen konular birbirinden bağımsız değildir. Çözülmesi gereken problem, oluşturulması gereken çözüm, eğitimin tüm alanları ile ilgilidir. Bireysel çalışmadan çok grup çalışması ön plana çıkar. Kısacası, öğrenci, bilgi çağında iş yaşamında karşılaştığı problemleri ileride nasıl analiz edecek, çözecek ve çözümlerini sunacaksa, BDÖ sırasında da aynı yaklaşımla çalışır ve öğrenir [15,16].

2.7. Bilgisayar Destekli Öğretimin Öğrencinin Gelişimine Etkisi

Araştırmalara göre, aktif olarak bilgisayar destekli öğretim olanaklarından yararlanan öğrencilerin standart değerlendirme metotlarıyla yapılan incelemelerde başarılı bulunmalarının yanı sıra, genellikle ölçülmeyen birçok beceriyi de kazandıkları saptanmıştır. Bu becerilerden bazıları şunlardır [10] :

- Bağımsız öğrenme ve öğrenme sürecini kendiliğinden başlatabilme.
- Paylaşarak ve ortak çalışabilme.
- Bilgiyi aktif şekilde araştırma ve sunma.
- Karmaşık olaylarla ve problemlerle kolayca baş edebilme.
- Kendine güven.
- Teknolojiyi her an ve her yerde etkili biçimde kullanabilme.

- Kendi ilgi ve beceri alanını rahatça belirleyebilme.
- Geleceğe pozitif bakma.
- Sorumluluklarını çabuk öğrenme ve üstlenebilme.
- Yaratıcılığın ortaya çıkmasını sağlar.
- Sosyal iletişimde bulunma yeteneğini geliştirir.
- Her öğrencinin kendi hızlarında ve düzeylerinde ilerleme olasılığı verir.
- Kendine güveni artırır.
- Problem çözme ve dikkatini bir problem üzerine yoğunlaştırma yeteneğini geliştirir.
- Öğrencinin öğrenme zamanından tasarruf sağlar.
- Belgeleme, dosyalama ve belgelere başvurma alışkanlığını kazandırır.
- Önceki çözümleri araştırıp bunları yeni bir çözüm için kullanabilme yeteneğini geliştirme, yeni çözüm bulmasını sağlar.
- Matematik ve dil yeteneğini geliştirir.
- Paylaşım duygusunu geliştirir.
- Daha çok bilgiye ulaşma imkânı verir.
- Anında dönüt sağlandığı için kaçırılan ders veya konu öğrenci tarafından tekrar edilebilir.
- Benzeşimler sayesinde öğrencilere özgü mekânlar sağlar.

2.8. Bilgisayar Destekli Öğretimin Öğretmene Faydaları

Bilgisayar teknolojisini benimseyerek bu teknolojiyle tanışmanın ve kullanmanın öğretmenlere başlıca faydaları şunlardır [15]:

- Eğitimde bilgi çağını yakalamak.
- Hemen her kurumda aranan bir mesleki yetenek kazanmak.
- Yeni bilgiler öğrenme ve kendini geliştirme imkânı.
- Öğretmenlik görevini daha etkili bir biçimde yerine getirebilmek.
- Öğrencilerin artan istek ve katılımları sayesinde derslerin kolay işlenebilmesi.
- Sınıf performansının artması,

- Öğrencinin derse aktif katılımının sağladığı için öğretmenin işini kolaylaştırır.
- Öğretmenin farklı seviyelerdeki öğrencileri izleyerek onlara ayrı ayrı zaman ayırabilme olasılığı sağlar.
- Kanaat için ek alternatif sunar.
- En sıkıcı dersleri kolay ve zevkli hale getirerek öğretmene yardımcı olur.
- Konuyu kaçıran öğrencilere, öğretmeni engellemeden konuyu tekrar etme olanağı sağlanır.

2.9. Bilgisayar Destekli Öğretim İçin Temel Gereksinimler

Öğretmenin okulunda ve sınıfında BDÖ uygulayabilmesi için temel gereksinimler şunlardır:

- Bilgisayar Donanımı
- Bilgisayar Yazılımları
- İnsan Kaynakları
- Eğitim Programında Düzenlemeler

2.9.1. Bilgisayar donanımı

Donanım, bilgisayar teknolojisinin mekanik ve elektronik kısımlarına verilen addır. Başlangıçta, BDÖ uygulamaları için gerekli donanım bilgisayar, yazıcı ve projeksiyon aletinden ibarettir. Öğretmen ve okul BDÖ konusunda tecrübelendikçe ve eğitimin bu boyutunu benimsedikçe diğer çevre birimleri de eklenerek kullanılabilir [10].

2.9.2. Bilgisayar yazılımları

Yazılımlar bilgisayarları işleten ve onlardan gerçek anlamda yararlanmamızı sağlayan araçlardır.

Uygulama yazılımları etkili BDÖ çalışmaları için gerekli araçlar arasındadır. Eğitim yazılımları ise, BDÖ için temel yazılımlardır. Bu yazılımlar özellikle öğrenci için hazırlanırlar. Verilmek istenen bilgiyi renkli ve çekici bir biçimde ve ses, grafik, yazı, canlandırma, film, resim gibi değişik medyaları kullanarak iletirler. Bu yazılımların en önemli özellikleri etkileşimli olmalarıdır. Yani öğrenci eğitim yazılımlarını kullanarak pasif bir izleyici değil, aktif bir kullanıcıdır. Deneme yanılma ile öğrenir, kendi hızında ilerler ve bilgiye anında ve tekrar tekrar ulaşabilir.

BDÖ uygulayan öğretmen yazılım konusunda da uzman olmak zorunda değildir. Günümüz yazılım teknolojisi özellikle son yıllarda olağanüstü yol katetmiş ve artık üç yaşındaki çocukların bile kolaylıkla kavrayıp uygulayabildiği basitliğe indirgenmiştir. İşletim sistemi zaten bilgisayarın bir parçasıdır ve kullanımı çok kısa bir eğitim gerektirir. BDÖ için gerekli uygulama yazılımları aynı basitliktedir. Bir uygulama yazılımının nasıl çalıştığı kavrandığında, diğer tüm uygulama yazılımları neredeyse eğitim görmeden, kendiliğinden keşfedilebilir. Eğitim yazılımları ise çocukların rahatlıkla keşfedip kullanabileceği biçimde hazırlanmışlardır.

Bu yazılımların öğrenilmesi zaten söz konusu değildir. Eğitim yazılımlarını kullanmak, kapısı bilgisayar ekranından açılan canlı ve çekici bir dünyaya yolculuğa çıkmaya benzetilebilir [10].

2.9.3. İnsan kaynakları

Bilgisayar destekli eğitimin okullarda uygulanabilmesi için üç tür insan kaynağına ihtiyaç vardır: 1. Yönetim 2. Teknik destek 3. Öğretmen.

Yönetim, okul idaresinden sorumlu kişi ya da kişilerdir ve BDÖ'nün okulda uygulanması için gerekli kaynakları temin etmekle ve BDÖ'nün okulda uygulanması için gerekli kaynakları temin etmekle ve BDÖ'yü desteklemekle bu çalışmalara katkıda bulunurlar.

Öğretmen BDÖ uygulamalarında en önemli insan kaynağıdır. Çünkü BDÖ, ancak öğretmen istekli olduğu sürece ve onun yaratıcılığı boyutunda gerçekleştirilebilir. Kendi ders konularının hangilerinin BDÖ ile işlenebileceğini araştırmak, kullanılacak eğitim ve uygulama yazılımlarına karar vermek, uygulama mekân ve zamanını belirlemek, BDÖ ile yapılan çalışmaları yönlendirmek ve bu çalışmaların öğrenciler üzerindeki etkilerinin takibini yapmak tamamen öğretmenin inisiyatifindedir.

2.9.4. Eğitim programında düzenlemeler

BDÖ okullara ve sınıflarda kullanılırken, mevcut eğitim sisteminde de bunun için gerekli düzenlemelerinin yapılması şarttır.

Bu, köklü bir değişiklikten çok, her ders için sene başında çıkartılan eğitim planında bilgisayar destekli öğretim araçlarının hangi konularda, ne biçimde ve ne süreyle kullanılacağına belirlenmesi ve bu uygulamaların plana sadık kalınarak yerine getirilmesi şeklinde gerçekleştirilebilir.

Burada anahtar kişi yine öğretmendir. Kendi ders konularında okulda mevcut BDÖ araçlarının hangilerini ve ne şekilde kullanacağına öğretmen karar verir. Geleneksel ders işleme metot ve araçlarına bilgisayar teknolojisinin getireceği zenginlik, öğrencilerin artan ilgisi ve başarı oranı, bu teknolojiyi küçük ölçeklerle de olsa kullanan öğretmenlerin mesleki yaşamlarına yeni bir boyut katacaktır [10].

2.10. Bilgisayar Destekli Öğretimde Yöntemler

Bilgisayar destekli öğretimde, bilgisayar çok çeşitli biçimlerde kullanılabilir ve farklı amaçlara yönelik ders yazılımları hazırlanabilir. Bu yazılımlar şunlardır:

2.10.1. Alıştırma ve tekrar

Bilgisayarın en yaygın uygulamalarından biri, işlenmiş konularla ilgili alıştırma ve

tekrar yaptırma amacı ile kullanılmasıdır. Alıştırma ve tekrar yaptırmanın amacı; bilgi ve becerilerin pekiştirilmesi; öğrenmenin kalıcılığının sağlanması; üst düzey davranışların (analiz, sentez vb.) öğrenilmesine zemin hazırlanması içindir [17].

2.10.2. Bire-bir öğretim

Birçok araştırma en ideal öğrenmenin bir öğretmenin yalnızca bir öğrenciyle çalışması sonucu gerçekleşen öğrenme olduğunu göstermektedir. Günümüz koşullarında böyle bir eğitim sisteminin olamayacağı açıktır. Öte yandan bilgisayarların okullarda kullanılmaya başlamasıyla bire-bir öğretim uygulamaları da bir ölçüde başlamıştır. Bu uygulamalar, bir konu ile ilgili olgu, kavram, yöntem, ilke, genelleme ve bilimsel yasaların bilgisayardan öğrenilmesini amaçlamaktadır [17].

2.10.3. Problem çözme

Özellikle matematik ve fen bilimleri ile ilgili derslerde öğrencinin, problem oluşturma ve çözme yeteneğini geliştirmek büyük önem taşımaktadır. Bilgisayar kullanarak problem çözme yeteneği geliştirilebilir [17].

2.10.4. Deney ve laboratuvar çalışması (benzetim)

Laboratuvar çalışmaları, genellikle ilke, kural ve bilimsel yasaları gerçek ya da yapay durumlarda öğretme etkinliğidir [17].

2.10.5. Eğitici oyunlar

İyi planlanmış bir oyunun öğretim açısından önemi büyüktür. Eğitici oyunlar, çoğu kişide bulunan ilgiden yararlanarak, ilginç öğrenme ortamları yaratmaktır. Eğitim amaçlı oyunlar; yaratıcı oyunlar ve öğretici oyunlar olarak iki gruba ayrılabilir. Bilgisayar destekli öğrenmede deneme, keşfetme, problem çözme ve karar vermeyi gerektiren oyunların yanısıra okuma ve anlama becerisini geliştiren oyunlardan da yararlanma olanağı vardır [17].

2.10.6. Bilgi deposu

Bilgisayarlardan sözlük, ansiklopedi veya kütüphane erişim terminali olarak yararlanmak mümkündür. Çoklu ortam teknolojilerindeki hızlı gelişme ve fiyatların düşmesi sonucu, bilgisayarın bilgi deposu olarak kullanımı yaygınlaşmaktadır [17].

2.11. Bilgisayar Destekli Öğretim Uygulamaları

Daha önce belirtildiği gibi, bilgisayar destekli Öğretim uygulamaları genel sistemin içerisinde ve bu sistemle uyumlu biçimde yürütülür. Her derste ya da her konuda BDÖ çalışması yapmak mümkün değildir. Burada önemli olan, öğretmenin eğitim yılı boyunca bazı ders konularını BDÖ aracılığıyla işlemeye çalışmasıdır. Bu çalışmalar da, senelik ders planında önceden belirlenmelidir.

Öğretmen, öncelikle, hem işlemesi gereken ders konularına, hem de elindeki BDÖ araçlarına yani donanıma, uygulama yazılımlarına ve eğitim yazılımlarına bakarak kendisine “bu araçlarla bu bilgiyi öğrencilerime nasıl aktarabilirim?” sorusunu sormalıdır. Ders planları mevcut BDÖ araçlarına göre değiştirilmemeli, bu araçlar ders planlarına uygun bir şekilde kullanılmalıdır. BDÖ yönetsel bazda incelendiğinde temel olarak 4 çeşit olarak uygulanabilir.

2.11.1. Laboratuvar yöntemi

Eğitimde bilgisayar teknolojisini kullanmanın en basit ve geleneksel yöntemi bir eğitim kurumuna laboratuvar kurmaktır. Bu yöntemin asıl amacı hedef kitleye bilgisayar okur-yazarlığı kazandırmak ve dersleri laboratuvar ortamında mümkün olduğunca interaktif olarak sunmaktır. Laboratuvarın kurulduğu eğitim kurumunda öğrenci sayısı laboratuvarda bulunan bilgisayar sayısından fazla olduğundan, öğrenciler dönüşümlü olarak bilgisayarlardan yararlanabilmektedir. Dolayısıyla; bu yöntemde öğrenci bilgisayar ile fazla çalışma imkânı bulamamaktadır [10,17].

2.11.2. Her sınıfa PC yöntemi

Bu yöntemde; her eğitim sınıfına birer adet bilgisayar, sunum cihazı ve gerekli çevre birimleri kurulur. Ayrıca okul bir network ortamı ile bütünleştirilir. Böylece her ders teknolojiyle bütünleştirilmiş olur. Bu yöntemde amaç; öğrencilere bilgisayar okuryazarlığı vermek değil, her dersi bilgisayar teknolojisiyle bütünleştirerek öğrenmenin kalitesini artırmaktır. Eğitici dersine girmeden önce derse ilgili konuda gerekli materyalleri bilgisayar ortamında hazırlar ve bilgisayar sistemini kullanarak öğrencilerine aktarır. Böylece öğrenci her dersi görerek öğrenmiş olur [10,17].

2.11.3. Kişisel PC yöntemi

Bu yöntemde; her öğrencinin ve öğretmenin taşınabilir bir bilgisayarı vardır. Ayrıca eğitim ortamı bir ağ ortamına sahiptir. Öğrenci tüm ders materyallerini, ödevlerini ve ders hazırlıklarını kişisel bilgisayarında yapar. Eğitim ortamına geldiğinde öğrenci kişisel bilgisayarını okulun ağ yapısına birleştirir ve derse katılır. Eğitici ise ders ile ilgili tüm hazırlıklarını kendi kişisel bilgisayarında yapar ve derse girdiğinde kendi kişisel bilgisayarını okulun ağ ortamına birleştirir ve dersini anlatır. Eğitici ve öğrenci arasındaki tüm haberleşme elektronik ortamda yapılır. Ayrıca bu yöntemle eğitici ve öğrenciler evlerinden video konferans yoluyla ders yapabilirler. Bu yöntem diğer yöntemlere göre en ideali, ancak en pahalısıdır [10,17].

2.11.4. İnternet yoluyla eğitim yöntemi

Bu yöntem senkron ve asenkron olarak iki biçimde gerçekleştirilebilir. Senkron yöntemde; eğitici ve öğrenciler, gerçek zamanlı olarak video konferans, chat gibi uygulamalarla günün belirli saatlerinde mekândan bağımsız olarak bir sınıf ortamındaymış gibi eğitimi gerçekleştirirler. Asenkron yöntemde; dersin içeriği İnternet ortamına aktarılır. Öğrenciler zamandan ve mekândan bağımsız olarak İnternet sitesine bağlanarak eğitimlerini gerçekleştirirler. Bu yöntem açık öğretim programları için kullanılabilir [10,17].

