

2017-2018 Bahar Yarıyılı
Balıkesir Üniversitesi
Endüstri Mühendisliği Bölümü

EMT2226

Nesne Yönelimli Programlamaya Giriş

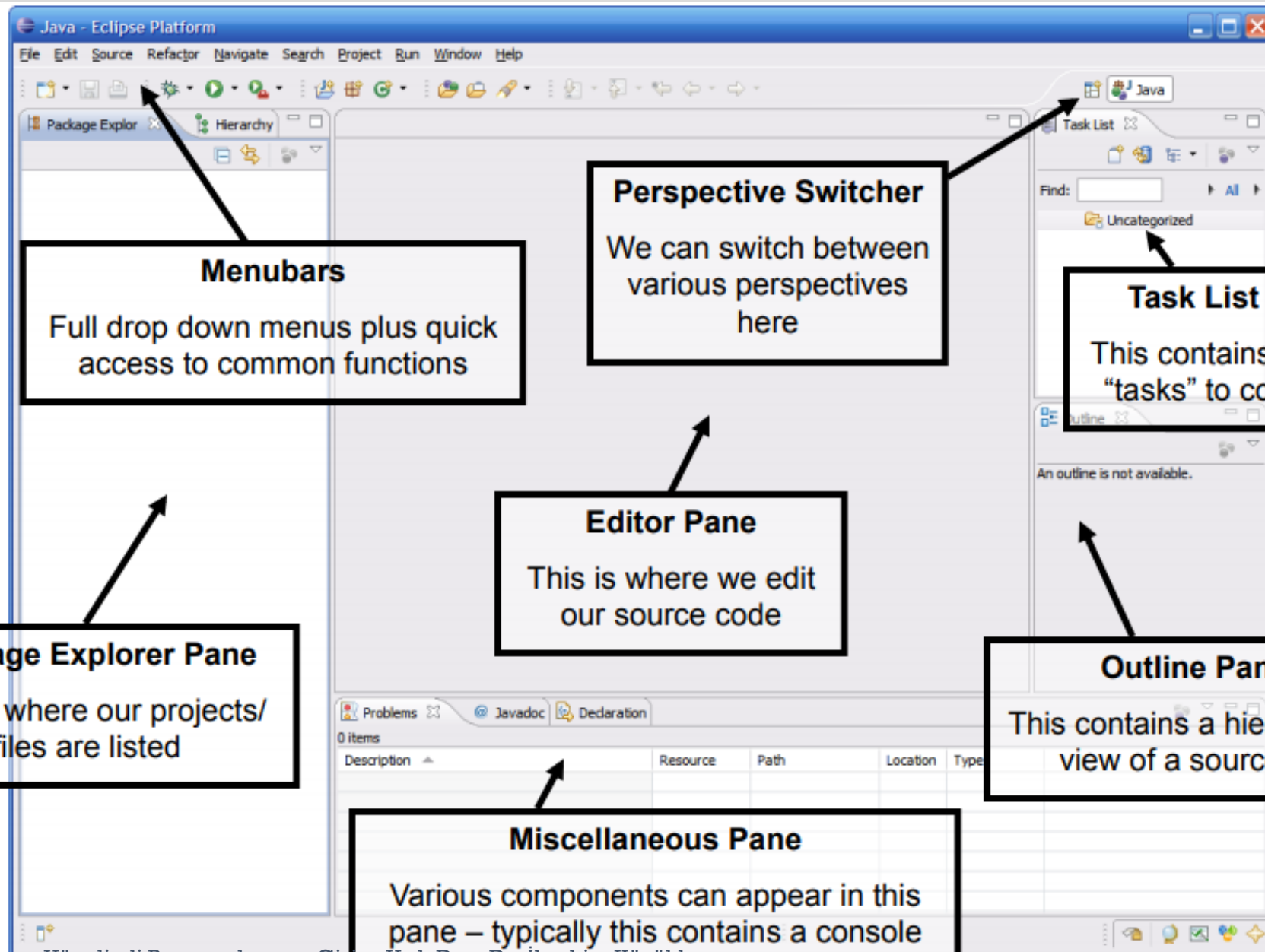
Yrd. Doç. Dr. İbrahim Küçükkoç
Web: ikucukkoc.baun.edu.tr
Email: ikucukkoc@balikesir.edu.tr

