

8. Ders: JAVA Programlama Dili (Veri Tipleri, Tanımlayıcılar, Operatörler)

Fırat Üniversitesi Teknoloji Fakültesi Yazılım Mühendisliği Bölümü

YMH111 Algoritma ve Programlama-I

Dr. Öğr. Üyesi Yaman Akbulut

JAVA Programlama Dili

- <http://www.kriptarium.com/algorithm.html> (Yardımcı kaynak)
- JAVA Programlama Dili (Ders: 11-22) video. (Mutlaka İzleyiniz!)
 - Ders 11: JAVA programlarının derlenmesi ve yorumlanması (izle)
 - Ders 12: JVM, JRE ve JDK arasındaki temel farklar (izle)
 - Ders 13: Java Değişkenler (izle)
 - Ders 14: Değişkenler ile alakalı kod örnekleri (izle)
 - Ders 15: Veri Tipleri (izle)
 - Ders 16: Java İlkel/Primitive Veri Tipleri (izle)
 - Ders 17: Değişken Aralıkları (izle)
 - Ders 18: Java Operatörler (izle)
 - Ders 19: Operatör Örnekleri 1 (x++ ve ++x) (izle)
 - Ders 20: Operatör Örnekleri 2 (izle)
 - Ders 21: Mantıksal ve Aritmetiksel Operatörler (izle)
 - Ders 22: Kaydırma Operatörleri (izle)

JAVA'da program yazma ve çalıştırma ortamı

1. Java Development Kit (JDK) (Java geliştirme kiti (seti))
+ Windows notepad (yazı editörü) (alternatif [notepad++](#))

2. Integrated Development Environment (IDE)

Entegre geliştirme ortamı

- NetBeans
- Eclipse
- IntelliJ IDEA

Özel Karakterler

{ } braces

() parentheses

[] brackets

// double slashes

" " quotation marks

; semicolon

/*

for block comments

*/

/**

for documentation

*/

Java Keywords /Reserved Words

abstract	double	int	super
assert	else	interface	switch
boolean	enum	long	synchronized
break	extends	native	this
byte	final	new	throw
case	finally	package	throws
catch	float	private	transient
char	for	protected	try
class	goto	public	void
const	if	return	volatile
continue	implements	short	while
default	import	static	
do	instanceof	strictfp*	

Java Veri Tipleri

Primitive data types:

- Boolean
- Character
- Integer
- Floating point

Non-Primitive data types:

- Classes
- Interfaces
- Arrays

Numerik Veri Tipleri

tipi	kapasitesi	değer aralığı
byte	8-bit işaretli tamsayı	$-2^7 \dots 2^7-1$
short	16-bit işaretli tamsayı	$-2^{15} \dots 2^{15}-1$
int	32-bit işaretli tamsayı	$-2^{31} \dots 2^{31}-1$
long	64-bit işaretli tamsayı	$-2^{63} \dots 2^{63}-1$

float 32-bit sayı ondalıklı sayı (kesir) 32-bit IEEE 754
Negatif aralık: $-3.4028235E + 38$ den $-1.4E - 45$
Pozitif aralık: $1.4E - 45$ ten $3.4028235E + 38$

double 64-bit sayı ondalıklı sayı (kesir) 64-bit IEEE 754
Negatif aralık: $-1.7976931348623157E + 308$ ten $-4.9E - 324$
Pozitif aralık : $4.9E - 324$ ten $1.7976931348623157E + 308$

Klavyeden numerik değer okuma

tipi	metot	açıklama
• byte	nextByte()	byte tipinde tamsayı okur.
• short	nextShort()	short tipinde tamsayı okur.
• int	nextInt()	int tipinde tamsayı okur.
• long	nextLong()	long tipinde tamsayı okur.
• float	nextFloat()	float tipinde ondalıklı sayı okur.
• double	nextDouble()	double tipinde ondalıklı sayı okur.

Örnek 1: SayiOkuma.java

```
1  //Nümerik Sayı Okuma
2  //
3  import java.util.Scanner;
4
5  public class SayiOkuma {
6      public static void main(String [] args) {
7          Scanner giriş = new Scanner(System.in);
8          System.out.print("Bir byte değeri giriniz:");
9          byte byteDeger = input.nextByte();
10
11          System.out.print("Bir short değeri giriniz:");
12          short shortDeger = input.nextShort();
13
14          System.out.print("Bir int değeri giriniz:");
15          int intDeger = input.nextInt();
16
17          System.out.print("Bir long değeri giriniz:");
18          long longDeger = input.nextLong();
19
20          System.out.print("Bir float değeri giriniz:");
21          float floatDeger = input.nextFloat();
22
23          System.out.print("Bir double değeri giriniz:");
24          double doubleDeger = input.nextDouble();
25      }
26 }
```

Tanımlayıcılar

Tanımlayıcılar

Classes (Sınıflar),
Methods (Yöntemler),
ve Variables (Değişkenleri) tanımlar.