2.12. Bilgisayar Destekli Öğretimdeki Sorunlar

Bilgisayar destekli öğretim yöntemi, günümüzde Dünyanın gelişmiş ülkelerinde, gelişmekte olan ve az gelişmiş ülkelerin eğitim sistemlerinde kullanılmaktadır. Bu yöntem yetişkin öğrenme özellikleri açısından da uygun ve etkili bir öğretim yöntemidir. Ancak ilgili çalışmalar incelendiğinde bu yöntemin uygulanmasının beraberinde birtakım sorunları da getirdiği görülmektedir. Bunlar [18];

1. Okulların, nitelikli eğitim verip vermediğine bakılmaksızın, bilgisayarla donatılması yoluna gidilmektedir.
2. Bilgisayar yazılımlarının sayısı sınırlıdır. Ders programları ile yazılımların içeriği arasında tutarlılık sağlanamamakta, hazır paket programların kalitesi tartışma konusu olmaktadır.
3. Ders yazılımlarının istenilen kalitede ve amaca uygun olarak hazırlanması uzun zaman almakta ve ekip çalışması gerektirmektedir.
4. Bilgisayar sistemleri pahalıdır. Eğitim sistemlerinin, özellikle okulların böyle pahalı bir uygulamayı nasıl yüklenebileceği tartışma konusudur.
5. Bilgisayar eğitimi, bilgisayarla eğitim ve bilgisayar destekli öğretim kavramları birbirlerinin yerine kullanılmakta ve bu yüzden, girişimlere ve uygulamalara karşı olumsuz tepkilerin doğmasına neden olmaktadır.
6. Öğretmenlerin ve yöneticilerin gerek hizmet öncesi, gerekse hizmet içi eğitimlerle yeterince yetiştirilip yetiştirilmediği tartışma konusudur.
7. Bilgisayar destekli öğretimin henüz yeni olması ve genç kuşaklar üzerindeki olumlu ve olumsuz etkileri saptanacak aşamaya gelmediği için ve konu ile ilgili araştırma sayısının çok az olması nedeni ile ilgili korkular sürmektedir.
8. Bilgisayarların öğretimde etkin bir şekilde kullanımını donanımına bağlıdır.
9. BDÖ bazı amaçlara ulaşmada etkilidir. Ancak bu tüm BDÖ programlarının tüm öğrenciler için etkili olduğu anlamına gelmemektedir.

2.13. Farklı Okul Kademelerinde Bilgisayar Destekli Öğretim

Bilgisayar, hemen her okul kademesinde yani her öğrenim aşamasında

kullanılabilecek bir araçtır. Okullarda bilgisayar sayısı oldukça az olmakla beraber okul dışında öğrencilerin bilgisayarla ilişkileri daha sıktır. Ancak bu sıklık çoğunlukla eğlence amaçlıdır. Özellikle ilk ve orta dereceli okullardaki öğrenciler bilgisayarı bir oyun aracı olarak kullanmaktadırlar. Evlerinde bilgisayarları olmadığı için bilgisayar olan ortamlarda oyun oynayan bu öğrencilere, oyun dışında eğitime katkısı olabileceğini göstermek bunu uygulamak öğrencinin, hem gelişimini hem de eğitimini olumlu yönde etkileyebilir. Örneğin ilkokulda okuyan 10 yaşındaki bir çocuğa bilgisayarda basit bir yazım editörü kullanarak ders programı hazırlatmak, günlük tutturmak, eğlenceli eğitim yazılımları yoluyla ders çalışmasını sağlamak veli açısından oldukça yararlı bir girişimdir. Ancak unutulmamalıdır ki eğitim yazılımının pedagojik yapısı, dili, kullanım kolaylığı gibi kaliteyi belirleyici özelliklerinin iyi olmasına özen gösterilmelidir. Öğrenci lise seviyesinde bilgisayarı artık sıkça kullanmaya başlaması artık yeni şeyler denemesine sebep olacaktır, programlama yapacaktır, oyun oynarken seçici ve abartısız olacaktır. Lise seviyesinde okul içi bir gazetenin öğrenciler tarafından bilgisayardaki yazım editörleri kullanılarak hazırlanması, öğrencilerin kendine olan güvenlerini arttıracaktır. Başarı duygusu ile gelişim görevlerini çok iyi bir şekilde yerine getirecektir. Ayrıca zekâ oyunları ile zihin jimnastiği yapmak yine çok faydalı olacaktır. Yine lise seviyesindeki öğrenciler evlerindeki bilgisayarlar yolu ile matematik, fizik, coğrafya, kimya, biyoloji, geometri gibi dersleri tekrar etme ve daha somut anlama şansına sahip olabilirler. Üniversite ise bilgisayar destekli eğitimin uygulanması en elverişli ortamdır. Çünkü artık öğrenci olgunlaşmıştır ve karakteri oturmuştur. Yani temel gelişimini genel anlamda tamamlamıştır. Bilgisayar ortamında verilen bir üniversite eğitimi sayesinde çok daha verim alınması muhtemeldir [18].

2.14. Dünya’da Bilgisayar Destekli Öğretim

Dünya’da örgün eğitimde Bilgisayar Destekli Öğretim uygulamaları, ülkelerin eğitimde bilgisayar uygulamaları ve Bilgisayar Destekli Öğretim’ e geçiş tarihleri ve uygulanan önemli projeler açısından ele alınıp değerlendirildiğinde; ABD’nin bu konuda Dünya ülkeleri arasında ilk sırada yer aldığı, bu ülkeyi İngiltere, Fransa,

Almanya ve İsveç gibi ülkelerin izlediği görülmektedir. Dünya’ da örgün eğitimde BDÖ uygulamalarında; a) serbest piyasa modeli, b) özel girişim devlet işbirliği modeli ve c) merkez yönetim modeli olmak üzere belli başlı üç modelin uygulandığı dikkati çekmektedir.

Dünya’da yetişkin öğretmen, asker, çiftçi, okur-yazar olmayan kesim, kamu görevlileri, özel eğitime muhtaç kimseler vb. eğitimde bilgisayar destekli öğretimden yararlanma konusunda yapılan araştırmaların geniş bir hedef kitleyi kapsadığı ve özellikler yetişkin okur-yazarlık programlarında BDÖ’ den yararlanma konusunda sayıca daha fazla araştırma ve uygulamaların gerçekleştirildiği söylenebilir.

Dünya’da hizmet içi eğitimde BDÖ’ den yararlanma konusunda önemli sayıda araştırma ve geliştirme etkinliklerine girildiği ve araştırmaların hedef kitlesini genel de öğretmenlerin oluşturduğu dikkati çekmektedir. Yapılan araştırmaların bulgularına göre, hizmet içi eğitimde BDÖ’ den yararlanmaya yönelik olarak hizmet içi eğitime katılan personel (öğretmen ve diğer) olumlu tutum ve davranış göstermektedirler. Personel görüşlerine göre hizmet içi eğitimde BDÖ’ den yararlanma yazılım, donanım, öğretmen yetiştirme gibi sorunları da beraberinde getirmekte ve uygulamalarda yetişkin öğrencilerin karakteristikleri ve öğrenme ilkelerine dikkat edilmesi gereklidir.

Dünya’da uzaktan eğitimde bilgisayar destekli öğretimden yararlanma konusunda yapılmış olan araştırma bulgularına göre; uzaktan eğitimde BDÖ’ den yararlanma, bireysel, bağımsız, sürekli ve etkileşimli öğretimi gerçekleştirmekte ve öğretimde yaygınlık, çeşitlilik, hızlilik, işlevsellik ekonomiklik vb. olanakları sağlayabilmektedir [19].

2.15. Türkiye’de Bilgisayar Destekli Öğretim

Türkiye’ de örgün eğitimde bilgisayar eğitimi ve BDÖ ilişkin çalışmalar 1984 yılında Milli Eğitim Bakanlığı tarafından başlatılmıştır. MEB Bilgisayar Eğitimi ve Bilgisayar Destekli Eğitim projesinin gelişim aşamaları a) 1984-1988 (Ön hazırlık

çalışmaları) b) 1989-1991 (Bilgisayar Destekli Eğitim Çalışmaları) ve c) 1992 yılından günümüze kadar gerçekleştirilen çalışmalar (Uygulanmasına 1998-1999 Öğretim Yılı'nda başlanması kararlaştırılan "Eğitimde Çağı Yakalamak 2000 Projesi" de bu dönem açısından ele alınabilir).

Türkiye' de örgün eğitimde bilgisayar kullanımı ve BDE konusunda, MEB' den başka Bilim ve Teknolojiden Sorumlu Devlet Bakanlığı ve Türkiye Bilimsel Teknik Araştırma Kurumu (TÜBİTAK)'nun da bir takım etkinliklerde buldukları ve MEB ile ortaklaşa çalışmalar yürüttükleri görülmektedir. TÜBİTAK' ın BDE konusunda; eğitim kuruluşları ile ortak projeler hazırlayıp, uygulamak, araştırma, donanım ve yazılım üretimi (veya ithal yoluyla edinilmesinde) danışmanlık ve yurt dışındaki kuruluşlarla işbirliği yapmak gibi amaçları bulunmaktadır [11].

Türkiye'de örgün eğitimde BDE uygulamalarının değerlendirilmesinde; a) MEB "BDE Projesi" ve "Eğitimde Çağı Yakalamak 2000 Projesi", b) Türkiye' de örgün eğitimde BDE konusunda yapılan araştırma-geliştirme etkinlikleri ve c) Gelişmiş ülkelerdeki uygulamalar gibi ölçütler esas alınabilir. Bu üç ölçüt açısından ele alınıp değerlendirildiğinde; MEB tarafından yürütülen ve yaklaşık 15 yıllık bir geçmişe dayanan BDE projesinin hedeflere ulaştığı söylenemez. Türkiye' de örgün eğitimde BDE konusunda yapılan araştırma-geliştirme etkinliklerinin büyük bir çoğunluğunu üniversitelerde lisansüstü tez niteliğindeki araştırmalar oluşturmaktadır. Ancak araştırmaların çok büyük bir bölümünün betimsel nitelikte oldukları, deneysel ve uygulamanın başarı şansını artırmaya yönelik araştırmalara çok az yer verildiği dikkati çekmektedir. Yine örgün eğitimde BDE' den yararlanma konusunda gelişmiş bazı ülkelerdeki araştırma-geliştirme etkinlikleri ve uygulamaları ile kıyaslandığında, Türkiye'nin bu açıdan bir hayli geride olduğu söylenebilir. Ancak bu kıyaslamayı yaparken gelişmekte olan ülkelerde eğitimde karşılaşılan ve genelde maliyet ve ekonomik problemler konusunda yoğun problemleri de gözardı etmemek gerekir [11].

Yetişkin eğitimi konusunda Türkiye'deki uygulamalar açısından bakıldığında; Türkiye'nin en eski yetişkin eğitimi sistemine sahip bulunan ülkeler arasında

bulunduđu ve yaygın (yetişkin) eğitimin anayasa, kanun, kanun hükmünde kararname ve yönetmeliklerde yer aldığı ve bu eğitim ile ilgili bir takım politikaların ve MEB'nin koordinatörlüğünde bir takım projelerin gerçekleştirildiđi görülmektedir. MEB'nin yanısıra çeşitli bakanlıklar, özel ve gönüllü kuruluşlar ve özel sektörün yetişkinlere yönelik eğitim programları bulunmaktadır. Türkiye' de bilgisayar konusunda ihtiyaç duyulan eğitimlerden birisi de yetişkin eğitimi olmasına rağmen, yetişkin eğitiminde bilgisayar ve BDE' den yararlanma konusunda gerçekleştirilmiş araştırma sayısı yok denecek kadar azdır. Yetişkinlerin bilgisayar eğitimi ve yetişkin eğitiminde BDE'den yararlanma konusu, bu konuda önemli gelişmeler kaydetmiş ülkelere oranla, eğitimde bilgisayar uygulamalarına henüz yeni başlamış olan Türkiye açısından daha farklı yöntem ve politikaların izlenmesini gerektirmektedir. Türkiye'de yetişkin eğitiminde BDE'den yararlanma sorunu "yetişkin eğitimi merkezi" gibi merkezi bir kurumun organizatörlüğünde bir proje olarak ele alınmalı ve projenin yakın ve uzak hedefleri ve izlenecek strateji, politika ve yöntemler belirlenmelidir. Yetişkin eğitiminde BDE projesi hizmet içi eğitim programları ve okur-yazarlık, sosyal kültürel kurs programları olmak üzere iki boyutlu olarak ele alınmalı ve projedeki öncelik bilgisayar eğitimine verilmeli, daha sonraki aşamada ise BDE uygulamalarına geçilmelidir [11].

Yetişkin hizmet içi eğitimi, Dünya'nın gelişmiş ülkelerinde olduđu kadar, gelişmekte olan bir ülke olarak Türkiye için de oldukça önem taşıyan bir konudur. Konu ile ilgili olarak yapılan araştırmaların bulgularına göre, Türkiye'de gerçekleştirilen hizmet içi eğitim etkinliklerinde eğitim gereksinimi belirlemeden, planlama, uygulama ve değerlendirmeye kadar tüm aşamalarda birçok sorunlar vardır. Bilgisayar ve bilgisayar destekli eğitim, hizmet içi eğitimde öğrenme-öğretme süreçlerinde geleneksel öğretim yöntemlerini güçlendirebilecek ve sistemi tamamlayabilecek yeni teknolojik uygulamalardan birisidir. Türkiye'de kamu kesiminde hizmet içi eğitimde BDE uygulamalarına geçmeden önce kurumsal ve bilimsel birtakım araştırmaların yapılması ve kamu kesimindeki mevcut koşullar ve olanaklar da dikkate alınarak hizmet içi eğitimde BDE için bir model geliştirilmesi gereklidir [11].

Türkiye’ de çoklu sunum ortamları ile iletişim ve uzaktan eğitim teknolojileri gibi kavramlar çok yeni kavramlar olup, uzaktan eğitim teknolojilerinin bizzat geliştirilmesi konusunda henüz kapsamlı bir proje atılımı olmamıştır. Türkiye’de uzaktan eğitimde BDE’den yararlanma konusunda sınırlı sayıda da olsa başta Anadolu Üniversitesi AÖF olmak üzere, Fırat Üniversitesi, İTÜ, ODTÜ ve Başkent Üniversitesi gibi bazı üniversiteler tarafından birtakım proje ve araştırma-geliştirme etkinliklerinin gerçekleştirildiği görülmektedir [11,17].

Uzaktan eğitimde BDE’den yararlanma konusu genel olarak değerlendirildiğinde, Türkiye’ de bu konuda yapılan araştırma ve uygulamaların ve uzaktan eğitimde kullanılan teknolojilere oranla oldukça yetersiz olduğu söylenebilir [11,19].

2.16. Eğitim Yazılımı Geliştirme

Yazılım geliştirmedeki yaklaşım, yazılımların öğrenci merkezli çalışmaya izin vererek, somuttan soyuta ve informal bilgidan formal bilgiye şeklinde eğitim konularını işleme zorunluluğudur.

Bir öğretmenin okula gönderilmeden önce bazı eğitimlerden geçirilmesi nitelikli eğitim için gereklidir. Her işte olduğu gibi eğitsel yazılım geliştirme etkinliği de bir takım ön hazırlıklar ve planlamalar gerektirir. Yazılımın öğretilecek içeriğe ilişkin tutarlı ve geçerli bilgilere sahip olması ve öğrencilerle uygun bir şekilde bilgi iletişimi kurması için değişik uzmanların katkısı bir ihtiyaçtır. Öğretmenler, konu alanı uzmanları, eğitim bilimciler, program geliştiriciler, tasarımcılar ve bilgisayar programcıları birlikte çalışarak hazırlayacakları yazılımı öğrenci üzerinde etkili kılabilirler [3].

2.16.1. Bdö yazılımı geliştirmede temel aşamalar

BDÖ yazılım geliştirme aşamalarını dokuza ayırabiliriz.

Ders hedeflerinin ve öğrenci gereksinimlerinin belirlenmesi

Bireylere kazandırılacak bilgi ve davranışlardan bir ders sonunda ulaşılabacak olanlar “hedef” ifadeleri haline getirilir. Hedeflerin belirlenmesi işlemi eğitimde program geliştirme uzmanlarının kontrolünde bir grup uzman tarafından yapılır. Hedefler, etkinliklerin türü ve sınırı hakkında bilgi kaynağıdır.

Yazılım rasyonelinin belirlenmesi ve doğrulanması

Ders hedefleri, kavram haritası ve öğrenci zorlukları göz önüne alınarak konunun bilgisayar olanaklarıyla nasıl kolay hale getirileceği düşünülür. Yazılım rasyoneli oluşturmada kavram ağı ve konu analizine bakarak ne tür mekanizmalarla konunun işleneceği üzerinde düşünülüp alan uzmanı ve öğretmenlerle görüş alışverişinde bulunulmalıdır.

Rasyonelin kavramsal ve fonksiyonel tasarıma dönüştürülmesi

Rasyonelde belirtilen kavramların bilgisayar ortamında hangi platform ve araçlarla gösterileceği, ekran objelerinin ekrandaki konuları ve estetik özellikleri, hangi ekran objesinin (simge, düğme, menü, pencere, grafik, resim) ne tür bir işlevi olacağı belirlenir.

Tasarımın gözden geçirilmesi

Kâğıt üzerindeki tasarımlarda görülebilecek aksaklıklar üzerinde rasyonel ve hedefler dikkate alınarak gerekli düzeltmeler yapılır.

Tasarımın model olarak programlanması

Tasarımın hızla inşa platformları kullanılarak bir model olarak programlanır.

Model programının deęerlendirilmesi

Biçimlendirmeye yönelik deęerlendirme, ürün geliřtirmenin belli ařamalarında yapılan ara deęerlendirmelerdir ve geliřtirme sürecinin amaca uygun gidip gitmedięine iliřkin bilgi verir.