ECLIPSE IDE FOR JAVA

The screenshot displays the Eclipse IDE interface. On the left, the Package Explorer shows a project named 'FarLands' with a sub-package 'com.shc.silenceengine.core' containing several Java files, including 'Display.java'. The main editor window shows the code for 'Display.java', which is responsible for creating an OpenGL window. The code includes conditional logic for different operating systems (macOS vs. others) to set appropriate window hints. It also checks for development mode to enable debugging. The window is created using 'glfwCreateWindow' and the OpenGL context is initialized with 'GLFW_OPENGL_FORWARD_COMPAT', 'GLFW_OPENGL_PROFILE', and 'GLFW_OPENGL_CORE_PROFILE'. The code concludes by creating the GL context and setting a batcher.

```
148 if (System.getProperty("os.name").toLowerCase().contains("mac"))
149 {
150     glfwWindowHint(GLFW_CONTEXT_VERSION_MAJOR, 3);
151     glfwWindowHint(GLFW_CONTEXT_VERSION_MINOR, 2);
152 }
153 else
154 {
155     glfwWindowHint(GLFW_CONTEXT_VERSION_MAJOR, 3);
156     glfwWindowHint(GLFW_CONTEXT_VERSION_MINOR, 3);
157 }
158 glfwWindowHint(GLFW_OPENGL_FORWARD_COMPAT, GL_TRUE);
159 glfwWindowHint(GLFW_OPENGL_PROFILE, GLFW_OPENGL_CORE_PROFILE);
160 glfwWindowHint(GLFW_VISIBLE, visible ? GL_TRUE : GL_FALSE);
161 glfwWindowHint(GLFW_RESIZABLE, resizable ? GL_TRUE : GL_FALSE);
162
163 if (Game.development)
164 {
165     System.setProperty("org.lwjgl.util.Debug", "true");
166     glfwWindowHint(GLFW_OPENGL_DEBUG_CONTEXT, GL_TRUE);
167 }
168
169 // Create the window
170 long window = glfwCreateWindow(width, height, title, monitor, parent);
171
172 if (window == NULL)
173     throw new SilenceException("Error creating Display. Your system is unsupported.");
174
175 // Take care of OpenGL context
176 glfwMakeContextCurrent(window);
177 glfwSwapInterval(vSync ? 1 : 0);
178
179 GLContext.createFromCurrent();
180
181 Game.setBatcher(new Batchter());
```

Console: No consoles to display at this time.



JAVA

- Java, ilk olarak Sun Microsystems tarafından geliştirilen nesne yönelimli programlama dilidir.
- 1995 yılında piyasaya sürülmüştür. James Gosling, Haziran 1991’de “Oak” adıyla geliştirmiştir.
- Platform bağımsızlığı (programı bir kere yaz, derle ve her platformda çalıştır), tasarım amaçlarındandır. Birçok donanım, işletim sistemi ve platformda çalışabilmektedir.
- En popüler dildir.
- Bazı dillerden (Python vb.) daha karmaşıktır.
- Bazı dillerden (C++ vb.) daha kolaydır.

JVM

- JVM (Java Virtual Machine) veya Java Sanal Makinesi: Java’nın özgün sanal makine modelini gerçekleştiren bilgisayar programları ve yapılarından oluşan bir bütündür.
- “Java Bytecode” adı verilen, bilgisayar ara dilini kabul eder. Bir programın bytecode’u, Java dili derleyicileri tarafından üretilir.

ILK PROGRAM

- Prog1 adlı java programı :

```
class Prog1
{
    public static void main(String args[])
    {
        System.out.println("Merhaba");
    }
}
```