Bir tanımlayıcı Harfler, rakamlar, _ alt çizgi ve \$ işaretlerinden oluşabilir.

Bir tanımlayıcı rakam ile başlayamaz.

Bir tanımlayıcı keywords'den oluşamaz.

Bir tanımlayıcı true, false veya null olamaz.

Bir tanımlayıcı istenen uzunlukta olabilir.

Tanımlayıcılar

Java büyük küçük harfe duyarlıdır.

yaman, Yaman, YAMAN farklı tanımlayıcıdır.

Açıklayıcı isimlendirme

Tanımlayıcılar program içindeki değişkenlerin, metotların, sınıfların ve diğer şeylerin isimlendirilmesi için kullanılır.

İsimlendirmenin açıklayıcı olması program okunabilirliği açısından çok önemlidir. `ogrenciSayisi`; `ogrSay`, `ogrSayi`, `ogrenciSayi` dan daha iyi bir isimlendirmedir.

İsimlendirmelerde kısaltmalardan kaçınınız.

`i`, `j`, `k`, `x`, `y` benzeri değişken isimlendirmeleri genelleme açısından uygundur. Bunları çoğunlukla indis olarak kullanıyoruz.

\$ karakterini isimlendirmede kullanmayınız.

Tanımlayıcıları isimlendirme

Değişkenlerin ve yöntemlerin (metot) isimlendirilmesi

Değişken (variable) ve yöntemleri (method) isimlendirmeye küçük harf ile başlayın, küçük devam edin.

`yaricap` değişken, `alan` değişken, `print` yöntem

Sınıfların isimlendirilmesi

Sınıfların (class) ilk harflerini Büyük yapın.

`AlanHesapla` sınıf, `System` sınıf

`System` sınıfı Java'da tanımlı olduğundan kendi sınıflarınıza bu ismi vermemelisiniz!

Sabitlerin isimlendirilmesi

Sabitlerin (constant) bütün harflerini BÜYÜK yapın.

`PI` sabit, `ENBUYUK` sabit

Değişken tanımlama

Veri tipleri: `int` `double` `byte` `short` `long` `float` `char` `boolean`

Örnek tanımlamalar

```
veritipi degisken1, degisken2, ..., degiskenn;
```

```
int sayi;  
double yaricap;  
double donusumOrani;
```

```
int i, j, k;
```

```
int sayi = 1;
```

```
int sayi;  
sayi = 1;
```

```
int i = 1, j = 2;
```

Atama Operatörü (=)

Örnek atamalar

```
int y = 1;  
double yarıcap = 2.0;  
int x = 5 * (3 / 2);  
x = y + 1;  
double alan = yarıcap * yarıcap * 3.14159;
```

```
x = x + 1;
```

```
1 = x; //yanlış kullanım
```

degisken = ifade şeklinde kullanılır.

$x = 2 * x + 1$

matematikte bir denklem

$x = 2 * x + 1$

Java'da $2 * x + 1$ ifadesi x değişkenine atanır.

Sabit Tanımlama (final)

LISTING 2.4 ComputeAreaWithConstant.java

```
1  import java.util.Scanner; // Scanner is in the java.util package
2
3  public class ComputeAreaWithConstant {
4      public static void main(String[] args) {
5          final double PI = 3.14159; // Declare a constant
6
7          // Create a Scanner object
8          Scanner input = new Scanner(System.in);
9
10         // Prompt the user to enter a radius
11         System.out.print("Enter a number for radius: ");
12         double radius = input.nextDouble();
13
14         // Compute area
15         double area = radius * radius * PI;
16
17         // Display result
18         System.out.println("The area for the circle of radius " +
19             radius + " is " + area);
20     }
21 }
```

Aritmetik Operatörler

	anlamı	örnek	sonuç
+	toplama	$22 + 1$	23
-	çıkarma	$22.0 - 0.1$	21.9
*	çarpma	$200 * 30$	6000
/	bölme	$1.0 / 2.0$	0.5
%	mod (kalan)	$20 \% 3$	2

Hem bölünen hem de
bölen `int` ise:

5 / 2 sonuç 2 olur

5 / 2 sonuç 2.5 olmaz

% Mod işlemi kalanı verir:

5 % 2 sonuç 1 olur

10 % 2 sonuç 0 olur

20 % 3 sonuç 2 olur

Artı (+) Operatörü

```
ArtiOperatoru.java
1  //Artı (+) operatörü
2
3  public class ArtiOperatoru {
4      public static void main(String [] args) {
5          byte top;
6          top = 3 + 5;
7          System.out.println("3+5=" + top);
8          System.out.println("Yazilim Muhedisligi Bolumu");
9          System.out.println("Yazilim " + "Muhedisligi " + "Bolumu");
10     }
11 }
```

```
C:\Program Files\Java\jdk-15.0.1\bin\yeni>javac ArtiOperatoru.java
```

```
C:\Program Files\Java\jdk-15.0.1\bin\yeni>java ArtiOperatoru
```

```
3+5=8
```

```
Yazilim Muhedisligi Bolumu
```

```
Yazilim Muhedisligi Bolumu
```

Ödev 1:

```
OrtalamaHesapla.java
1 //Ortalama Hesapla
2 import java.util.Scanner; //Scanner, java.util paketinde
3
4 public class OrtalamaHesapla {
5     public static void main(String [] args) {
6         //Scanner nesnesi oluşturur
7         Scanner input = new Scanner(System.in);
8
9         //Kullanıcıdan 3 sayı girmesi istenir
10        System.out.print("Uc sayi giriniz:");
11        double sayi1 = input.nextDouble();
12        double sayi2 = input.nextDouble();
13        double sayi3 = input.nextDouble();
14
15        //Ortalama hesaplanır
16        double ortalama = (sayi1 + sayi2 + sayi3) / 3;
17
18        //Sonuç gösterilir
19        System.out.println(sayi1 + " " + sayi2 + " "
20            + sayi3 + " " + "Ortalaması: " + ortalama);
21    }
22 }
```

Ödev 2: `KardesYasi.java`

İki kardeşe ait yaş bilgisinin kullanıcı tarafından klavyeden girilmesi istenmiştir ve sonrasında kardeşlerin yaşlarının toplamaları ve ortalaması hesaplanıp, sonucu çıktı alınmıştır.

Bu kodu Java programlama dilinde yazınız.

Ödev 3:

Aşağıdaki algoritmayı Java koduna dönüştürünüz (yazınız)

Adım 1: Başlangıç değeri 100 olan `mil` isimli bir `double` değişken tanımlayınız.

Adım 2: Değeri 1.609 olan `KILOMETRE_MIL_KATSAYISI` isimli bir `double` sabit tanımlayınız.

Adım 3: `kilometre` isimli bir `double` değişken tanımlayınız, `mil` ve `KILOMETRE_MIL_KATSAYISI` nı çarpınız ve sonucu `kilometre`ye atayınız.

Adım 4: `kilometre`yi konsolda gösteriniz.

Adım 4'ten sonra `kilometre` nedir?

Ödev 4:

Aşağıdaki kodu bir kaynak dosyasına yazınız. Çalışıp değer döndürebilmesi için ana metodu ve diğer gereken kodları ekleyiniz. Dosyanın ismi Kuvvet.java olsun.

`pow` metodu `Math` sınıfında tanımlanmıştır. Syntax (sözdizimi) `Math.pow(a, b)` şeklindedir. `a` üssü `b`'nin sonucunu geri döndürür. Örnek olarak `Math.pow(2, 3)` ifadesinde a^b (2^3) sonuç 8 olarak gelir, burada `a` ve `b`; `pow` metodunun parametresidir.

```
System.out.println(Math.pow(2, 3)); // Displays 8.0
System.out.println(Math.pow(4, 0.5)); // Displays 2.0
System.out.println(Math.pow(2.5, 2)); // Displays 6.25
System.out.println(Math.pow(2.5, -2)); // Displays 0.16
```

Kaynak: Introduction to JAVA Programming 10th Edition, Y. Daniel Liang

Ödev 5:

Aşağıdaki kod satırlarının bir dosyaya yazıp çalıştırınız. Dosyanın adı NoktaliSayi.java olsun.

Birinci satırın sonucu ile ikinci satırın sonucunu inceleyiniz, karşılaştırınız.

```
System.out.println("1.0 / 3.0 = " + 1.0 / 3.0);
```

```
System.out.println("1.0F / 3.0F = " + 1.0F / 3.0F);
```

Ödev 6:

Java ve Bilişim terimleri olarak

multiprogramming, multithreading, multiprocessing,

runtime error, logic error, compile error,

pseudocode, variable, console,

identifier, suffix, prefix,

nedir?

İngilizce ve Türkçe karşılıklarını yazınız.