Tam sürümün programlanması

Yazılımın özellikleri gereęi seçilecek programlama ortamı önemlidir. Programlama platformundan beklenen arabirim özellikleri yoğun olan ve görsel arabirimin tercih edildięi sistemler için nesne yönelimli platformlar daha uygundur.

Tam sürümün geçerlenmesi

Geçerleme çalışması önce laboratuvar koşullarında birkaç öğrenci ile yapılır. Eğer yazılım öğrencilere ve geliřtirme grubunda onaylanırsa ikinci ařama olan geçerleme çalışmasına geçilir. İkinci geçerleme çalışması yeni bir grup öğrenci ile yapılır.

Tam sürümün deęerlendirilmesi

Sınıf veya okulun bilgisayar ortamında yazılımın kullanımı farklı uygulamalar gerektirebilir. Gerçek ortamlarda yeterli sayıda öğrenci ile yapılan deęerlendirme çalışmalarının genellenebilirlik özellięi vardır. Öğrenci çalıştıracağı makine üzerinde veya kullanacağı laboratuvar da test edilmelidir.

3. JAVA PROGRAMLAMA DİLİ

Java programlama dili nesne yönelimli bir programlama dilidir.

3.1. Nesne Yönelimli Programlama

Nesne yönelimli programlama(Object Oriented Programming-OOP) da veri program kurgusunun en önemli parçasını oluşturur. Bu model ise fonksiyonları yaptıkları işlere göre gruplamak yerine programın en önemli parçası olan veriyi kullandıkları nesnelere içinde saklar ve gruplar. Nesneye yönelik olarak tanımlanmış fonksiyonlar ise sadece lokal nesnelere sakladıkları verilerle iş görürler. Java, Perl gibi diller...

Nesne Yönelimli Programlama tamamıyla geleneksel programlama yapısından ayrılır. Çünkü veri ve işlemleri bir tek veri yapısına yerleştirir. İşlevsel programlamada veriler ve işlemler ayrıdır. Veri, üzerinde işlem için prosedür ve fonksiyonlara gönderilir. Nesne Yönelimli Programlama bu yaklaşımdaki çoğu problemi çözer. Çünkü veriler ve işlemler aynı varlığın parçalarıdır.

Nesne Yönelimli Programlamanın en küçük veri yapısı nesnedir. Nesne, birbiriyle ilişkili durumlar ve işlemler kümesidir. Genellikle bir nesneye ait bir işlem gerçekleştiğinde nesneye ait durumlardan bazıları değişir. Aynı şekilde işlemler de bazı durumların değerine göre davranır [20].

3.1.1. Sarmalama (Encapsulation)

Sarmalama, kod ve veriyi birbirine bağlayarak dışarıdan gelebilecek müdahale ve kötüye kullanmalarına karşı onları koruyan bir mekanizmadır.

3.1.2. Kalıtım (Inheritance)

Kalıtım, bir nesnenin başka bir nesnenin özelliklerini devralmasıdır. İleri düzeyde türetilen bir alt sınıf, sınıf hiyerarşisinde atalarının tüm özelliklerini devralır.

3.1.3. Çok biçimlilik (polymorphism)

Çok biçimlilik, bir arabirimin genel bir eylemler sınıfı için kullanılmasını sağlar. Çok biçimlilik kavramı “bir arabirim, çok metod” şeklinde açıklanabilir.

3.2. Java'ya Giriş

Java platformu bilgisayar ağının varlığı da göz önüne alınarak uygulamaların/ programların farklı işletim sistemleri üzerinde çalıştırılabilmesi düşüncesiyle geliştirilmiş yeni bir teknolojidir.

Java platformu hem programlama dili, hem de bir ortam olarak düşünülebilir. İlk bilgisayarlardan bu yana birçok programlama dili geliştirilmiştir. Bunların arasında JAVA 1995 yılında ortaya çıkmış olan bir programlama dilidir.

Java 4. nesil dil ailesindedir. Java program dilinde yazılan program kodları derlendikten sonra byte-code adı verilen ara bir dile çevrilir. Bu ara dil ise derlenmiş olduğu halde bile her makine tarafından yorumlanmaya hazırdır. Çünkü *byte-code* olarak derlenmiş programı, bir makinenin anlayacağı dile çeviren, o makinede derlenmiş ve Java Virtual Machine (JVM) olarak adlandırılan ilave bir programdır. Bu program günümüzde hemen her makinede bulunmaktadır [20].

3.2.1. Java'nın tarihçesi

Java ilk olarak 1991 yılında *Sun Microsystems' te* görevli James Gosling ve Patrick Naughton tarafından Gren kod adıyla, TV ve mikrodalga fırın gibi elektronik ev eşyalarını kontrol etmekte ve bilgi alışverişinde kullanılacak bir programlama dili olarak tasarlandı [20-21].

Internet'te projesinin geleceği için potansiyel gören Sun, önce programlama dilini Oak olarak yeniden isimlendirdi. Daha sonra Hot Java isimli Web Browser Oak kullanılarak geliştirildi. Daha sonra Oak teknolojisi Java olarak yeniden

isimlendirildi ve Java' nın, birleşik sistem çatıları oluşturmada en çok tercih edilen teknoloji ve program dili olduğu bugünlere gelindi [20,21].

3.2.2. Java' nın uygulama alanları

- Uygulamalar
- Komut satırı ile çalışabilen programlar
- Grafiksel kullanıcı arabirimleri (GUI)
- Appletler
- Servletler
- Gömülü sistem uygulamaları
- Java Server Sayfaları (JSP)

3.2.3. Java'nın özellikleri ve avantajları

- Basittir,
- Dinamiktir
- Nesneye yöneliktir
- Taşınabilir ve farklı platformlarda
- Java programları uluslar arası özellikleri taşıyabilir.
- Güvenlidir
- Çok görevlidir ve çok kullanımlıdır [22]

3.2.4. Java versiyonları

Java'nın versiyonlarını Java 1.0, Java 1.2, Java 1.3, Java 1.4 olarak sayabiliriz. Java versiyonlarındaki en dikkat çekici değişiklik Java 1.2 ile olmuş ve sun firması bu versiyondan sonraki Java teknoloji platformunu Java 2 olarak isimlendirmiştir.

3.2.5. Java programlamak için neler gerekli

1. Java Software Development Kit (JSDK)

2. Bir metin editörü
3. (Appletler için opsiyonel olarak) bir web tarayıcı

3.2.6. Programın derlenmesi

Derleme işlemini JSDK'nın derleyici programı olan javac.exe yapar. Programı derlemek için komut satırında dosyanın bulunduğu klasörde
C:\uygulamalar\javac Merhaba.java yazıp enter tuşuna basılır.

Hata yoksa uygulamalar klasöründe ikinci bir dosya oluşur. Bu Merhaba.java isimli kaynak dosyanın derlendikten sonra byte-code' a dönüşmüş hali olan Merhaba.class dosyasıdır. Dosyayı çalıştırmak için:

C:\uygulamalar\java Merhaba yazıp enter tuşuna basılır [22].

3.2.7. Java sanal makinesi

Java ile yazılan kaynak kodlar sanal bir işlemcinin anlayacağı makine koduna çevrilir, fakat bu kod bildiğimiz gerçek makine kodu olmadığından bu koda byte-code denir. Yani yazılan kaynak kod derlenerek byte-code' dönüştürülür. Bu byte-code'lar programın çalışması esnasında Java Sanal Makinesi tarafından gerçek makine koduna dönüştürülür.

Java dili ile yazılan kaynak kodlar .java uzantılı dosyalarda saklanır. javac derleyicisi bu kaynak kodları .class uzantılı dosyalarda saklanan byte-code'a çevirir. Bu byte-code'lar java yorumlayıcısı tarafından çalıştırılır. JRE(Java Runtime Engine) ve SDK(Software Development Kit) java kodlarını çalıştıran sistemdir. Bir makinede bir Java programı çalıştırılabilmesi için JRE yüklenmelidir. Fakat JRE bir derleyici barındırmaz sadece derlenmiş programları çalıştırır. Dolayısıyla Java kodlarının derlenebilmesi için sisteme SDK yüklenmelidir. SDK içerisinde JRE'i de barındırır yani SDK yüklendikten sonra tekrar bir JRE'ye ihtiyaç yoktur [23].

3.2.8. Java programlarının yapısı

imported classes

```
public class sınıfadı {
    public static void main (string args [ ] ) {
        değişkenlerin tanımlanması
        çalıştırılabilir ifadeler
    }
    varsa diğer metotlar
}
```

3.2.9. Sınıfların(class) yapısı

```
public class Programımızınİsmi {
    public static void main (string [ ] args){
    }
}
```

3.2.10. Metotların yapısı

```
AnahtarKelime dönüşüpti metodunİsmi(parametreler) {
}
```

3.2.11. Tanımlayıcılar

Tanımlayıcılar sınıf, metot ve değişken adları için kullanılır. Bir tanımlayıcı büyük veya küçük harf, rakam, alt çizgi ve dolar karakteri içerebilir. Sayısal bir ifadeyle karıştırılmaması için tanımlayıcılar bir rakam ile başlayamaz. Java harf kipi duyarlılığı olan bir dildir [24].

3.2.12. Açıklamalar

Java kaynak kodunun içerisine kod değeri olmayan açıklama yazılabilmesi için

aşağıdaki yolları sunar.

- /* yorum */ uzun açıklamalar için
- // yorum tek satırlık açıklamalar için

3.3. Veri Tipleri, Değişkenler ve Diziler

Java'daki veri tipleri, değişkenler ve diziler aşağıda anlatılmıştır.

3.3.1. Veri tipleri

Çizelge 3.1. Veri tipleri

<i>Adı</i>	<i>Türü</i>	<i>Genişlik</i>	<i>Aralık</i>
Boolean	Mantıksal		True/false
Char	Metinsel	16-bit	
Byte	Tamsayı	8-bit	$-2^7 \dots 2^7 - 1$ arası
Short	Tamsayı	16-bit	$-2^{15} \dots 2^{15} - 1$ arası
Int	Tamsayı	32-bit	$-2^{31} \dots 2^{31} - 1$ arası
Long	Tamsayı	64-bit	$-2^{63} \dots 2^{63} - 1$ arası
Double	Kayan noktalı	64-bit	$1.798 \times 10^{308} \dots 4.9 \times 10^{-324}$ arası
Float	Kayan noktalı	32-bit	$3.403 \times 10^{38} \dots 1.401 \times 10^{-45} - 1$ arası

3.3.2. Değişkenler

Değişkenler, belli bir türdeki değeri belli bir isim altında saklamaya yarayan program elementidir. Değişken, bir Java programı için temel depolama birimidir. Bir değişken bir tip, bir tanımlayıcı ve isteğe bağlı bir ilk değer atayıcı belirtilerek tanımlanır. Java'da bütün değişkenler kullanılmadan önce tanımlanmak zorundadır [25].

Tip tanımlayıcı[=değer][,tanımlayıcı[=değer]...]

3.1.Örnek : int d=3,e,f=5;

3.3.3. Diziler

Diziler(array) nesnedir; içerisinde belirli sayıda eleman bulunur. Dizinin içerisindeki elemanlara eksi olmayan bir tam sayı ile ifade edilen dizi erişim indisi ile erişilir. Dizinin ilk elemanının indisi 0'dır. Dizi içerisindeki elemanlar aynı türden olmak zorundadır. Her tipte dizi oluşturulabilir ve diziler bir veya daha fazla boyutlu olabilir. Dizi tanımlamak için, tip dizi_değişkenadi[]; ifadesi kullanılır [26].

3.2.Örnek : int ay_gunleri[];

Fiziksel bir değer atamak için, new anahtar sözcüğü kullanılır.

3.3.Örnek : ay_gunleri=new int[12];

Dizi elemanlarına erişim için: dizi_değişkeni[indeks];

3.4.Örnek : ay_gunleri[1]=28;

3.3.4. Karakter katarları

Java'nın, string adındaki karakter katarı tipi basit bir tip değildir. String, bir nesne tanımlar. String tipi, karakter katarı değişkenler tanımlamak için kullanılır. Ayrıca karakter katarı dizileri tanımlamak için de kullanılır. Bir tırnak içerisindeki karakter katarı string tipinde değişkene atanabilir [27].

3.5.Örnek : String str= "Merhaba";

3.4. Operatörler

Aritmetik, bir artırma bir azaltma, ilişkisel, mantıksal, atama, ?, string operatörü vardır.

3.4.1. Aritmetik operatörler

Çizelge 3. 1. Aritmetik operatörler

<i>Oparatör</i>	<i>Kullanılış Şekli</i>	<i>Açıklama</i>
+	Değişken1+Değişken2	Değişken1 ile Değişken2'yi toplar
-	Değişken1- Değişken2	Değişken1 ile Değişken2'yi çıkarır
*	Değişken1*Değişken2	Değişken1 ile Değişken2'yi çarpar
/	Değişken1/ Değişken2	Değişken1, Değişken2 tarafından bölünür
%	Değişken1%Değişken2	Değişken1'in Değişken2'ye bölümünden kalan

3.4.2. Bir artırma bir azaltma operatörleri

Çizelge 3. 2. Bir artırma bir azaltma operatörleri

<i>Oparatör</i>	<i>Kullanılış Şekli</i>	<i>Açıklama</i>
++	Değişken++	Önce değişkenin değerini hesaplar sonra değişkenin değerini bir artırır.
++	++Değişken	Önce değişkenin değerini artırır sonra değişkenin değerini hesaplar.
--	Değişken--	Önce değişkenin değerini hesaplar sonra değişkenin değerini bir azaltır.
--	-- Değişken	Önce değişkenin değerini azaltır sonra değişkenin değerini hesaplar.

3.4.3. İlişkisel operatörler

Çizelge 3. 3. İlişkisel operatörler

<i>Oparatör</i>	<i>Kullanılış Şekli</i>	<i>İşlevi/Anlamı</i>
>	Değişken1>Değişken2	Değişken1, Değişken2' den büyükse
>=	Değişken1>=Değişken2	Değişken1, Değişken2' den büyükse veya eşitse
<	Değişken1 < Değişken2	Değişken1, Değişken2' den küçükse
<=	Değişken1<=Değişken2	Değişken1, Değişken2' den küçükse veya eşitse
=	Değişken1=Değişken2	Değişken1, Değişken2' ye eşitse
!=	Değişken1 !=Değişken2	Değişken1, Değişken2' ye eşit değilse

3.4.4. Mantıksal operatörler

Mantıksal operatörler birden çok karşılaştırma işlemini birleştirip tek bir koşul ifadesi haline getirilmesi için kullanılır [28].

Çizelge 3. 4. Mantıksal operatörler

<i>Oparatör</i>	<i>Kullanılış Şekli</i>	<i>İşlevi/Anlamı</i>
& &	Değişken1 && Değişken2	VE operatörü
	Değişken1 Değişken2	VEYA operatörü
^	Değişken1 ^ Değişken2	YA DA operatörü
!	! Değişken	DEĞİL' ini alma operatörü

3.4.5. Atama operatörleri

Atama işlemi, bir değeri veya değişkeninin içeriğini bir başka değişkene yerleştirmektir [28].

değişken=deyim;

Çizelge 3. 5. Atama operatörleri

<i>Oparatör</i>	<i>Kullanılış Şekli</i>	<i>Eşittir</i>
+ =	Değişken1 + = Değişken2	Değişken1=Değişken1+Değişken2
- =	Değişken1 - = Değişken2	Değişken1=Değişken1-Değişken2
* =	Değişken1 * = Değişken2	Değişken1=Değişken1*Değişken2
/ =	Değişken1 / = Değişken2	Değişken1=Değişken1/Değişken2
% =	Değişken1 % = Değişken2	Değişken1=Değişken1%Değişken2
& =	Değişken1 & = Değişken2	Değişken1=Değişken1&Değişken2
=	Değişken1 = Değişken2	Değişken1=Değişken1 Değişken2
^ =	Değişken1 ^ = Değişken2	Değişken1=Değişken1^Değişken2

3.4.6. Soru işareti operatörü

Burada deyim1, boolean olarak değerlendirilebilen herhangi bir deyim olabilir. Eğer deyim1 true ise, deyim2; değilse deyim3 değerlendirilir.

deyim1? Deyim2:deyim3

3.6.Örnek: Ratio=denom= =0?0:num/denom;

3.4.7. String + Operatörü

+ operatörü String verilerde birleştirme görevi görür; eğer ifade String ile başlarsa, onu izleyen veri tipleri de kendiliğinden String'e dönüştürülür.

3.5. Kontrol İfadeleri

Kontrol ifadeleri seçim, döngü ve sıçrama ifadeleri olarak üçe ayrılır.

3.5.1. Java'nın seçim ifadeleri

Java'nın seçim ifadeleri if, switch olmak üzere iki tanedir.