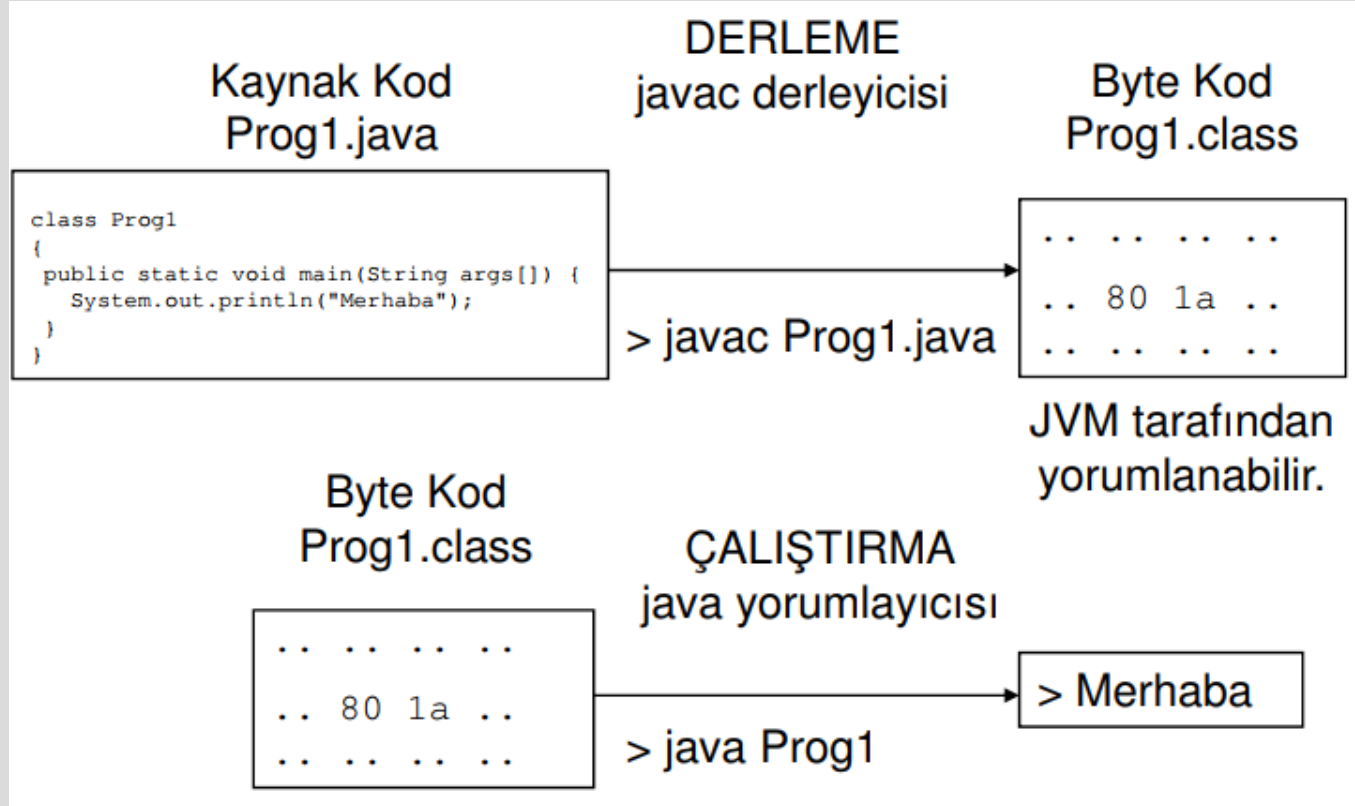
ekrana merhaba yazdırmaktadır.

Program Yapısı

```
class CLASSNAME {
    public static void main(String[] arguments) {
        STATEMENTS
    }
}
```

ILK PROGRAM

- JVM'i (Java Virtual Machine) açmak üzere Java komutu kullanıldığında işletilen bilgisayar programına Java uygulaması (application) denilir.



CIKTI

```
System.out.println(some String)
```

- Örnek:

```
System.out.println("output");
```

Konsola yazdırır

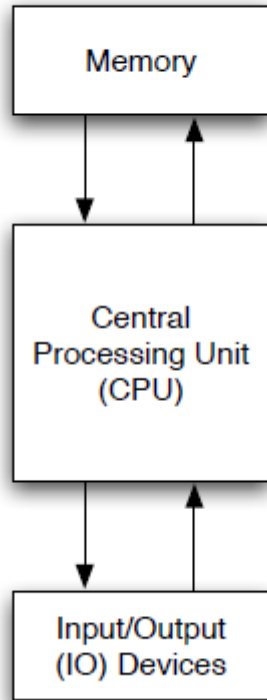
Konsola 'Output' yazdırır

- `System.out.print()` metodunda yazı yazıldıktan sonra aynı satırda kalınır.
- `System.out.println()` metodunda ise bilgi yazıldıktan sonra alt satıra geçer.

- Java programlarında // Satırın geri kalan kısmının açıklama satırı olduğunu gösterir.
- Geleneksel /* Açıklama */ işaretleri de özellikle çok satıra yayılan açıklamalarda kullanılır.
- Deyimler “;” ile sonlandırılmalıdır.
- Her java programı en az bir sınıf tanımı içermelidir.

```
// Bu bir yorumdur  
/* Bu da bir yorumdur */  
/* Bu da  
çok satırlı  
yorumdur */
```


The Computer



CPU Instructions

$Z = x + y$

Read location x

Read location y

Add

Write to location z

Programlama Dilleri

CPU komutlarından daha kolay anlaşılabilir


CPU tarafından anlaşılabilmesi için dönüştürülmesi gerekir

İKİNCİ PROGRAM

```
class Hello2 {  
    public static void main(String[] arguments) {  
        System.out.println("Hello world."); // Print once  
        System.out.println("Line number 2"); // Again!  
    }  
}
```

TIPLER

- Depolanabilen veya manipüle edilebilen değer türleridir.

Type	Size in bits	Values
boolean		true or false
byte	8	-128 to +127
char	16	Unicode
short	16	integer
int	32	integer
long	64	integer
float	32	float
double	64	double

TIPLER

- Tam sayıların sınırları

Type	Minimum Value	Maximum Value	Size in Bytes
byte	-128	127	1
short	-32,768	32,767	2
int	-2,147,483,648	2,147,483,647	4
long	-9,223,372,036,854,775,808	9,223,372,036,854,775,807	8

- Ondalıklı sayıların sınırları

Type	Minimum	Maximum	Size in Bytes
Float	$-3.4 * 10^{38}$	$3.4 * 10^{38}$	4
Double	$-1.7 * 10^{308}$	$1.7 * 10^{308}$	8

DEGISKENLER

- Belirli bir tipteki deęeri tutmak için tanımlanmış konum.

