

Java ile

Nesneye Yönelik

Programlama

| | | |
|-------|--|----|
| 1 | Nesneye Yönelik Programlamaya Giriş..... | 9 |
| 1.1 | Modelleme | 9 |
| 1.2 | Bilgisayar Programları ve Modelleme | 10 |
| 1.3 | Nesneye Yönelik Modelleme | 11 |
| 1.3.1 | Sınıf ve Nesne Kavramları | 11 |
| 1.4 | Nesneye Yönelik Programlama | 13 |
| 1.5 | Nesneye Yönelik Çözümleme ve Tasarım | 15 |
| 1.6 | Nesneye Yönelik Programlama Dilleri | 15 |
| 1.7 | Özet..... | 16 |
| 2 | Java Platformu | 18 |
| 2.1 | Platformdan Bağımsızlık | 19 |
| 2.2 | Java Sanal Makinesi (<i>Java Virtual Machine - JVM</i>) | 22 |
| 2.3 | Java Uygulama Programlama Arayüzü (<i>Java Application Programming Interface - API</i>)..... | 25 |
| 2.4 | Özet..... | 28 |
| 3 | Java Dili İle İlgili Temel Bilgiler | 30 |
| 3.1 | Sözdizim Kuralları (<i>Syntax Rules</i>) | 30 |
| 3.1.1 | Derleyiciler..... | 30 |
| 3.1.2 | <i>Whitespace</i> karakterleri..... | 30 |
| 3.2 | Açıklama Satırları (<i>Comment Lines</i>) | 31 |
| 3.2.1 | ile yapılan açıklamalar | 32 |
| 3.2.2 | * ... */ ile yapılan açıklamalar | 32 |
| 3.2.3 | ** ... */ ile yapılan açıklamalar | 33 |
| 3.3 | Temel Türler (<i>Primitive Types</i>) | 35 |

| | | |
|-------|--|----|
| 3.4 | Değişken Tanımlama ve İlk Değer Atama (<i>Variable Declaration and Initialization</i>) | 36 |
| 3.4.1 | Değişkenin Tanım Alanı (<i>Scope</i>)..... | 37 |
| 3.5 | İşleçler (<i>Operators</i>)..... | 38 |
| 3.5.1 | Aritmetik işleçler:..... | 38 |
| 3.5.2 | Birleşik Aritmetik İşleçler: | 38 |
| 3.5.3 | Arttırma ve Azaltma İşleçleri: | 38 |
| 3.5.4 | İlişkisel (<i>relational</i>) işleçler: | 40 |
| 3.5.5 | Mantıksal İşleçler: | 40 |
| 3.5.6 | İşleçlerin Öncelikleri: | 41 |
| 3.6 | Denetim Deyimleri (<i>Control Statements</i>) | 42 |
| 3.6.1 | <code>if - else</code> deyimi | 42 |
| 3.6.2 | <code>if - else if</code> deyimi | 45 |
| 3.6.3 | <code>switch</code> deyimi | 46 |
| 3.7 | Döngüler (<i>Loops</i>)..... | 48 |
| 3.7.1 | <code>while</code> döngüsü | 48 |
| 3.7.2 | <code>do - while</code> döngüsü..... | 49 |
| 3.7.3 | <code>for</code> döngüsü | 50 |
| 3.8 | Özet..... | 52 |
| 4 | Uygulama Geliştirme Ortamı | 53 |
| 4.1 | JDK ve JRE | 53 |
| 4.2 | Java ile ilk program..... | 55 |
| 4.2.1 | İlk program..... | 56 |
| 4.2.2 | Derleme..... | 56 |
| 4.2.3 | Çalıştırma ve <code>main</code> yöntemi | 57 |

| | | |
|-------|---|-----|
| 4.3 | Eclipse Platformu | 59 |
| 5 | Java ile Nesneye Yönelik Programlama | 61 |
| 5.1 | Sınıf Tanımları | 61 |
| 5.2 | Nesne Oluşturma | 66 |
| 5.2.1 | <code>new</code> işleci | 66 |
| 5.2.2 | Referans Tür | 67 |
| 5.2.3 | Yığın (<i>Heap</i>), Yığıt (<i>Stack</i>) ve Çöp Toplayıcı (<i>Garbage Collector</i>) | 67 |
| 5.2.4 | Kurucular (<i>Constructors</i>)..... | 70 |
| 5.2.5 | <code>this</code> anahtar sözcüğü..... | 73 |
| 5.2.6 | Kurucu Aşırı Yükleme (<i>Constructor Overloading</i>) | 74 |
| 5.2.7 | Kurucuların Birbirini Çağırması..... | 77 |
| 5.2.8 | Yöntem Aşırı Yükleme (<i>Method Overloading</i>) | 79 |
| 5.2.9 | Diziler..... | 81 |
| 5.3 | Sarmalama (<i>Encapsulation</i>) İlkesi ve Erişim Düzenleyiciler (<i>Access Modifiers</i>)..... | 86 |
| 5.3.1 | Sarmalama İlkesi | 86 |
| 5.3.2 | Paket Kavramı | 91 |
| 5.3.3 | Erişim Düzenleyiciler | 95 |
| 5.3.4 | <code>get/set</code> Yöntemleri | 96 |
| 5.4 | Sınıf/Nesne Değişkenleri ve Yöntemleri..... | 101 |
| 5.4.1 | <code>static</code> anahtar sözcüğü | 103 |
| 5.4.2 | <code>main</code> yöntemi | 110 |
| 5.4.3 | <code>static</code> kod blokları | 111 |
| 5.4.4 | <code>final</code> anahtar sözcüğü ve sabit tanımlama | 114 |