If

If, Java'nın koşullu dallanma ifadesidir. Programın çalışmasını iki farklı yol arasında yönlendirmek için kullanılır [28].

```
if (koşul) { Durum true olduğunda çalışması istenen kod bloğu
    } else
    Durum false olduğunda çalışması istenen kod bloğu
    }
```

Switch

Switch ifadesi Java'nın çok yollu dallanma ifadesidir. Programın çalışmasını bir deyimın değerine bağlı olarak farklı kısımlara dallandırmanın kolay bir yoludur [28].

```
switch (deyim) { case değer1: //Çalışması istenen kod bloğu; break;
                 case değer2: //Çalışması istenen kod bloğu; break;
                 case değerN: //Çalışması istenen kod bloğu; break;
                 default:     //Çalışması istenen kod bloğu;
                 }
```

3.5.2. Java' nın döngü ifadeleri

Java'nın döngü ifadeleri while, do-while ve for olmak üzere üç tanedir.

While

```
while (koşul) {   çalışması istenen kod bloğu;
                 }
```

Koşul herhangi bir boolean deyim olabilir. Döngü gövdesi, koşul deyimini true olduğu sürece çalıştırılır. Koşul false olduğunda kontrol, döngüden sonraki ilk koda geçer .

Do-while

```
do   { Çalışması istenen kod bloğu
      } while (koşul);
```

Do-while döngüsü her tekrarlandığında, gövde bir kez çalıştırılır, sonra koşul deyimini kontrol edilir. Eğer ifade true ise döngü tekrarlanır, aksi halde biter. Java'nın bütün döngülerinde olduğu gibi bir boolean deyim olmak zorundadır [28].

For

```
for (başlangıç;koşul;artış){çalışması istenen kod bloğu;
    }
```

Döngü ilk başladığında, döngünün başlangıç kısmı çalıştırılır. Sonra koşul değerlendirilir. Koşul doğruysa döngünün gövdesi çalıştırılır. Koşul yanlışsa döngü sonlandırılır. Daha sonra döngünün artış kısmı çalıştırılır. Buradaki deyim döngü kontrol değişkenini eksiltir veya artırır. Bu işlem kontrol deyimini false olana kadar tekrarlanır [28].

3.5.3. Sıçrama ifadeleri

Java'nın sıçrama ifadeleri break, continue ve return ifadeleridir.

break

Break kullanılarak, koşullu deyimden ve gövdenin geri kalanındaki kodun atlanması sağlanabilir. Döngü içinde bir break ile karşılaşıldığında döngü bitirilir ve kontrol döngüden sonra gelen ifadeye geçer.

continue

continue ifadesi, while ve do-while döngülerinde, kontrolü doğrudan döngüyü kontrol eden koşullu deyimde aktarır. For döngüsünde ise kontrol, önce iterasyon, sonra da koşullu deyim kısmına geçer. Üç döngü içinde aradaki kod atlanır.

return

Return ifadesi, bir metottan açıkça bir geri dönüş elde etmek için kullanılır. Yani, program kontrolünün tekrar metod çağrısına aktarılmasını sağlar. Dolayısıyla return ifadesi içinde çalıştığı metodu hemen sonlandırır.

3.6. Sınıfların Tanıtımı

Sınıf, bir nesnenin şeklini ve doğasını tanımlar. Sınıf yeni bir veri tipi tanımlamaktadır. Bir kez tanımlanan bu yeni tip, bu tipte yeni nesnelere oluşturmak için kullanılabilir. Bu yüzden, sınıf nesne için bir şablondur ve bir nesne sınıfın örneğidir.

Sınıfın her bir örneği(nesnesi), bu değişkenlerin kendine özgü bir kopyasını taşır. Bu sebeple bir nesnenin verisi diğer nesnelerin verisinden farklıdır [29].

3.6.1. Sınıfın genel biçimi

```
class sinif_adi {
    tip ornek_degisken1;
    tip ornek_degisken2;
    tip ornek_degiskenN;
    tip metot_adi1 (parametre_listesi){ //metodun gövdesi
    }
    tip metot_adiN (parametre_listesi){ //metodun gövdesi
    }
}
```

Bir sınıf, class anahtar sözcüğü ile tanımlanabilir. Sınıf içinde tanımlanan veri veya değişkenler örnek değişkenler olarak adlandırılır. Kod metodların içinde yer alır. Java sınıfları, bir main() metoduna sahip olmak zorunda değildir.

3.6.2. Basit bir sınıf

```
3.7.Örnek : class Kutu {
    double genislik;
    double yukseklik;
    double derinlik;
}
```

Çok basit sınıflar yalnızca veri ya da kodu içerirken, pek çok gerçek sınıf her ikisini de içerir. Bir sınıfın kodu, o sınıfın verisine ulaşmak için bir arabirim tanımlar. Nesne oluşturmak için new anahtar sözcüğü kullanılır.

3.8.Örnek : Kutu benimkutum=new Kutu();

Bir sınıfın örneğini oluşturduğunuz her seferde, sınıf tarafından tanımlanan her örnek değişkenin, kendine özgü bir kopyasını içeren bir nesne oluşturursunuz. Bu değişkenlere erişmek için nokta(.) operatörü kullanılır. Genel olarak nokta operatörü nesne içindeki örnek değişkenlere ve metodlara erişmek için kullanılır.

3.9.Örnek: benimkutum.derinlik=100;

Bir nesnenin örnek değişkenlerine yapılan değişiklik diğer örnek değişkenler üzerinde geçerli olmaz [29].

3.6.3. Nesnelerin bildirimi

Bir sınıf oluşturduğumuzda yeni bir veri tipi de oluşturmuş olursunuz. Daha sonra bu tip o tipte nesnelere tanımlamak için kullanılır. Bir sınıfın nesnelere elde etmek, iki adımlı bir işlemdir.

Önce sınıf tipinde bir değişken tanımlanır. Bu değişken bir nesne tanımlamaz. Sadece, bir nesneye referansta bulunabilen bir değişkendir. İkinci olarak, nesnenin fiziksel ve gerçek bir kopyası oluşturulur. Daha sonra bu kopya o değişkene atanır. Bu new operatörünü kullanarak yapılır [29].

New operatörü, bellekte bir nesne için dinamik olarak yer ayırır ve ona bir referans döndürür. Bu referans aşağı yukarı, nesnenin new tarafından ayrılan bellekteki adresidir. Bu referans sonra değişkende depolanır. Bu sebeple, java' da tüm sınıf nesnelere için dinamik olarak bellekte yer ayrılmalıdır [28].

```
3.10.Örnek: Kutu benimkutum;           // nesneye referans tanımlanıyor.
           Benimkutum = new Kutu ();    // bir Kutu nesnesi ayrılıyor.
```

3.6.4. Metotlara giriş

```
tip metot_adi(parametre_listesi) {           // metodun gövdesi           }
```

Burada, tip metot tarafından döndürülen veri tipini tanımlar. Eğer metot bir değer döndürmeyecekse, tipi void olmak zorundadır. Metot_adi geçerli kapsam içinde başka öğeler tarafından kullanılmamalıdır. Parametreler, çağrıldıklarında metoda aktarılan argümanların değerini tutan değişkenlerdir. Eğer metodun parametresi yoksa parametre listesi boş olur.

Değer döndüren metotlar

Void' in dışında başka bir dönüş tipine sahip metotlar, return ifadesinin örnekteki gibi kullanımıyla çağrıldıkları rutinlere değer döndürürler.

```
3.11.Örnek : class Kutu{   double genislik;
                        double yukseklik;
                        double alan() { return genislik*yuksekkik; }
                        }
```

```
class KutuHesap {public static void main(String args[ ]) {
    Kutu benimkutum= new Kutu ();
    double alan;
    benimkutum.genislik=10;
    benimkutum.yuksekkik=10;
    alan= benimkutum.alan( );
    System.out.println("Alan="+alan);    }
```

Parametre Alan Metotlar

Parametreler, bir metodun geliştirilmesini sağlar. Yani parametreleri olan bir

metot, değişik veriler üzerinde işleyebilir veya bazı farklı durumlarda kullanılabilir.

```

3.12.Örnek :class Kutu {
    double Kutu{
        double genislik;
        double yukseklik;
        double derinlik;
        double alan() { return genislik*yukseklik; }
        void setDim( double g, double y) {
            genislik=g;
            yukseklik=y;
        }
    }
}

class KutuHesap {public static void main(String args[ ]) {
    Kutu benimkutum= new Kutu ( );
    double al;
    benimkutum.setDim(10,20,15);
    al=benimkutum.alan();
    System.out.println("Alan="+al);
}
}

```

3.6.5. Yapılandırıcılar

Java, nesnelere oluşturulduklarında kendilerini ilk kullanıma hazırlama imkânı sağlar. Bu otomatik ilk kullanıma hazırlama işlemi bir yapılandırıcı kullanılarak yapılır.

Bir sınıf için açıkça bir yapılandırıcı tanımlamazsanız, Java varsayılan bir yapılandırıcı oluşturur. Varsayılan yapılandırıcı, tüm örnek değişkenlerine otomatik olarak sıfır ilk değeri atar. Kendi yapılandırıcınızı tanımladığınızda ise varsayılan yapılandırıcı devreden çıkar [28,29].

```

3.13.Örnek : class Kutu{
    double genislik;
    double yukseklik;
    Kutu( ) { System.out.println("Kutu değerleri");
}
}

```

```

        genislik=0;
        yukseklik=10;    }
        double alan() {    return genislik*yuksekklik;}
    }

```

3.6.6. Artık toplama

Bir nesneye herhangi bir referans yoksa, artık o nesneye ihtiyaç olmadığı kabul edilir ve nesnenin bellekte kapladığı yer serbest bırakılır.

3.6.7. Finalize() metodu

Finalize() metodu içinde, bir nesnenin yok edilmesinden önce yapılması gereken eylemler belirtilir. Artık toplayıcı periyodik olarak çalışır ve nesnelere kontrol eder. Bir nesne serbest bırakılmadan önce, Java çalışma zamanı nesne üzerinde finalize() metodunu çağırır.

3.7. Metot ve Sınıflar

Java'nın metod ve sınıf ları aşağıda ayrıntılarıyla anlatılmıştır.

3.7.1. Metotları aşırı yüklemek

Java' da, parametre tanımlamaları farklı olmak şartıyla, aynı sınıf içinde aynı adı paylaşan iki veya daha fazla metod tanımlanabilir. Bu durumda metoda aşırı yüklenmiş (overloaded) metod denir. Metodun aşırı yüklenmesi Java'nın çok biçimliliği uygulama yollarından biridir.

Aşırı yüklenmiş bir metod çağrıldığında Java, gerçekte hangi aşırı yüklenmiş metodun çağrıldığına karar vermek için, argümanların tipini ve sayısını kullanır. Bu yüzden aşırı yüklenmiş metotların parametrelerinin tipi ve sayısı farklı olmak zorundadır [28].

```

3.14.Örnek : class OverloadedDemo { void test() {
    System.out.println("Parametresiz...");    }
    //test metodunu iki tamsayi parametre icin asiri yuklemek
    void test (int a, int b) {
    System.out.println("a ve b:"+a+" "+b);    }
    //test metodunun bir double parametre icin asiri yuklemek
    void test (double a) {
    System.out.println("double a:"+a);        }
    }
public static void main (String args[ ]) {
    OverloadedDemo ob=new OverloadedDemo ( );
    double sonuc;
    ob.test( );
    ob.test(10,20);
    sonuc=ob.test(123.25);
    System.out.println("ob.test(123.25) sonuc:"+sonuc);
    }

```

3.7.2. Nesneleri parametre olarak kullanmak

Metotlara parametre olarak basit tiplerin yanında nesne de aktarılabilir.

```

3.15.Örnek :class Test { int a, b;
    Test (int i, int j) {    a=i;  b=j;    }
    boolean equals (Test o) {    if (o.a==a && o.b== b) return true;
                                else return false;    }
    }
class PassOb { public static void main(String args[ ]) {
    Test.ob1= new Test(100,22);
    Test.ob2= new Test(100,22);
    System.out.println("ob1== ob2:"+ob1.equals(ob2));    }
}

```

3.7.3. Yineleme

Yineleme, bir şeyi kendisi cinsinden tanımlama işlemidir. Java' daki haliyle bir metodun kendi kendini çağırabilme özelliğidir.

```
3.16.Örnek : class Faktoriyel { int fakt (int n) { int sonuc;
                if (n==1) 1;
                sonuc=fakt(n-1)*n;
                return sonuc; } }
```

3.7.4. Erişim kontrolüne giriş

Bir üyeye nasıl erişileceği, erişim belirteçleri tarafından belirlenir. Java'nın erişim belirteçleri: public, private ve protected' dir. Protected, sadece kalıtım gerekli olduğunda geçerlidir. Bir sınıfın üyesi public belirteci ile belirtildiğinde, programın diğer tüm kodu tarafından bu üyeye erişilebilir. Bir üyenin sınıfı private olarak belirtildiğinde, ona sadece kendi sınıfının diğer üyeleri erişebilir [28].

```
3.17.Örnek : public int i;
                private double j;
                private int myMethod (int a, char b){ }
```

3.7.5. Static anahtar sözcüğü

Static olarak tanımlanan bir üyeye, sınıfından hiçbir nesne oluşturulmadan önce ve hiçbir nesneye referans olmadan erişilebilir. Hem metotları, hem de değişkenleri static olarak tanımlanabilir. Static bir üyenin en yaygın örneği main() metodudur.

3.7.6. Final anahtar sözcüğü

Bir değişken final olarak tanımlanması, bu değişkenin içeriğini düzenlemelere karşı engeller. Bir final değişkeni tanımlandığında bir ilk atamasının yapılması zorunludur.

3.18.Örnek : final int FILE_NEW=1;

3.7.7. İç içe yerleşmiş ve iç sınıflar

Bir sınıf içinde başka sınıflar tanımlayarak iç içe yerleşmiş sınıflar oluşturmak mümkündür. İç içe yerleşmiş bir sınıfın kapsamı, onu kuşatan sınıfın kapsamı ile sınırlıdır.

3.7.8. String sınıfına giriş

Oluşturduğumuz her bir karakter katarı aslında string tipinde bir nesnedir. Karakter katarı sabitleri bile aslında birer string nesnedir. String tipi nesnelere değişmezler. String nesnesi bir kez oluşturulduğunda, onun içeriği bir daha düzenlenemez.

String nesnelere birleştirmek için + operatörü kullanılır. İki karakter katarının eşit olup olmadığını equals() metodunu kullanarak test edebilirsiniz. Bir karakter katarının uzunluğunu, length() metodunu çağırarak öğrenebilirsiniz. Bir dizinin içindeki belirli indeksdeki karakteri charAt() metodunu çağırarak elde edebilirsiniz.

3.8. Kalıtım

Kalıtım, bir nesnenin başka bir nesnenin özelliklerini devralmasıdır.

3.8.1. Kalıtımın temelleri

Kalıtım kullanılarak ilişkili öğelerin ortak özelliklerini tanımlayan genel bir sınıf oluşturulabilir. Daha sonra, belirli sınıfların her biri kendine has özellikleri de ekleyerek bu sınıfı kalıtım yoluyla devralabilir.

Bir sınıfı türetmek için, basitçe o sınıfın tanımını bir başka sınıfına, extends anahtar sözcüğünü kullanılır. Oluşturulan herhangi bir sınıf için, yalnızca bir üst sınıf belirtilebilir.

```
class alt_sinif_adi extends ust_sinif_adi { //sinif gövdesi
    }
}
```

3.8.2. Üye erişimi ve kalıtım

Bir alt sınıf üst sınıfını tüm üyelerini içermesine rağmen, üst sınıfın private olarak tanımlanan üyelerine erişemez.

3.8.3. Super kullanımı

Bir alt sınıfın, üst sınıfa göndermede bulunması gerekirse bu, super anahtar sözcüğü kullanılarak yapılır. Super' in iki genel formu vardır. Birincisi, üst sınıf yapılandırıcısını çağırır. İkincisi, bir üst sınıfın, bir alt sınıfın üyesi tarafından gizlenen bir üyesine erişmek için kullanılır.

3.8.4. Yapılandırıcıların çağırılması

Bir sınıf hiyerarşisinde yapılandırıcılar türetilme sırasına göre, yani üst sınıftan alt sınıfa doğru çağrılır. Üstelik super(), bir alt sınıf yapılandırıcısının içinde çalıştırılması zorunlu ilk ifade olduğundan, super() kullanılsın ya da kullanılmıyorsa bu sıra aynıdır. Eğer super() kullanılmamışsa, her alt sınıfın varsayılan veya parametresiz yapılandırıcısı çalıştırılır.

3.8.5. Metodu geçersiz kılmak (Overriding)

Bir sınıf hiyerarşisinde, alt sınıfa ait bir metot üst sınıftaki bir metotla aynı ada ve tipe sahipse, alt sınıftaki metot, üst sınıftaki metodu geçersiz kılmış olur.