- Örnek: **String** **araba**

Tip

Değişken

Adınızı ve Soyadınızı,
değişkenler
kullanarak nasıl
yazdırırsınız?

Değer Atama

Use = to give variables a value.

Example:

```
String foo;  
foo = "IAP 6.092";
```

- Değişkenlere anlamlı isim verin, h ve s yerine hiz ve sayi gibi isimlendirmeler kullanın.
- Bir değişken tanımladığınızda ismini ve tipini verin.

int sayi, eggsPerBasket;

double yaricap;

- Bir değişkenin tipi onun tutacağı değer türünü belirler (int→tamsayı, double → noktalı sayı, char → karakter, vb.).
- Bir değişken kullanılmadan önce tanımlanmalıdır.

DEĞİSKEN

- Değişkenler harf, yazı, sayı gibi verileri tutar.
- Değişkenleri veriyi depolamak için yerler olarak düşünün.
- Hafıza yerleri olarak uygulanır.
- Bir değişken ile depolanan veri onun değeri olarak çağrılır.
- Değer hafızada saklanır.
- Bir değişkenin değeri değiştirilebilir.

DEĞER ATAMA

Can be combined with a variable declaration.

Example:

```
double badPi = 3.14;  
boolean isJanuary = true;
```

```
class Hello3 {  
    public static void main(String[] arguments) {  
        String foo = "IAP 6.092";  
        System.out.println(foo);  
        foo = "Something else";  
        System.out.println(foo);  
    }  
}
```



```

public class ComputeArea{
    /** Main method */
    public static void main(String[] args) {
        double yaricap;
        double alan;
        // yarıçapa değer ata
        yaricap= 20;
        // Alanı hesapla
        alan= yaricap* yaricap* 3.14159;
        // Sonuçları göster
        System.out.println("Yarıçapı " + yaricap + " olan çemberin alanı: " + alan);
    }
}

```

Değişken tanımlandı

Değer atandı

yaricap: #Değer Yok

alan: #Değer Yok

yaricap: 20

alan: 1256.636

▪ public class ComputeArea{

/** Main method */

public static void main(String[] args) {

double yarıcap;

double alan;

// yarıçapa değer ata

yarıcap= 20;

// Alanı hesapla

alan= yarıcap* yarıcap* 3.14159;

// Sonuçları göster

System.out.println("Yarıçapı " + yarıcap + " olan çemberin alanı: " + alan);

}

}

Değişken tanımlandı

Değer atandı

Konsola Yazılan:

Yarıçapı 20 olan çemberin alanı: 1256.636

yarıcap: #Değer Yok

alan: #Değer Yok

yarıcap: 20

alan: 1256.636

- Değişkenler
 - yarıcap
 - alan
- Atanan değerler
 - yarıcap= 20;
 - alan= yarıcap* yarıcap* 3.14159;

DEGISKENIN DEGERINI GUNCELLEME

```
int passengers;  
passengers = 0;  
passengers = passengers + 5;
```



Update the variable `passengers` to subtract 1 and add 5

```
int passengers;  
passengers = 0;  
passengers = passengers + 5;  
passengers = passengers - 3;
```

- `passengers = passengers - 1 + 5;`
- `passengers = 5 - 1;`
- `passengers = passengers - 5 + 1;`
- `passengers = 5;`
`passengers = passengers - 1;`



- Söz dizimi

***tip** degsikenadi_1, degiskenadi_2, ...;*

- Örnekler

int sayi, toplam;

double mesafe, ortalama;

char cinsiyet;

- Bir sınıf tipi, nesnelerin bir sınıfı için kullanılır ve hem veri hem de metot içerir.

“Bugün hava güzel” String türünden sınıfın bir değeridir.

- İlkel bir tür, sayı ve karakter türünden değer tutar.

int, **double**, ve **char** ilkel türlerdir.

İSİMLENDİRME KURALLARI

- Sınıf türleri büyük harf ile başlar (örn. **String**).
- İlkel türler küçük harf ile başlar (örn. **int**, **char**, **double**, **byte**).

Değişkenler nerede tanımlanmalıdır?

- Değişkeni kullanmadan önce veya
- { ile başlayıp } ile biten program bloğunuzun başında tanımlamalısınız.

```
public static void main(String[] args){  
    /* değişkenleri burada tanımlayınız*/  
}
```

- Java Dilinde değişkenlerin kullanılmadan önce değerleri olması gerekir.
- Değişkenler sadece tanımlandıkları blokta kullanılabilirler.