| | | |
|-------|--|-----|
| 5.5 | Kalıtım (<i>Inheritance</i>) | 115 |
| 5.5.1 | Nesneye Yönelik Bir Modelde Sınıflar Arası İlişkiler | 115 |
| 5.5.2 | Java'da Kalıtım | 118 |
| 5.5.3 | Kalıtımla Gelen Nitelik ve Yöntemlere Erişim | 125 |
| 5.5.4 | <code>protected</code> Erişim Düzenleyici | 127 |
| 5.5.5 | Kurucu Zinciri ve <code>super</code> Anahtar Sözcüğü..... | 129 |
| 5.5.6 | Yöntemlerin Geçersiz Kılınması (<i>Method Overriding</i>) | 135 |
| 5.5.7 | Yöntemlerde Geçersiz Kılmanın <code>final</code> ile Engellenmesi | 139 |
| 5.5.8 | <code>Object</code> sınıfı | 139 |
| 5.6 | Çokbiçimlilik (<i>Polymorphism</i>) | 142 |
| 5.6.1 | Ata Sınıf Referansından Alt Sınıf Nesnesine Ulaşma | 142 |
| 5.6.2 | Geç Bağlama (<i>Late Binding</i>) | 143 |
| 5.6.3 | Çokbiçimlilik Nasıl Gerçekleşir? | 144 |
| 5.6.4 | Çokbiçimlilik Ne İşe Yarar? | 149 |
| 5.6.5 | Soyut Sınıflar (<i>Abstract Classes</i>) | 150 |
| 5.6.6 | Arayüzler (<i>Interfaces</i>)..... | 164 |
| 5.7 | Aykırı Durumlar (<i>Exceptions</i>) | 175 |
| 5.7.1 | <code>try - catch</code> bloğu | 178 |
| 5.7.2 | <code>finally</code> deyimi | 183 |
| 5.7.3 | Java'da Aykırı Durumlar | 185 |
| 5.7.4 | Programcının Kodladığı Aykırı Durumlar | 188 |
| 5.7.5 | Aykırı Durumların Yönetilmesi | 191 |
| 5.8 | Girdi/Çıktı (<i>Input/Output - I/O</i>) İşlemleri..... | 197 |
| 5.8.1 | JavaBean Kavramı..... | 198 |
| 5.8.2 | JavaBean Yazma Kuralları | 200 |

| | | |
|-------|---|-------------------------------------|
| 5.8.3 | Serileştirme (<i>Serialization</i>) | 200 |
| 5.8.4 | <code>java.io.Serializable</code> Arayüzü | 202 |
| 5.8.5 | GUI Editor – Property Editor . | Error! Bookmark not defined. |
| 5.8.6 | JavaBean Örneği | Error! Bookmark not defined. |
| 6 | Eksik konular: | Error! Bookmark not defined. |

Önsöz

Bu kitap en azından Yapısal Programlama bilen okuyucular için hazırlanmıştır. Temel programlama bilgileri anlatılmayacak, okuyucunun algoritma kavramını bildiği ve algoritma geliştirebildiği varsayılacaktır.

Amacımız Java programlama dilini ya da Java platformunu anlatmak değil, Nesneye Yönelik Programlama yaklaşımını açıklamak ve bu yaklaşım ile uygulama geliştirmeyi öğretmektir. Programlama konularını öğrenmenin en iyi yolunun programlar yazmak olduğunu düşündüğümüz için, nesneye yönelik programlar yazmak durumundayız ve bunun için bir dil kullanılması gerekiyor. Bu kitap için programlama dili Java olarak seçilmiştir ancak benzer içerik örneğin C++ dili ile de verilebilir.

Kitap içerisinde kullanılacak terimler için genelde Türkiye Bilişim Derneği sözlüğü esas alınacak, kullanılan Türkçe terimlerin İngilizce karşılıkları da parantez içinde vermeye çalışılacaktır. Karşılığı henüz oturmamış ya da kullanıldığı bağlamda istenen anlamı veremediği düşünülen terimler için ise, yaygın kullanılmakta olan terimler tercih edilecek, Türkçe terimler ile karşılanamayan bazı sözcükler/teknolojiler için İngilizce terimler *italik* yazılacaktır.

Giriş