Eğer geçersiz kılınan fonksiyonun üst sınıf versiyonuna erişmek istiyorsanız, bunu super kullanarak yapabilirsiniz.

3.8.6. Dinamik metot göndermesi

Dinamik metot göndermesi, geçersiz kılınan bir metoda yapılan çağrının derleme zamanında değil, çalışma zamanında çözümlenmesi demektir. Java'nın çalışma zamanı çok biçimliliğini nasıl uyguladığını gösterir.

Geçersiz kılınan bir metot, bir üst sınıf referansı aracılığıyla çağrıldığında, java metodun hangi versiyonunun çalıştırılacağına, çağrının yapıldığı anda göndermede bulunan nesnenin tipine göre karar verir. Bu karar çalışma sırasında verilir. Hangi metodun çalıştırılacağına, göndermede bulunan nesnenin tipini belirler [28].

3.8.7. Metotları geçersiz kılmanın nedeni

Çok biçimlilik, türevlerinin tümü ortak olan metotlar tanımlamak için genel bir sınıf sağlar. Böylece alt sınıfların, bu metotların hepsinin ya da bazılarının kendilerine özgü uygulamalarını tanımlamasına verilir.

3.8.8. Özet sınıfları kullanmak

Bazen yalnızca tüm alt sınıfları tarafından paylaşılacak ve içeriği onlar tarafından doldurulacak, genel bir form tanımlayan üst sınıflar isteyeceksiniz. Böyle bir sınıf, alt sınıfların gerçekleştirmek zorunda olduğu metotların doğasını belirler.

Bazı metotların alt sınıflar tarafından abstract tip belirteci kullanılarak geçersiz kılınması gerekebilir. Çünkü üst sınıfta tanımlanmış bir uygulama yoktur. Bu yüzden, bir alt sınıf bunları geçersiz kılmalıdır; üst sınıfta tanımlanmış versiyonunu kullanamaz. Özet bir metot tanımlamak için genel form:

abstract tip adi (parametre_listesi);

Görüldüğü gibi, bir metot gövdesi yoktur. Bir yada daha fazla özet metot içeren her sınıf özet olarak tanımlanmak zorundadır. Bir sınıfı özet olarak tanımlamak için, sınıf

tanımlamasındaki class anahtar sözcüğünün önünde abstract anahtar sözcüğünü kullanmak yeterlidir. Özet tipte bir nesne olamaz. Özet sınıflarda somut metotlara da izin verilir [28].

```

3.19.Örnek : abstract class Calisan { public String pozisyon= "Calisan";
                public abstract void calis(); // soyut yordam
                public void zamIste() { // soyut olmayan yordam
                    System.out.println("Calisan zamIste"); }
            }
class Mudur extends Calisan {
                public Mudur () { //yapılandırıcı
                    pozisyon= "Mudur"; }
                public void calis() { // iptal etme
                    System.out.println("Mudur Calisiyor"); }
                public void zamIste() { //iptal etme
                    System.out.println("Mudur zam istiyor"); }
            }
public class AbIsYeri {
                public static void mesaiBasla(Calisan[] c) {
                    for (int i=0; i<c.length; i+ +) { c[i].calis(); }
                }
                public static void main(String args[] ) {
                    Calisan[] c= new Calisan[3];
                    //c[0]=new Calisan(); //soyut sınıflar new ile doğrudan oluşturulamazlar
                    c[0]=new Mudur();
                    mesaiBasla( c); }
            }

```

3.8.9. Final' in kalıtımla kullanılması

Bir metodun geçersiz kılınma işlemini kabul etmemek için, tanımlamanın başına final belirteci kullanılır.

```
3.20.Örnek : class A{ final void meth( ) {
                System.out.println("this is a final method"); }
                }
```

Bazen bir sınıfın türetilmesini önlemek için, sınıf tanımlaması final ile yapılır. Bir sınıfı final olarak tanımlamak, tüm metotlarının da final olarak tanımlanmasına sebep olur. Bir sınıf hem final, hem de abstract olarak tanımlanamaz.

```
final class A{ }
```

3.8.10. Object sınıfı

Java’ da bütün sınıflar object sınıfının alt sınıfıdır. Bu, Object tipinde bir referans değişkeninin diğer herhangi bir sınıfın nesnesine göndermede bulunacağı anlamına gelir.

3.9. Paketler ve Arabirimler

Paketler, sınıf ad uzayını bölmelere ayrılmış olarak korumak için kullanılan, sınıfların tutulduğu taşıyıcılardır. Bir paket sayesinde, bir sınıf oluşturulabilir ve başka bir yerde depolanmış aynı adı taşıyan başka sınıflarda çatışma endişesi olmadan bu sınıfı kendi paketinizde depolayabilirsiniz.

Interface anahtar sözcüğünün kullanımı ile Java size tam bir özet arabirim oluşturmanızı sağlar. Interface kullanarak, bir yada daha fazla sınıf tarafından kullanılacak bir metotlar kümesi tanımlayabilirsiniz. Interface’ in kendisi, gerçek bir uygulama tanımlamaz. Bir sınıf birden fazla arabirim uygulayabilir. Buna karşılık, bir sınıf sadece tek bir üst sınıftan türetilir [30].

3.9.1. Paketler

Ad çakışmalarını önlemek için, her sınıf için benzersiz adlar seçilir. Bir süre sonra,

ad uzayını yönetecek başka yol olmaksızın, her ayrı sınıf için uygun ve açıklayıcı adlar bulmakta zorlanabilirsiniz. Bir sınıf için seçtiğiniz adın kabul edilebilir derecede benzersiz olmasını ve başka programcılar tarafından seçilen sınıf adlarıyla çakışmamasını sağlayacak başka bir yola ihtiyacınız vardır. Java' da sınıf ad uzayını daha yönetebilir bölmelere ayırma mekanizmasına paket adı verilir. Bir paket içinde, bu paketin dışındaki kodun erişemeyeceği sınıflar oluşturabilirsiniz. Ayrıca, yalnızca aynı paketin diğer üyelerine açık sınıf üyeleri oluşturabilirsiniz [30].

3.9.2. Paketleri tanımlamak

Paket oluşturmak için Java kaynak dosyasına ilk ifade olarak package komutunu dahil etmek yeterlidir. Bu dosyada tanımlanan her sınıf belirtilen pakete ait olacaktır. Eğer package ifadesini ihmal ederseniz, sınıf adları, adı olmayan varsayılan bir pakete konur.

3.21.Örnek : `package Benimpaketim;`

Java, paketleri depolamak için dosya sistemi dizinlerini kullanır. Örneğin Benimpaketim'in parçası olarak tanımlanan her .class dosyası, Benimpaketim adlı bir dizinde depolanmak zorundadır.

Package ifadesi basitçe bir dosya içinde tanımlanan sınıfların hangi pakete ait olduğunu belirtir.

Bir paket hiyerarşisi oluşturabilirsiniz. Bunu yapmak için, her bir paket adını üsttekinden bir nokta kullanarak ayırmanız yeterlidir [30].

3.22.Örnek : `package java.awt.image;`

3.9.3. Paketleri bulmak ve classpath

Varsayılan durumda Java çalışma zamanı sistemi başlangıç noktası olarak, geçerli

Tüm standart Java sınıfları, java adında bir pakette depolanır. Normalde, kullanılan her paket veya sınıfın aktarılması gerekir [30].

3.9.6. Arabirimleri tanımlama

Interface anahtar sözcüğünü kullanılarak, bir sınıfın arabirimini onun uygulamasından tamamen özetlenebilir. Arabirimlerin örnek değişkenleri yoktur ve metotları gövdesiz olarak tanımlanmıştır. Arabirim bir kez tanımlandığında, çok sayıda sınıf onu kullanabilir. Ayrıca bir sınıf da birçok arabirime sahip olabilir.

Bir arabirim oluşturmak için, sınıfın arabirim tarafından tanımlanan metotların tümünü oluşturması gerekir. Ancak her sınıf, kendi uygulamasının detaylarının kararını kendisi verir. Interface anahtar sözcüğü sayesinde Java, çok biçimliliğin “bir arabirim, çok metot” özelliğinden tam olarak yararlanılmasını sağlar [30].

```
erişim interface ad {
    Donus_tipi metot_adi1(parametre_listesi);
    Tip son_değişkenadi1=deger;
    donustipi metotN(parametre_listesi);
    tip son_değişkenadiN=değer;
}
```

Burada erişim ya public olur ya da kullanılmaz. Tanımlanan metotların gövdesi yoktur. Değişkenler, arabirim tanımlanmasında tanımlanabilir. Bunlar kendiliğinden final ve static' tir. Yani uygulanan sınıf tarafından değiştirilmezler. Ayrıca, başlangıç değerleri sabit bir değer olmak zorundadır.

3.9.7. Arabirimlerin uygulanması

Arabirim bir kez tanımlandığında, bir ya da daha fazla sınıf onu uygulayabilir. Bir arabirimi uygulamak için, sınıf tanımında implements cümleciğini ekledikten sonra arabirim tarafından tanımlanan metotları oluşturmak yeterlidir.

```
erişim class sinifadi [extends ust_sinif]
```



```
[implements arabirim[,arabirim..]] {
    // sınıf gövdesi }
```

Burada erişim ya public'tir ya da hiç kullanılmaz. Arabirimi uygulayan metotlar, public olarak tanımlanmak zorundadır. Ayrıca, arabirimi uygulayan metodun tip imzası ile tam olarak eşleşmek zorundadır.

3.9.8. Kısmi uygulamalar

Bir sınıf bir arabirim içeriyor, ancak bu arabirim tarafından metotları tam olarak uygulamıyorsa, bu sınıfın aşağıda gösterildiği abstract olarak tanımlanması gerekir.

3.9.9. Arabirimlerde değişkenler

Paylaşılan sabitleri basit bir arabirim tanımlanması ile birden çok sınıfa aktarmak için arabirimler kullanılır. Bu arabirim tanımlanması istenilen değerler için ilk değerlerin atandığı değişkenleri içerir. Bu arabirimi bir sınıfa dahil ettiğinizde tüm değişken adları, kapsam içinde birer sabit olarak bulunur.

```
3.24.Örnek : abstract class Incomplete implements Callback {
    int a,b;
    void show() { System.out.println(a+" "+b); }
}
```

3.9.10. Arabirimler genişletilmesi

Bir arabirim, extends anahtar sözcüğünü kullanarak diğer bir arabirimi kalıtım yoluyla devralabilir. Bir sınıf, başka bir arabirimden türetilen bir arabirimi kullandığında, arabirimin kalıtım zincirinde tanımlanan tüm metotları için uygulamalar sağlamak zorundadır [28].

```
3.25.Örnek : interface A;
```

```
interface B extends A;
```

3.10. İstisna Yönetimi

Bir Java istisnası, kodun bir parçasında meydana gelen istisnai koşulu (yani hatayı) tanımlayan bir nesnedir. İstisnai bir koşul ortaya çıktığında, bu istisnayı temsil eden bir nesne oluşturulur ve hataya sebep olan metodun içine fırlatılır. Bu metodun kendisi istisnayı yönetmeyi veya göz ardı etmeyi seçebilir.

Java istisna yönetimi beş anahtar üzerinden yönetilir. Bunlar try, catch, throw ve finally' dir [31].

```
try { // hataları izlenecek kod bloğu
    }
catch (istisnaTipi1 exob) {
    //istisnaTipi1 için istisna yöneticisi
    }
finally{ //try bloğu bitmeden önce çalıştırılması gereken kod bloğu
    }
```

3.10.1. İstisna tipleri

Tüm istisna tipleri yerleşik Throwable sınıfının alt sınıfıdır. Throwable' ın hemen altında, istisnaları iki farklı dala ayıran iki alt sınıf vardır. Dalın biri Exception başlığı altında toplanmıştır. Bu sınıf, kullanıcı programlarının yakalaması gereken istisnai durumlar için kullanılır. Diğer dalın başında ise, normal şartlar altında program tarafından yakalanması beklenilmeyen istisnaları tanımlayan Error bulunur. Error tipindeki istisnalar, Java çalışma zamanı ortamının kendisi ile ilgili hataları göstermek için Java çalışma zamanı sistemi tarafından kullanılır.

Program tarafından yakalanmayan her istisna, sonuçta varsayılan yönetici tarafından işlenir. Varsayılan istisna yöneticisi, istisnayı açıklayan bir karakter katırı

görüntüler, hatanın başladığı yerden itibaren bir yığın izlemesi yazdırır ve programı sonlandırır [31].

3.10.2. Try ve catch kullanımı

Bir çalışma zamanı hatasına karşı korunmak ve onu yönetmek için, izlemek istenilen kod, bir try bloğunun içine alınır. Try bloğunun hemen arkasından, yakalamak istediğiniz istisna tipini tanımlayan, bir catch cümleciği eklenir. Bir istisna fırlatıldığı zaman, programın kontrolü try bloğundan catch bloğuna geçer.

Catch cümleciğinin kapsamı, hemen öncesinden gelen try kesiminde tanımlanan ifadelerle sınırlandırılmıştır [31].

İyi yapılandırılmış catch cümleciklerinin amacı, istisna durumları çözümlmek ve hiç hata olmamış gibi programın devam etmesini sağlamaktır.

3.10.3. Throw

Programınızın açıkça throw ifadesini kullanarak bir istisnayı fırlatması mümkündür. Programın akışı throw ifadesinden sonra durur ve sonraki hiçbir ifade çalıştırılmaz. Çevreleyen en yakın try bloğunda istisna tipiyle eşleşen bir catch cümleciği aranır. Eğer bir eşleşme bulunursa kontrol bu ifadeye aktarılır.

3.10.4. Throws

Throws cümleciği bir metodun fırlatabileceği istisna tiplerini listeler. Bu, error veya RuntimeException ile bunların alt sınıfları hariç tüm istisnalar için gereklidir. Bir metodun fırlatabileceği tüm diğer istisnalar, throws cümleciği içinde tanımlanmak zorundadır.

tip metot_adi(parametre_listesi) throws istisna_listesi

```
{ //metot govdesi
}
```

3.10.5. Finally

finally, try/catch bloğu tamamlandıktan sonra çalışacak bir kod bloğu oluşturur. Try/catch bloğunun devamına yazılır. Finally bloğu bir istisna fırlatılsa da, fırlatılmasa da çalıştırılır.

```
3.26.Örnek : static void procA ( ) {
                try {    System.out.println("inside procA");
                        Throw new RuntimeException("demo");    }
                finally {System.out.println("procA's finally");    }
            }
```

3.10.6. Özel istisna alt sınıfları oluşturmak

Java' nın yerleşik istisnalar pek çok yaygın hatayı yönetebilmesine rağmen, uygulamalara özgü durumları yönetmek için özel istisna tiplerinizi oluşturulabilir. Bunu yapmak için yalnızca Exception' ın bir alt sınıfını tanımlamak yeterlidir.

3.10.7. Zincirleme İstisnalar

Zincirleme istisna özelliği bir istisna ile başka bir istisnayı ilişkilendirmenizi sağlar. İkinci istisna, birinci istisnanın nedenini tanımlar.

Throwable(Throwable istisnaNedeni)

Throwable (string msg,Throwable istisnaNedeni)

4. EĞİTİM YAZILIMI UYGULAMASININ TASARIMI

Eğitim yazılımı BDÖ yazılım geliştirme aşamalarına uygun olarak tasarlanmıştır.

4.1. Uygulamann Amacı

Bu uygulama çalışmasının amacı, Java programlama dilinin bilgisayar destekli öğretiminin gerçekleştirilmesi için bir yazılım geliştirmektir.

4.2. Uygulamada Kullanılan Teknolojiler

Bu uygulamada bölüm sayfalarının oluşturulması ve düzenlenmesi için Authorware programından, animasyonlar için Flash programından yararlanılmıştır. Ayrıca uygulamada kullanılan resimlerin düzenlenmesi için Paint Shop Pro programı ile Photoshop programı kullanılmıştır.

Uygulama çalışmasında, Java diline ait bilgilerin toplanmasında çeşitli internet sayfaları ve kitaplardan yararlanılmıştır

4.2.1. Authorware

Macromedia Authorware, web ve online eğitim uygulamalarında multimedya tabanlı bir yazarlık çözümdür. Bir öğretim materyali geliştiricisi, bir eğitim tasarımcısı, eğitimci veya konu uzmanı olarak multimedya tabanlı eğitim uygulamaları geliştirebilir, bunları öğrencilere Web, LAN veya CD-ROM üzerinde sunabilir ve öğrencilerin sonuçlarını derleyerek verilen eğitim değerlendirilebilir.

Authorware eğitim tasarımcılarının işini kolaylaştıracak birçok özellik ile donatılmıştır. Authorware'de her türlü olay veya etkileşim bir nesne aracılığıyla yapılır. Nesnelere, yukarıdan aşağıya devam eden "akış diyagramı"na sürüklenip bırakılarak projeye eklenmiş olurlar. Bu şekilde oluşturulan projelere resim, ses, video, flash animasyonları, etkileşimler eklenebilir.