DEĞİŞKEN İSİMLERİ

- Değişken isimleri sadece aşağıdakilerden oluşmalıdır.
 - Harfler (Türkçe karakter kullanmayın, İ, ö, ç, ü, ğ, ş)
 - Rakamlar (0'dan 9'a kadar)
 - Altçizgi (_)
- Fakat ilk karakter rakam olmamalıdır.
- Değişken isimleri boşluk, nokta(.), yıldız(*), veya diğer özel karakterleri içermez:
Ornek: 7-11 netscape.com util.* (izin verilmez)
- Değişken isimleri gereksiz şekilde uzun olmamalıdır.
- Java büyük-küçük harf duyarlı olduğundan **sayi**, **Sayi**, ve **SAYI** değişkenleri farklı olarak algılanır.
- Java dilinde kullanılan anahtar kelimeler değişken adı olarak kullanılamazlar.
- Örneğin bir değişkene **print**, **int**, **publicclass** gibi isimler verilemez.

İLKEK (PRIMITIVE) TÜRLER

- Tam sayı türü (**byte**, **short**, **int** ve **long**)

- int en çok kullanılandır.

- Noktalı sayı (**float** ve **double**)

- double en kullanılandır.

- Karakter türü (**char**)

- Mantıksal veri türü (**boolean**)

- integer türü

0 -1 365 12000

- Noktalı sayı türü

0.99 -22.8 3.14159 5.0

- Karakter türü

'a' 'A' '#' ''

- boolean türü

true false

BASIT OPERATORLER

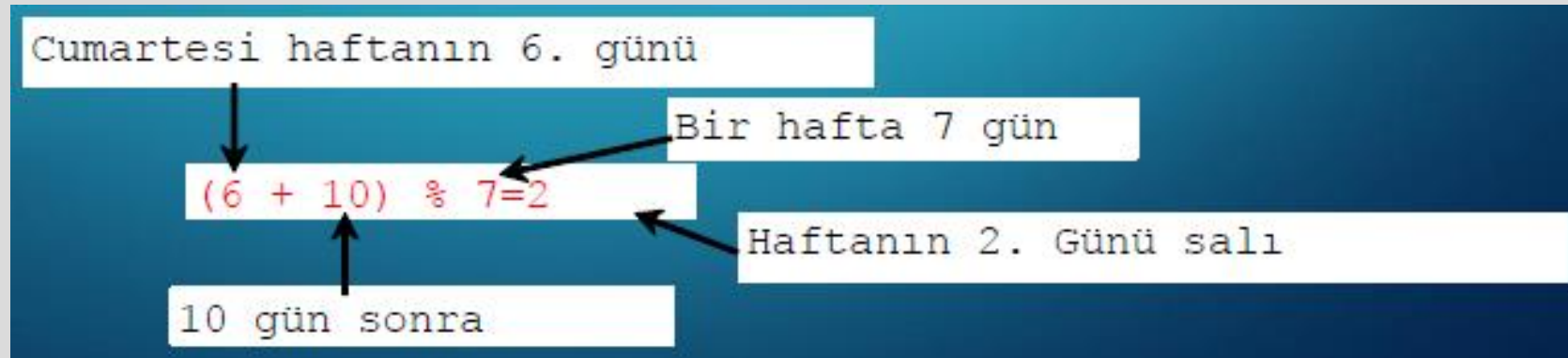
Operator	Sembol	Ornek
Atama	=	A = 5 (Sonuc: A=5)
Ekleme	+	3+1 (Sonuc: 4)
Cıkarma	-	3-1 (Sonuc: 2)
Carpma	*	3*3 (Sonuc: 9)
Bolme	/	6.0 / 3.0 (Sonuc: 3.0)
Mod	%	20%3 (Sonuc: 2)

TAMSAYI BOLME

- +, -, *, /, ve %
- $5/2$ ifadesinin sonucu 2 dir.
- $5.0/2$ ifadesinin sonucu 2.5 dir.
- $5\%2$ sonucu 1'dir (bölme sonucu kalan değer)

KALANLI BÖLME

- Kalanlı bölme programlamada oldukça kullanışlıdır. Örneğin, bir çift sayının ikiye bölümünden kalan daima 0 iken, tek sayının bölümünden kalan her zaman 1'dir. Bu bilgiyi sayının tek mi çift mi olduğunu bulmak için kullanabilirsiniz.
- Bugün Cumartesi olduğunu düşünelim ve arkadaşınız 10 gün sonra size gelecek. 10 gün sonra hangi gündür? Aşağıdaki ifade ile bunun Salı olduğunu bulabilirsiniz.



ARITMETİK İFADELER

Matematiksel ifade	Javada gösterim	Javada parantezli gösterim
$rate^2 + delta$	<code>rate*rate + delta</code>	<code>(rate*rate) + delta</code>
$2(salary + bonus)$	<code>2*(salary + bonus)</code>	<code>2*(salary + bonus)</code>
$\frac{1}{time + 3\ mass}$	<code>1/(time + 3*mass)</code>	<code>1/(time + (3*mass))</code>
$\frac{a - 7}{t + 9v}$	<code>(a - 7)/(t + 9*v)</code>	<code>(a - 7)/(t + (9*v))</code>

KISAYOLDAN ATAMA OPERATÖRÜ

<i>Operator</i>	<i>Örnek</i>	<i>Eşdeğeri</i>	
<code>+=</code>	<code>i += 8</code>	<code>i = i + 8</code>	
<code>-=</code>	<code>f -= 8.0</code>	<code>f = f - 8.0</code>	
<code>*=</code>	<code>i *= 8</code>	<code>i = i * 8</code>	
<code>/=</code>	<code>i /= 8</code>	<code>i = i / 8</code>	
<code>%=</code>	<code>i %= 8</code>	<code>i = i % 8</code>	

NÜMERİK VERİ DÖNÜŞÜMÜ

- Aşağıdaki atamaları göz önüne alalım:

```
byte i=100;
```

```
long k=i*3+4;
```

```
double d=i*3.1+k/2;
```

- İki farklı tür içeren bir denklemde, java aşağıdaki kurallara göre dönüşümü otomatik yapar:
 - 1.Değişkenlerden biri **double** ise, diğeri **double**'a dönüştürülür.
 - 2.Değilse, eğer biri **float** ise diğeri **float**'a dönüştürülür.
 - 3.Değilse, değişkenlerden biri **long** ise diğeri **long**'a dönüştürülür.
 - 4.Değilse, her iki değişken **int**'e dönüştürülür.

TÜR DÖNÜŞÜMÜ

- Anlaşılır dönüştürme

double d=3; (tip genişlemesi)

- Belirtilmiş dönüştürme

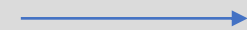
int i = (int) 3.0; (tip daralması)

int i = (int) 3.9; (Kesirli kısım atılır)

- Yanlış olan nedir?

int x=5/2.0;

byte, short, int, long, float, double



Aralık artar

Variable Arithmetic

(1) Addition

int x = 2+3; = 5

(2) Subtraction

int y = 4-5; = -1

(3) Multiplication

int temp = -20; = -20

(4) Division

int days = 7*4; = 28

double div = 5/2; = ~~2.5~~ = 2

double accurate = 5/2.0; = 2.5

double div2 = 24/5; = ~~4.8~~ = 4

double accurate2 = 24/5.0; = 4.8

NÜMERİK VE KARAKTER TÜRLERİ ARASINDA DÖNÜŞÜM

- `int i = 'a'; // int i = (int) 'a' ile aynı;`
- `char c = 97; // c = (char) 97 ile aynı;`

BASİT EKKRAN GİRİŞ/ÇIKIŞ FONKSİYONU

```
int a=5; double  
b=2.5; boolean  
c=true;  
  
char d='?';  
  
System.out.println(a);  
System.out.println(b);  
System.out.println(c);  
System.out.println(d);
```

Çıktı:
5
2.5
True
?

```
int a=5, b=17, c=4;  
System.out.println(a);  
System.out.println(b);  
System.out.print(«a=»+a);  
System.out.println(«, a+b= »+a+b);  
System.out.println(«a+b=»+(a+b));  
System.out.print(a+ « »+ b+ « »);  
System.out.print(a+ « »+ b+ « »);  
System.out.println(a+b+c);  
a=b-c;  
System.out.println(«a=»+a+ «, b=»+b+ «, c= »+c);
```

Çıktı:
5
17
a=5, a+b=517
a+b=22
517 5 17 26
a=13, b=17, c=4

Özel Karakterler

- \" Çift tırnak
- \' tek tırnak
- \\ ters eğik çizgi
- \n Yeni satır. Sonraki satırın başına gider.
- \r Satırbaşı. Geçerli satırın başına gider
- \t Tabtuşu. Sonraki tab sekmesine kadar imleci götürür.

Klavyeden Veri Okunması: Scanner (Tarayıcı) Sınıfı

- Scanner sınıfı `java.util` kütüphanesinin içinde yer alır ve kullanılması için aşağıdaki iki satırdan birinin programın başına yazılması gerekir.

```
import java.util.*;
```

```
import java.util.Scanner;
```

- İlk satır `java.util` kütüphanesini eklerken ikinci satır yalnızca Scanner uygulamasının paketin eklemek için kullanılır.
- Klavyeden veri girişini sağlayan `Scanner` sınıfının bir nesnesi aşağıdaki gibi oluşturulur.

```
Scanner klavye= new Scanner(System.in);
```

```
int a=klavye.nextInt();
```