Authorware'in hazır şablonu üzerinde etkileşimli alıştırmalar, konular, sözlük, arama gibi düğmeler oluşturularak etkileşimli ve konuyu bütün yönleriyle kavratan eğitim araçları vardır.

4.2.2. Flash

Flash bir vektör grafik, animasyon ve etkileşim programıdır. Flash'ta animasyonlar bir sahne üzerinde bir zaman çizelgesi kullanılarak oluşturulur. Flash ile öğrencilerin dikkatini aktif tutmasını ve konunun özünü kavramasını sağlayacak "animasyonlu" konu anlatımları oluşturuldu.

4.3.Uygulamanın Bölümleri ve Tasarımı

Uygulama konular ve testler bölümünden oluşmaktadır.

4.3.1. Konular bölümü

Eğitim yazılımı toplam 10 bölümden oluşmaktadır. Bu bölümler öğrencilerin anlayabileceği şekilde çeşitli kaynaklardan derlenerek oluşturulmuştur. Ekranda her bölümün adı ve bölüm alt konu başlığı bulunmaktadır. Ayrıca bu ekranda bölümün kaç sayfadan oluştuğu ve öğrencinin kaçınıcı sayfada olduğu görülmektedir. Ders ekranında öğrencinin diğer bölümlere ulaşmasını sağlayacak, istediği konular arasında rahatça dolaşabilmesini sağlayacak düğmeler bulunmaktadır. Öğrenci istediği bölüm ve konulara bu düğmelerle ulaşabilir.

Her bölümün sonunda özet bölümü bulunmaktadır. Bu özet ekranından sonra öğrencinin bölümle ilgili test ekranına ulaşmasını sağlayacak bir düğme bulunmaktadır.

Bu ekrandaki Konular düğmesi ile öğrenci bölümlerin bir listesini görür ve istediği konuya geçiş sağlayabilir. Sözlük düğmesiyle Java ile ilgili terimlerin anlamlarını

görebilir. Ara düğmesiyle istediği bir kelimeyi arattırabilir. Ses düğmesiyle konuların sesli anlatımına ulaşılabilir. Hedefler düğmesiyle bu yazılım hedeflerine ulaşılabilir.

4.3.2. Test bölümü

Her bölümün sonunda 10 sorudan oluşan test bölümü bulunmaktadır. Öğrenci test bölümüne geçmek için konu sonunda bulunan özet sayfasında bulunan test düğmesine tıklayarak ulaşmaktadır. Test ekranında öğrenci kaçınıcı bölümün testinde olduğunu ve kaçınıcı soruda olduğunu görebilmektedir. Sorular sorulduktan sonra öğrencinin doğru cevap verip vermediği anında öğrenciye gösterilir. Testin sonunda öğrenciye test özeti verilerek kaç soruya doğru kaç soruya yanlış cevap verdiği ve aldığı puan gösterilir.

4.3.3. Animasyonlar

Bölüm konularında bazı konuların daha iyi anlaşılabilmesi için Flash programı kullanılarak oluşturulmuş animasyonlara yer verilmiştir. Öğrenci bu animasyonları istediği kadar izleyebilmektedir. Böylece konunun daha iyi anlaşılması amaçlanmaktadır.

4.3.4. Etkileşimli örnek sorular

Konu aralarında konuyu pekiştirici örnek etkileşimli sorular bulunmaktadır. Bu sorular çoktan seçmeli sorular olabileceği gibi sürükle bırak yöntemi ile oluşturulmuş sorular olabilir. Öğrenci bu soruları tekrar tekrar çözebilir. Öğrenciye dönütler anında verilir.

4.3.5. Seslendirme

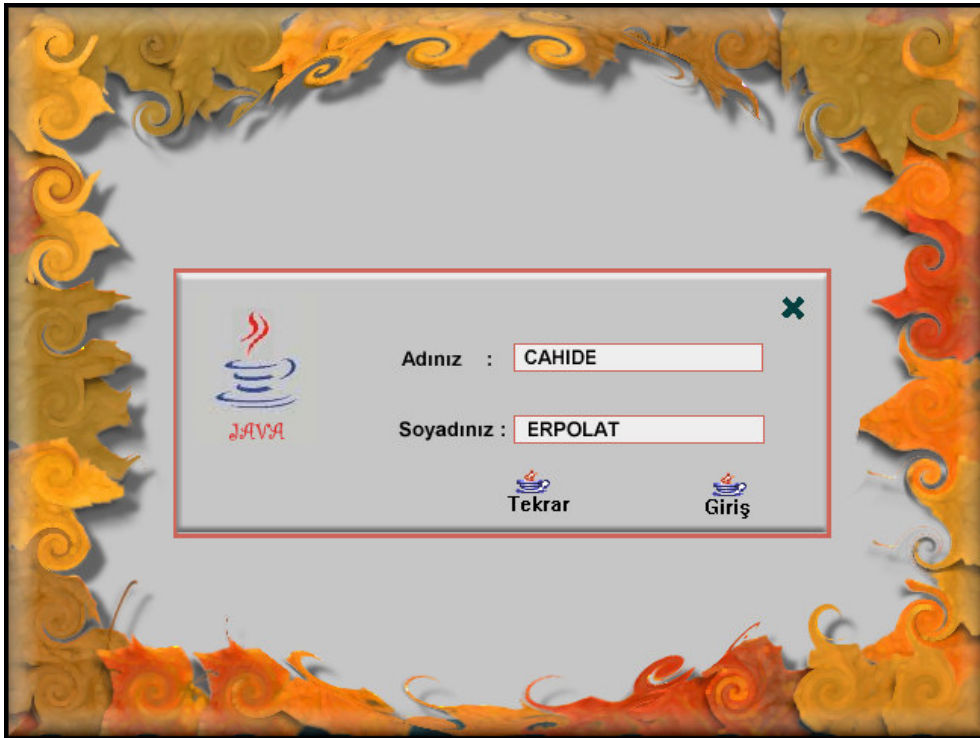
Konu ekranında ses düğmesine tıklanarak ekrandaki konuların sesli anlatımına geçilebilir. Sesi kapatmak için tekrar ses düğmesine basmak yeterlidir.

4.4. Eğitim Yazılımının Kullanımı

4.4.1. Eğitim yazılımının başlangıç ekranı

Resim 4.1’de gösterilen ekranda kullanıcı adını ve soyadını girip Giriş düğmesine basarak programı başlatabilir. Ad ve soyad bilgilerinin girişinde yanlış yapıldığında Tekrar düğmesi kullanılarak doğru bilgilerin girişi sağlanabilir. Bu ekranda programa giriş yapılmadan Çıkış düğmesine basılarak programdan çıkılabilir.

Bundan sonraki ekranda yazılımın içerdiği bölüm adları ve bu bölümlerin testleri listelenir. Burada istenilen bölümlere ve bu bölümlerin testlerine geçme imkânı sağlanır. Bu ekranda Konular, Sözlük, Hedefler, Ara, Ses, Giriş ve Yardım düğmeleri bulunmaktadır.



Resim 4.1. Eğitim yazılımının başlangıç ekranı



Resim 4.2. Eğitim yazılımının ikinci ana ekranı

4.4.2. Konular bölümü

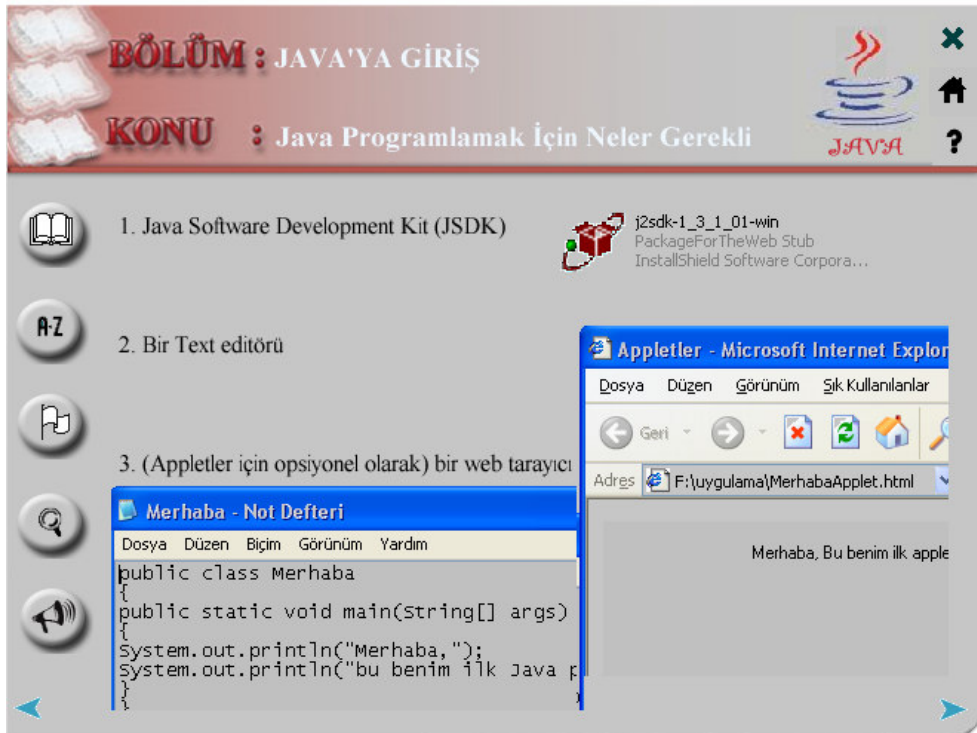
Bu ekranda yazılımın bölüm isimleri ve testleri listelenmiştir. Öğrenci bu ekranda istediği bölüm veya test adına tıklayarak bunlara giriş sağlayabilir.

4.4.3. Konu ekranı

Öğrenci istediği bölüm ismine tıkladıktan sonra o bölümün Resim 4.3'te gösterilen giriş ekranına ulaşır. Öğrenci sonraki sayfa düğmesini kullanarak Resim 4.4'te gösterilen bölümün konu ekranına geçebilir. Konu ekranında öğrencinin seçmiş olduğu bölüm ve konu ismi görüntülenir. Ayrıca bu ekranda o bölümü oluşturan sayfa sayısı, öğrencinin kaçınıcı sayfada olduğu da görülür. Öğrenci bu ekranda sonraki sayfa ve önceki sayfa düğmelerine tıklayarak sayfalar arasında dolaşabilir. Öğrenci ayrıca sonraki bölüm ve önceki bölüm düğmeleri ile de bölümler arasında da dolaşabilir. Bu ekranda ayrıca Konular, Ara, Hedefler, Yardım, Giriş, Kapat düğmeleri bulunmaktadır.



Resim 4.3. Eğitim yazılımının bölüm giriş ekranı

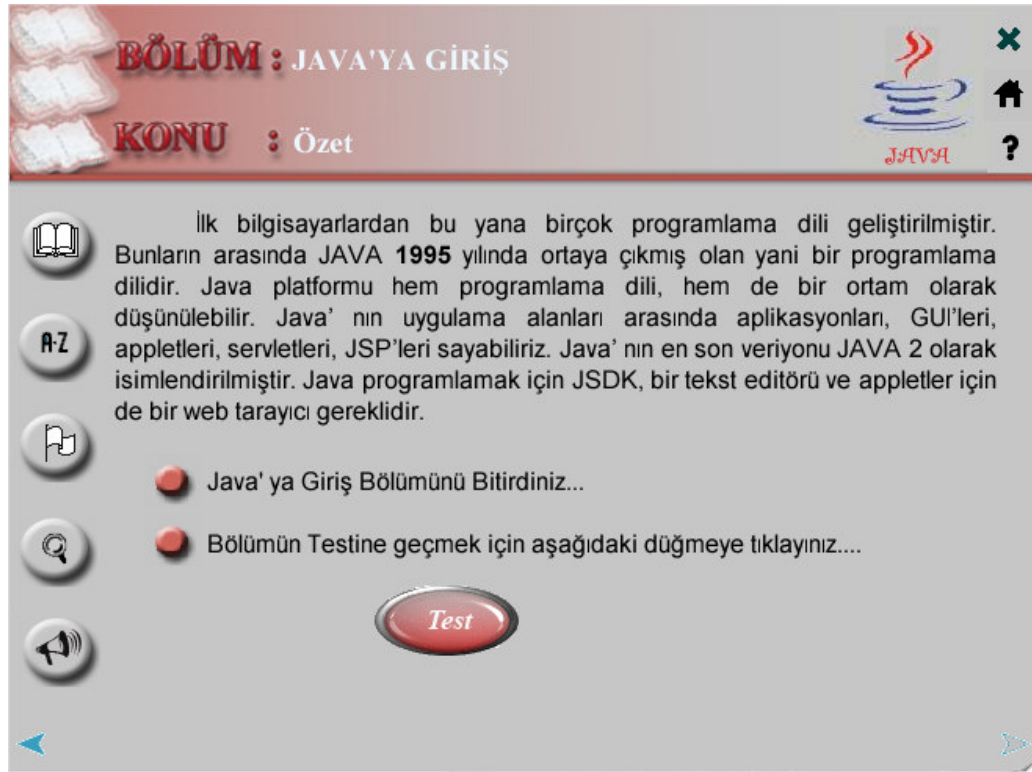


Resim 4.4. Eğitim yazılımının konu ekranı

4.4.4. Testler bölümü

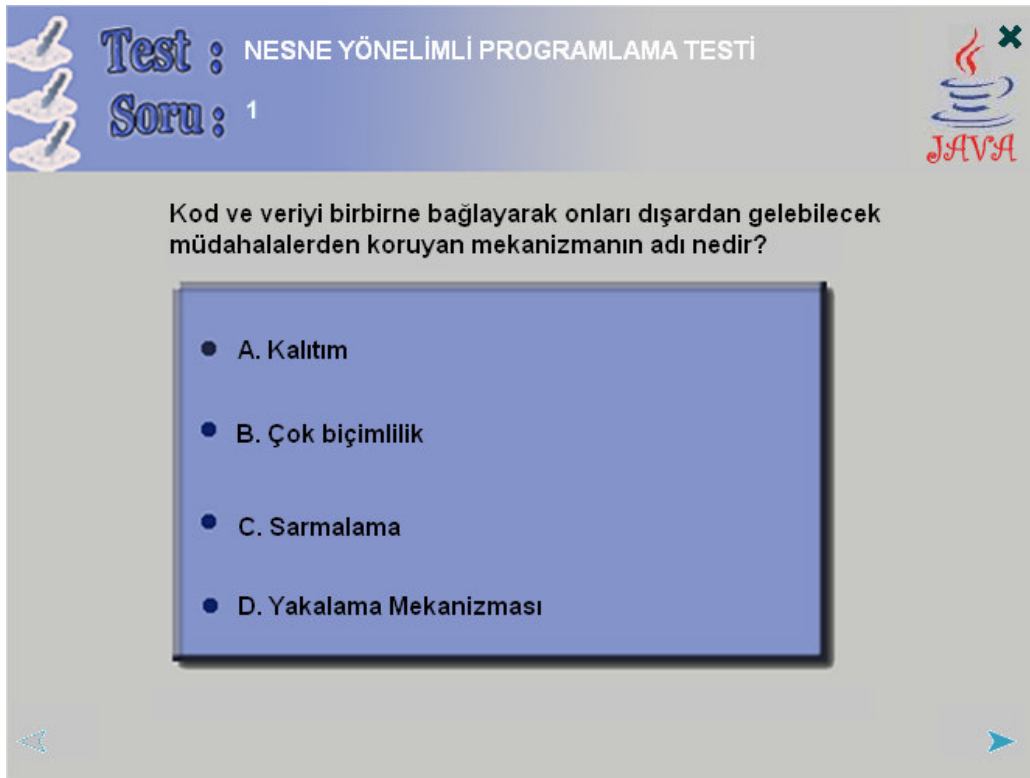
Öğrenci Resim 4.2' de gösterilen ikinci ana pencerede istediği bölümün testine tıkladıktan sonra öğrencinin karşısına ilk önce Resim 4.5'te gösterilen örnekteki gibi o bölümün özet ekranı gelir. Ayrıca bu ekrana bölüm bitirilerekte ulaşılabilir. Bölümün testine geçmek için bu ekrandaki Test düğmesine tıklamak gerekir.

Resim 4.6'daki Test ekranında öğrenci sorulara isterse fare ile tıklayarak veya klavyeden seçeneklerin harflerine tıklayarak cevap verebilir. Öğrencinin verdiği cevabın doğru olup olmadığı anında öğrenciye bildirilir. Öğrenci bu ekrandaki ileri ve geri tuşuyla sorular arasında dolaşabilir.