Bazi Scanner Sinifi Örnekleri

- `.nextByte()`: Klavyeden byte türünde bir sayı okur.
- `.nextShort()`: Klavyeden Short türünde bir sayı okur.
- `.nextInt()`: Klavyeden int türünde bir sayı okur.
- `.nextLong()`: Klavyeden long türünde bir sayı okur.
- `.nextDouble()`: Klavyeden double türünde bir sayı okur.
- `.nextFloat()`: Klavyeden Float türünde bir sayı okur.
- `.next()`: Klavyeden String türünden bir sözcük okur. Okunan sözcük imleç ile bir sonraki boşluk arasında kalan parçadır.
- `.nextLine()`: Klavyeden String türünde bir satır okur. Satır sonu karakteri okunan satıra dahil değildir.

```
int sayi= keyboard.nextInt();  
double mesafe= keyboard.nextDouble();  
String kelime= keyboard.next();  
String butunsatir= keyboard.nextLine();
```

Giriş için kullanıcıya bilgi vermeyi unutmayın

```
System.out.print("Bir sayı gir: ");
```

Klavyeden Bir Kişinin Adını, Soyadını, ve Yaş ve Boyunu Okuyup Ekranına Basan Bir Java Programı Yazalım

```
import java.util.*;
public class klavyegirisorneği{
public static void main() {
    int yas;
    double boy;
    String adsoyad;
    Scanner klavye=new Scanner(System.in);
    System.out.println("Adınızı ve soyadınız aralarında boşluk ile girin");
    adsoyad=klavye.nextLine();
    System.out.println("yaşınızı giriniz:");
    yas=klavye.nextInt();
    System.out.println("Boy bilginizi girin");
    boy=klavye.nextDouble();
    System.out.println("Adınız: "+adsoyad+" Yaş: "+yas+" Boy: "+boy);
}
```


İki Sayı Okuyup Toplamını Bulan Java Programını Yazalım

```
import java.util.*  
public class Toplama {  
    public static void main( String [] args) {  
        System.out.print("ilk sayı:\t");  
        Scanner klavye=new Scanner(System.in);  
        int ilk=klavye.nextInt();  
        System.out.println();  
        System.out.print("ikinci sayı:\t");  
        int ikinci=klavye.nextInt();  
        int toplam=ilk+ikinci;  
        System.out.println(ilk+" "+ikinci+" "+"="+toplam);  
    }  
}
```

Çıktı:

İlk Sayı: 5

İkinci sayı:8

5+8=13

String Sinifi

- `String` bir karakter dizisidir.
- `" "` karakterleri arasına yazılan her türlü cümle bir dizidir.
- `String` java'nın temel türlerinden biri değildir ve sınıf olarak ayrıca tasarlanmıştır.
- Sınıf olarak tasarlanması kendine özgü metot ve özellikleri olduğu anlamına gelir.
- Java'da aşağıdaki örneklerdeki gibi çift tırnak içinde gösterilirler.
 - `"Merhaba Java"`
 - `"Merhaba"`
 - `" "`
 - `"Balıkesir Endüstri Bölümü"`
- `String iletı=" Balıkesir Üniversitesi Endüstri Mühendisliği";`
- `System.out.println(iletı);`
- Dizgi değişkenleri doğrudan `System.out.println` komutuyla da ekrana yazdırılabilir.
- `System.out.println("Balıkesir Üniversitesi Endüstri Mühendisliği");`

Dizgilerin Birbirine Eklenmesi

- İki dizgi birbirine + işleci kullanılarak eklenir. İki kısa dizgi daha uzun bir dizgi oluşturmak için birleştirilebilir.

```
String ilk="Balikesir";  
String ikinci="Endustri Mühendisliği"  
String Balikesir=ilk+ "Üniversitesi"+ikinci;
```

- Balikesir dizgisi "Balikesir Üniversitesi Endustri Mühendisliği" şeklinde olur.

- İki dizgiyi aralarında boşluk bırakarak birleştirme

```
String ilk="Balikesir";  
String iki="Üniversitesi"  
String üniversitem=ilk+ " "+iki;
```

- Java'da string ile basit veri türleri de birleştirilebilir.

- Bu durumda yeni tür String olur.

```
String tümce=" Balikesir Üniversitesi "+ 1975+ "yılında kuruldu";
```