Resim 4.5. Eğitim yazılımının özet ekranı

Öğrenci bütün soruları cevapladıktan sonra Resim 4.7'deki test sonucu ekranı karşısına gelir. Burada öğrencinin vermiş olduğu doğru soru sayısı, yanlış soru sayısı ve testten aldığı puan öğrenciye gösterilir.



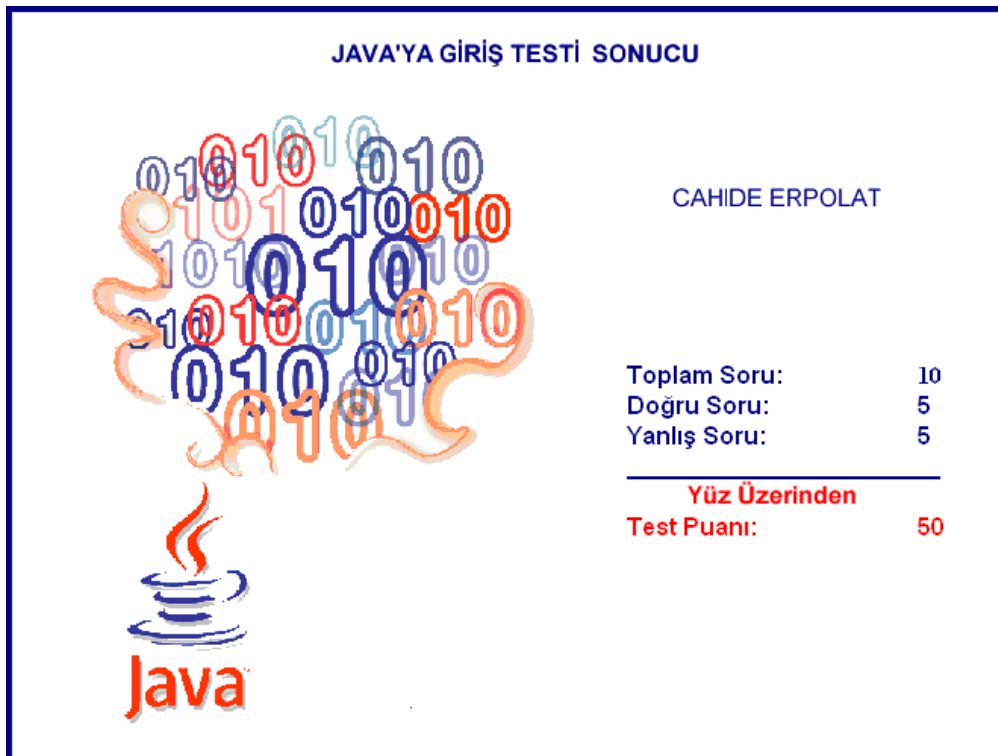
Test : NESNE YÖNELİMLİ PROGRAMLAMA TESTİ

Soru : 1

Kod ve veriyi birbirine bağlayarak onları dışardan gelebilecek müdahalelerden koruyan mekanizmanın adı nedir?

- A. Kalıtım
- B. Çok biçimlilik
- C. Sarmalama
- D. Yakalama Mekanizması

Resim 4.6. Test ekranı



JAVA'YA GİRİŞ TESTİ SONUCU

CAHIDE ERPOLAT

Toplam Soru:	10
Doğru Soru:	5
Yanlış Soru:	5

Yüz Üzerinden
Test Puanı: 50

Java

Resim 4.7. Test sonucu ekranı

4.4.5. Animasyonlar

Eđitim yazılımındaki animasyonlara ulařmak için Resim 4.8'deki düđmeye tıklamak gerekir. Bu düđmeye tıklandıktan sonra Resim 4.9'da gösterildiđi gibi ilgili animasyon ekranda gözükücektir.

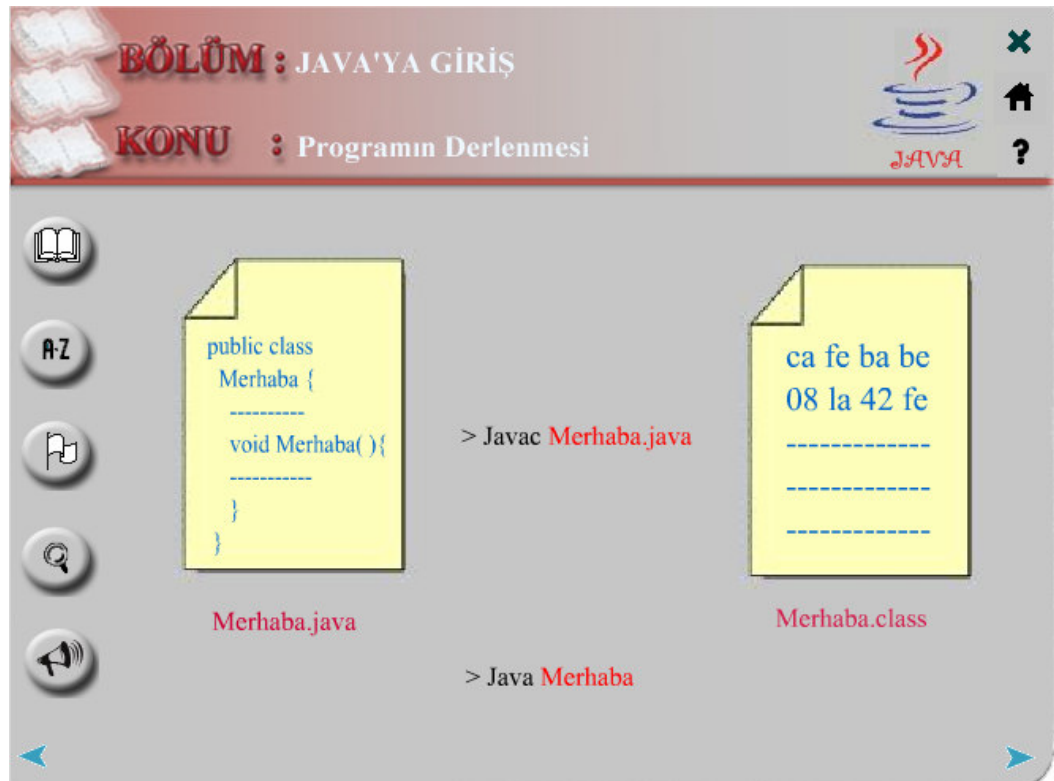


Animasyon

Resim 4.8. Animasyon düđmesi

4.4.6. Etkileřimli Örnek Sorular

Etkileřimli örnek sorularına yine animasyon düđmelerine tıklayarak ulaşabiliriz. Öğrenci Resim 4.10'da gösterilen örnek soruları ekranda verilen komutlarla çözer. Öğrenciye vermiř olduđu cevabın dođru veya yanlıř olduđu anında gösterilir.



Resim 4.9. Örnek animasyon ekranı



Resim 4.10 Etkileşimli Örnek soru

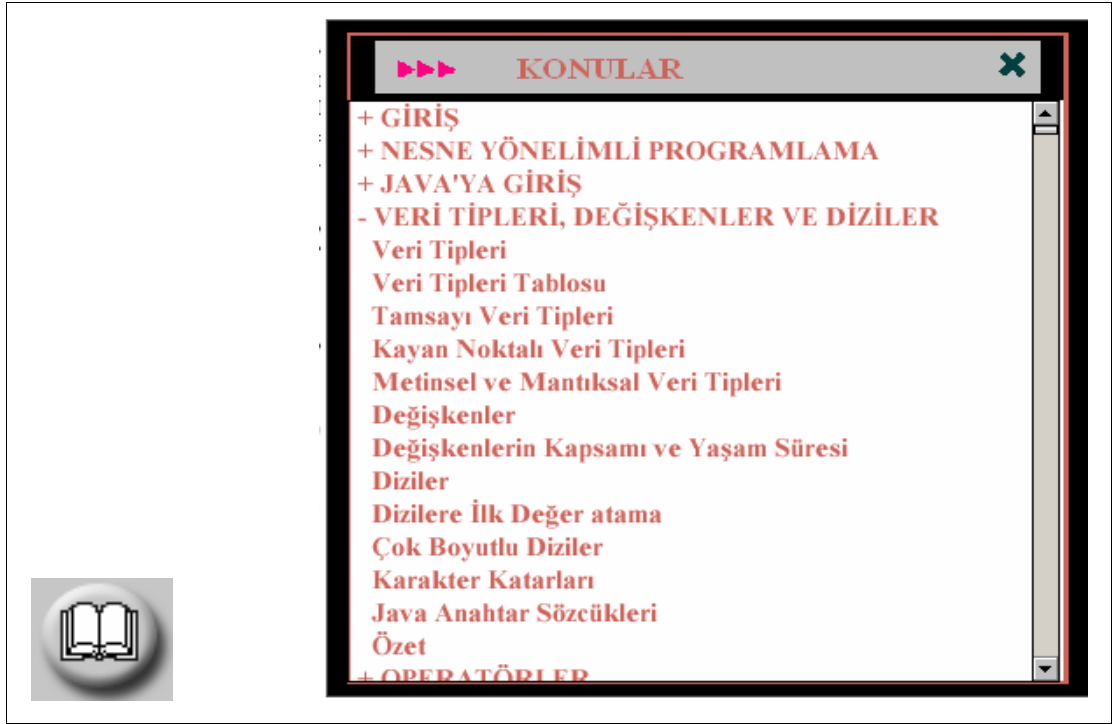
4.5. Uygulamada Kullanılan Düğmeler

4.5.1. Konular düğmesi

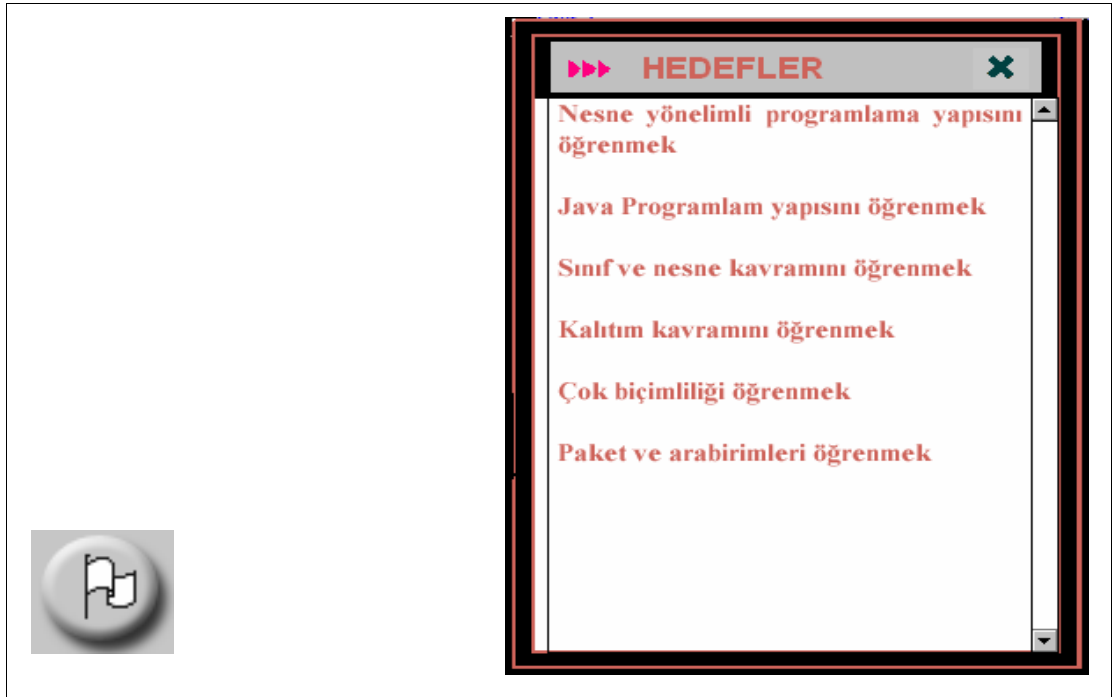
Resim 4.11’de gösterilen düğmeye basıldığında yine Resim 4.11’de gösterilen Konular penceresi ekrana gelir. Bu pencerede eğitim yazılımının bölümleri listelenmiştir. Öğrenci bu bölümlerin adına tıkladığında bölüm alt konuları listelenir. Öğrenci bu konu ismine tıklayarak istediği konu penceresine ulaşabilir.

4.5.2. Hedef düğmesi

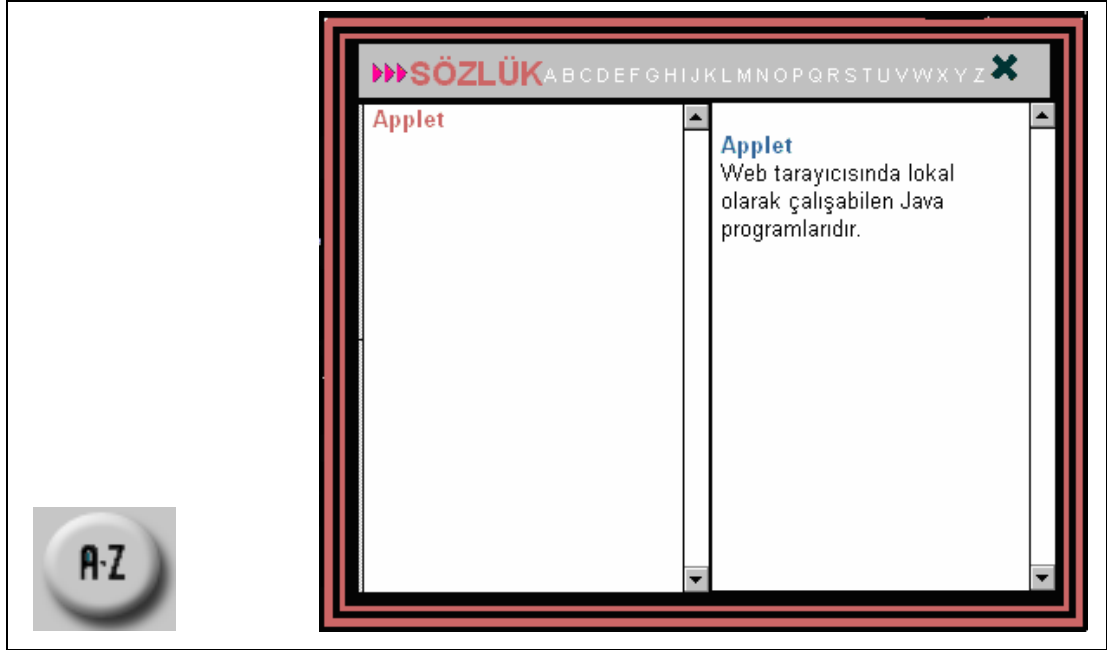
Resim 4.12’de gösterilen düğmeye tıkladığında yine Resim 4.11’de gösterilen pencere ekrana gelir. Bu pencerede eğitim yazılımının amaçladığı hedefler listelenir.



Resim 4.11. Konular düğmesi ve penceresi



Resim 4.12. Hedefler düğmesi ve penceresi



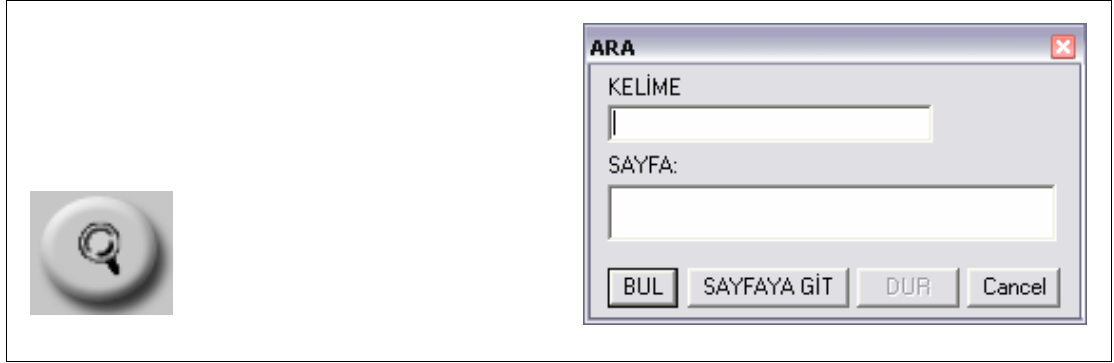
Resim 4.13. Sözlük düğmesi ve penceresi

4.5.3. Sözlük düğmesi

Resim 4.13'te gösterilen düğmeye tıklandığında yine Resim 4.13'te gösterilen pencere ekrana gelir. Bu pencerede Java ile ilgili terimlerin anlamlarına ulaşılabilir. Bunun için üst taraftaki harflere tıklamak, daha sonra da sol taraftaki kelime isimlerine tıklanılması gerekir.

4.5.4. Ara düğmesi

Resim 4.14'te gösterilen düğmeye tıklandığında yine Resim 4.14'te gösterilen pencere ekrana gelir. Bu pencerede kelime kısmına aranılacak kelime yazılır ve bul düğmesine tıklanır. Aranılan kelimenin bulunduğu sayfalar sayfa bölümünde görülür. Bu sayfa isimlerine tıklanılarak o sayfalara ulaşılabilir.



Resim 4.14. Ara düğmesi ve penceresi

4.5.5. Ses düğmesi

Resim 4.15'e tıklandığında konuların sesli anlatımına ulaşılabilir.



Resim 4.15 Ses düğmesi

4.5.6. Yardım düğmesi

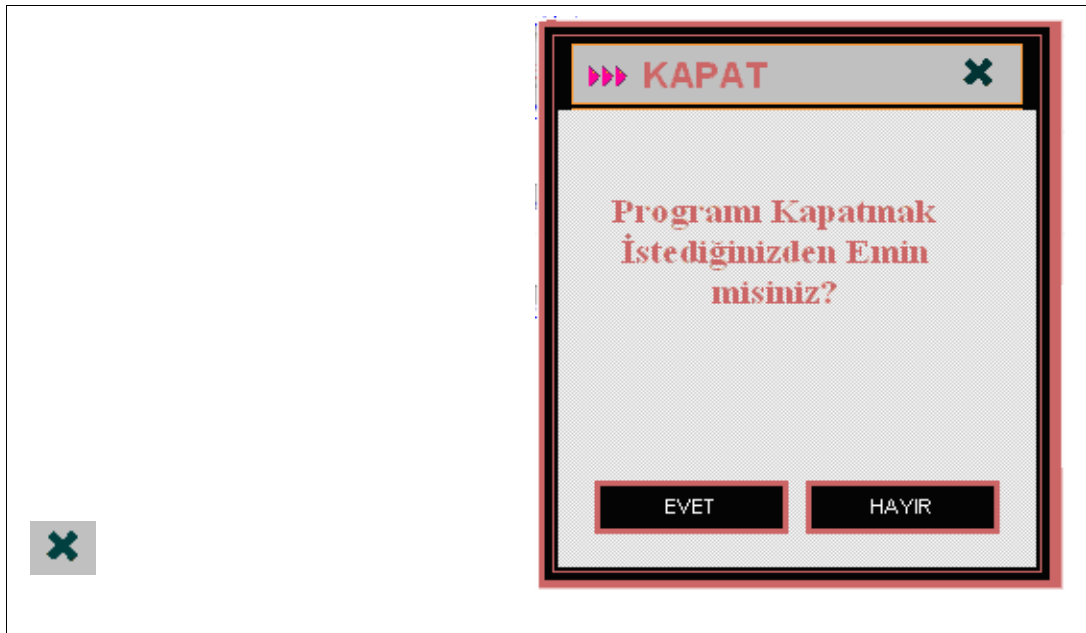
Resim 4.16'te gösterilen Yardım düğmesine tıklandığında yine Resim 4.15'te gösterilen Yardım penceresi ekrana gelir. Bu pencerede eğitim yazılımında kullanılan düğmelerin görevleri anlatılmıştır.

4.5.7. Kapat düğmesi

Resim 4.17'de gösterilen Kapat düğmesine tıklandığında yine Resim 4.16'da gösterilen Kapat penceresi ekrana gelir. Bu pencerede öğrencinin programdan çıkmak istediği onaylandıktan sonra programdan çıkış sağlanır.



Resim 4.16. Yardım düğmesi ve penceresi



Resim 4.17. Kapat düğmesi ve penceresi

4.5.8. Giriş düğmesi

Şekil 4.18’de gösterilen düğme kullanılarak Resim 4.2’ de gösterilen yazılımın ikinci ana ekranına ulaşılabilir.



Resim 4.18. Giriş düğmesi

4.5.9. Sonraki sayfa düğmesi

Resim 4.19’da gösterilen düğme bir sonraki sayfaya geçmek için kullanılır.



Resim 4.19. Sonraki sayfa düğmesi

4.5.10. Önceki sayfa düğmesi

Resim 4.20’da gösterilen düğme bir önceki sayfaya geçmek için kullanılır.



Resim 4.20. Sonraki sayfa düğmesi

4.5.11. Sonraki bölüm düğmesi

Resim 4.21’de gösterilen düğme bir sonraki bölüme geçmek için kullanılır.



Resim 4.21. Sonraki sayfa düğmesi

4.5.12. Önceki bölüm düğmesi

Resim 4.22’de gösterilen düğme bir önceki bölüme geçmek için kullanılır.



Resim 4.22. Önceki bölüm düğmesi

4.5.13. Animasyon düğmesi

Resim 4.23’de gösterilen düğme animasyon veya etkileşimli örnek soru ekranına geçmek için kullanılır.



Animasyon

Resim 4.23. Animasyon düğmesi

4.5.14. Sonraki komut düğmesi

Resim 4.24’de gösterilen düğme bir sonraki komuda geçmek için kullanılır.



Resim 4.24. Sonraki komut düğmesi

4.5.15. Önceki komut düğmesi

Resim 4.25’te gösterilen düğme bir önceki komuda geçmek için kullanılır.



Resim 4.25. Önceki komut düğmesi

5. SONUÇ VE ÖNERİLER

Bilgisayarlar, bilgi toplumunun vazgeçilmez araçlarıdır. Bilgisayar teknolojisi en büyük etkiyi eğitim alanında yapmaktadır. Bilgisayarlar, tüm dünyada okullara ve evlere etkili bir eğitim aracı olarak hızla girmektedir. 22.yüzyılda bilgisayar teknolojisinin eğitim sisteminin deęişmez bir parçası olması kaçınılmazdır. Ancak bu teknoloji bir amaç deęil, yalnızca bir araçtır.

Bilgisayar Destekli Öğretim' in eğitim alanında yarattığı belki de en önemli olanak, hem öğretmenin hem de öğrencinin ihtiyaç duydukları bilgiye bilgisayar teknolojisi sayesinde kitap sayfalarının donukluęuna hapsedilmiş bir yazı ya da resim olarak deęil, tam tersine, tüm duyuları harekete geçiren ses, yazı, grafik, resim, film, canlandırma öğelerinin tümünü içeren canlı, renkli ve çekici bir bütün olarak ulaşmalarıdır. Bilginin zenginlięi ve bu bilgiye ulaşım kolaylığı da öğrenmeye yeni bir biçim kazandırır.

Araştırmalara göre, aktif olarak bilgisayar destekli öğretim olanaklarından yararlanan öğrencilerin standart deęerlendirme metotlarıyla yapılan incelemelerde başarılı bulunmalarının yanı sıra, genellikle ölçülmeyen birçok beceriyi de kazandıkları saptanmıştır.

Java 1995 yılında ortaya çıkmış yeni bir programlama dilidir. Java dilinin basit, dinamik, nesneye yönelik, taşınabilir, çok görevli, çok kullanımlı olması ve farklı platformlarda çalışması kullanımını artırmıştır. Bu tez çalışmasında, belirtilen bu unsurlar ve iyi bir eğitim yazılımı geliştirme aşamaları göz önüne alınarak Java programlama dilinin bilgisayar destekli öğretimine yönelik bir eğitim yazılımı tasarlanmış ve geliştirilmiştir.

Eğitim yazılımı geliştirilirken BDÖ yazılım geliştirilme aşamaları göz önüne alınarak ders hedefleri ve öğrenci gereksinimleri belirlenmiştir. Bunun için çeşitli Java kaynak kitaplarından yararlanılarak eğitim yazılımının içerięi hazırlanmıştır.

Ders hedefleri, kavram haritası ve öğrenci zorlukları göz önüne alınarak konunun bilgisayar olanaklarıyla etkili bir şekilde öğretilmesi için konu 10 ana bölüme ayrılmıştır. Bu bölümlerde içeriklerine göre konulara ayrılmıştır. Her bölüm sonunda öğrenciye bölüm özeti hazırlanmıştır. Ayrıca bölüm özetinden sonra öğrencinin bilgisini ölçmesini sağlayacak 10 sorudan oluşan test hazırlanmıştır.

Bütün bu aşamalardan sonra yazılım Authorware programı kullanılarak geliştirilmiştir. Yazılımda konular, sözlük, hedefler, ara, ses, yardım, giriş ve kapat düğmeleri bulunmaktadır. Bu düğmeler öğrencinin yazılımı daha etkili bir şekilde kullanılması için geliştirilmiştir.

Öğrenilmesi zor olan konularda öğrencinin konuyu daha iyi kavraması için Flash programı kullanılarak hazırlanmış animasyonlara yer verilmiştir. Ayrıca programlama dillerinin öğrenilmesinde karşılaşılan zorluklar göz önüne alınarak örnek program parçalarının öğrenci kontrolünde adım adım çalıştırılmasına olanak verilmiştir. Böylelikle öğrenci programı adım adım çalıştırarak her adımda ne olduğunu görmekte ve sonra programın ekran çıktısını görmektedir.

Konular seslendirilmiştir. Öğrenci istediği anda ses düğmesine basarak konuların sesli anlatımına ulaşabilir. Eğitim yazılımı kullanım kolaylığı açısından CD veya internet ortamında kullanılabilir. Böylelikle bu yazılım istendiğinde internet üzerinden de kullanılabilir.

Türkiye'de eğitim teknolojisi alanındaki çalışmalar giderek artmaktadır. Ancak bunların gereksinimleri karşılayamamaktadır. Özellikle yeni bilgi teknolojilerinin eğitim kurumlarımıza girmesi çok yavaş olmaktadır.

Türk eğitim sistemine bilgisayar destekli eğitimin girmesi için öncelikle öğretmenlerin bu konuda yeterli bilgi ve becerilerle donatılmalarına gerek vardır. Bunun için hem eğitim fakültelerinin tüm programlarında bunlarla ilgili derslere yer verilmeli hem de çalışan öğretmenler hizmetiçi eğitim programlarından geçirilmelidir. Bunun yanısıra, okulların teknolojik donanımlarının sağlanmasına ve

eđitimde kullanılabilir nitelikte gereçlerin hızla geliştirilerek kullanıma sunulmasına gerek vardır.

KAYNAKLAR

1. İnternet: Bilgisayar Destekli Eğitim, Zaman Online. <http://www.zaman.com.tr/?hn=116324&bl=yazarlar&trh=20041129> (2004).
2. Çekbaş Y., Yakar H, Yıldırım B., Savrana A. ,“Bilgisayar Destekli Eğitimin Öğrenciler Üzerine Etkisi”, *The Turkish Online Journal of Educational Technology - TOJET*, 2 (4): 1-2 (2003).
3. Akpınar, Dr.Y., “Bilgisayar Destekli Öğretim ve Uygulamalar”, *Anı Yayıncılık*, Ankara, 175-209 (1999).
4. Ergün Mustafa, “İnternet Destekli Eğitim”, *Afyon Kocatepe Üniversitesi Sosyal Bilimler Dergisi*, 1 (11):1-10 (2005).
5. Meer, V. M., “PLATO:From computer-based education to corporate social responsibility”, *Iterarions: An Interdisciplinary Journal of Software History*, 2 :1-22(2003).
6. Keser, H. “Bilgisayar destekli eğitim için bir model önerisi”, Doktora Tezi, *Ankara Üniversitesi Sosyal Bilimler Enstitüsü*, Ankara, 10-20 (1988).
7. Şimşek, B., “Programlama dilleri öğretimi için anlayışlı bir eğitim sisteminin tasarımı ve gerçekleştirimi”, Yüksek Lisans Tezi, *Hacettepe Üniversitesi Fen Bilimleri Enstitüsü*, Ankara, 16-22 (1990).
8. Çomonoğlu, D., “Pascal için bir bilgisayar destekli öğretim sistemi tasarımı ve gerçekleştirimi”, Yüksek Lisans Tezi, *Hacettepe Üniversitesi Fen Bilimleri Enstitüsü*, Ankara, 29-53 (1990).
9. Mutlu, E., “Java tabanlı bir eğitsel yazılım tasarımı”, Yüksek Lisans Tezi, *İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü*, İstanbul, 37-58 (2001).
10. Dokümantasyon,L., “Bilgisayar destekli eğitim öğretmen klavuzu”, *Logomotif Yayınları*, Ankara, 5-22 (1999).
11. Uşun, S., “ Dünyada ve Türkiye’de Bilgisayar Destekli Öğretim”, *Pagema Yayıncılık*, Ankara, 51-79,149-207,209-308 (2000).
12. Şimşek, N., “Derste eğitim teknolojisinin kullanımı”, *Nobel Yayın*, Ankara, 5-35,57 (2002).
13. Aytaş, Gıyasettin., Eğitimde yeni teknoloji kullanımındaki virajlarda bilgisayar destekli eğitimin düşündürdükleri, *Milli Eğitim Vakfı Dergisi*, 1:24 (1991).

14. Ögüt, H., Altun, A.A., Sulak, S.A, Koçer, H.E, “Bilgisayar destekli, internet erişimli interaktif eğitim cd’si ile e-eğitim”, *The Turkish Online Journal of Educational Technology-TOJET*, 3 (10): 1-7 (2004).
15. Demirci, Yrd.Doç.Dr.N., “Bilgisayarla etkili öğretim stratejileri ve fizik öğretimi”, *Nobel Yayın*, Ankara, 41-95 (2003).
16. Balki, E., “Bilgisayar destekli eğitim”, *Akademik Bilişim 2002*, Konya, 12-21 (2002).
17. İnternet: Bilgisayarlar ve Eğitimde Kullanımı, Bizim Dergi. <http://www.angelfire.com/al/AtaturkAnadoluMeslek/dergi.html#Bilgisayarlar%20Ove%20Eğitimde%20Kullanım> (1997).
18. Ekiz, H., Bayam, Y., Ünal H., “Mantık devreleri dersine yönelik internet destekli uzaktan eğitim uygulaması”, *The Turkish Online Journal of Educational Technology-TOJET*, 2 (4): 1-8 (2003).
19. Akkoyunlu, Doç.Dr.B., İmer, Dr.G, “Türkiye’de eğitim teknolojisinin görünümü”, Çağdaş eğitimde yeni teknolojiler, Özer, Doç.Dr.B., *Anadolu Üniversitesi*, Eskişehir, 160-168 (1998).
20. Pekköz N., “Java”, *Pusula Yayıncılık*, İstanbul, 5-49,169-189 (2004).
21. Ganiz, T., “Java eğitimi araştırması”, Yüksek Lisans Tezi, *Marmara Üniversitesi Fen Bilimleri Enstitüsü*, İstanbul, 1-2,13-17,35-42 (1999).
22. Adeli, H., Kim, H., “Web-based interactive courseware for structural steel design using java”, *Computer-Aided Civil and Infrastructure Engineering*, 15: 158-166 (2000).
23. Valley, S., “Learning Java with Jbuilder”, *Insprise*, USA, 9/1-19/6,14/1,14/7 (2001).
24. Fernandez, J. M., “Java”, *Linuxfocu MAGazines*, 1-4 (1998).
25. Celements, N., Williams, G., “Java programming”, *Oracle*, USA, 2, 8/2-8-19 (2000).
26. Güngören, B., “Java ile temel programlama”, *Seçkin*, Ankara, 209-293 (2005).
27. Deitel, H. M., Deitel, P.J., “Java How to program”, Fourth Edition, *Prentice Hall*, New Jersey, 313-350,536-583 (2002).
28. Schildt, H., “Java 2”, *Alfa Yayınları*, İstanbul, 25-80 (2003).

29. İnternet: The Java™ Virtual Machine Specification.
<http://java.sun.com/docs/books/vmspec/html/VMSpecTOC.doc.html> (1996).
30. Altıntaş, A. B, “Java ve Yazılım Tasarımı”, *Papatya Yayıncılık*, İstanbul,131-154 (2003).
31. Şenyurt, B.S, “Java ile 24 kahve molası”, İstanbul, 131-140 (2004).

ÖZGEÇMİŞ

Kişisel Bilgiler

Soyadı, adı : ERPOLAT, Cahide
 Uyuğu : T.C.
 Doğum tarihi ve yeri : 12.08.1979 Gercüş
 Medeni hali : Bekar
 Telefon : 0 (312) 239 28 23
 Telefon (cep) : 0 505 243 17 46
 e-mail : tef9862032@hotmail.com, cerpolat@meb.gov.tr

Eğitim Derece

Eğitim Derece	Eğitim Birimi	Mezuniyet tarihi
Lisans	Gazi Üniversitesi Bilgisayar Sistemleri Öğretmenliği	2002
Lise	Çankaya Gazi Osman Paşa AML	1997

İş Deneyimi

Yıl	Yer	Görev
2002-2004	Beypazarı Cumhuriyet İÖO	Teknik öğretmen
2004-2005	Sincan Teknik ve EML	Teknik öğretmen
2005-2006	MEB Eğitek Ağ Yönetim Şubesi	Network Uzmanı

Yabancı Dil

İngilizce	KPDS	59
-----------	------	----

Hobiler

Bilgisayar programları yazmak, bilim kitapları okumak, fuarları ve tarihi eserleri gezmek