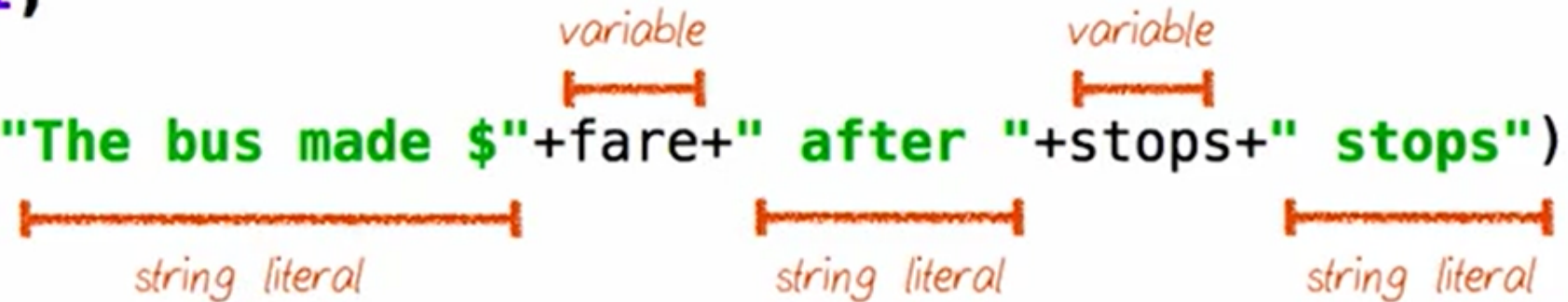
String Concatenation

```
int stops;  
int fare;  
stops = 0;  
fare = 0;  
stops = stops + 1;  
fare = fare + 5;  
stops = stops + 1;  
fare = fare + 3;  
stops = stops + 1;  
fare = fare + 7;  
System.out.println("The bus made $" + fare + " after " + stops + " stops");
```



Print output:

The bus made \$15 after 3 stops



Dizgi Türünde Verilerin Uzunluğunu Bulma

Dizgilerin uzunluğunu bulmak için `length()` adlı metot kullanılır.

```
int uzunluk= "Balikesir Endustri".length();  
System.out.println(uzunluk);
```

Ornek:

```
public class Uygulama1 {  
    public static void main (String[] args){  
        int uzunluk;  
        String ileti="Endustri Mühendisliđi";  
        uzunluk=ileti.length();  
        System.out.println(ileti+": "+uzunluk);  
        ileti=ileti+ " Lisans programı";  
        uzunluk=ileti.length();  
        System.out.println(ileti+ "= " + uzunluk);  
    }  
}
```

EMT2226 Nesne Yönelimli Programlamaya Giriş - Yrd. Doç. Dr. İbrahim Küçükkoç

Çıktı:

```
Endustri Mühendisliđi:21  
Endustri Mühendisliđi Lisans programı= 37
```

Bir String'teki Karakterlerin Pozisyonları

- Pozisyon indis olarak alınır.
 - “Java is fun.” stringinde 'f' 9. indiste yer alır.

0	1	2	3	4	5	6	7	8	9	10	11
J	a	v	a		i	s		f	u	n	.

0'dan 11'e kadar indisler vardır

String Sınıfının Bazı Metotları

- `.toLowerCase()`: Metodu çağıran dizginin tamamen küçük harfe çevrilmiş hali olan dizgiyi geri verir.

```
String iletı="MERHABA";  
String kucukHarfli=iletı.toLowerCase();  
//kucukHarfli deęiřkeninin deęeri "merhaba" olur
```

- `.toUpperCase()`: Metodu çağıran dizginin tamamen büyük harfe çevrilmiş hali olan dizgiyi geri verir.

```
String iletı="Merhaba";  
String buyukHarfli=iletı.toUpperCase();  
//buyukHarfli deęiřkeninin deęeri "MERHABA" olur
```

- `.substring(Başlangıçİndisi)`: Dizgi nesnesinin başlangıç numaralı karakterinden dizgi sonuna kadar olan bölümünü dizgi olarak geri verir.

```
String iletı="Merhaba";  
String parca=iletı.substring(4);  
//parca deęiřkeninin ierięi "aba" olur.
```

String Sınıfının Bazı Metotları

- `.substring(Başlangıç,son)`: Dizgi nesnesinin başlangıç numaralı karakterinden son numaralı karakterine kadar olan bölümü verir.

```
String iletı="Merhaba";  
String parca=iletı.substring(1,4);  
//parca deęiřkeninin ierięi "erh" olur.
```

- `.indexOf(DizgiAdi)`: String dizgide DizgiAdi ile tanımlı başka bir dizginin ilk görüldüğü yeri bulur. Eğer DizgiAdi dizgisi kaynak dizgide yoksa -1 değeri geri döner.

```
String iletı="Bilgisayar Mühendislięi";  
int yer=iletı.indexOf("Müh");  
//yer deęiřkeninin deęeri 11 olur.
```

- `.indexOf(Bir_Dizgi,Başlangıç)`: Dizgi üzerinde Başlangıç numaralı karakterden sonra Bir_Dizgi ile tanımlı başka bir dizginin ilk görüldüğü yeri bulur. Eğer Bir_Dizgi dizgisi kaynak dizgide yoksa -1 değeri geri döner.

```
String iletı="Bilgisayar Mühendislięi";  
int yer=iletı.indexOf("is",8);  
  
//yer deęiřkeninin deęeri 17 olur.
```


String Sınıfının Bazı Metotları

- `.lastIndexOf(Bir_Dizgi)`: Dizgi üzerinde `Bir_Dizgi` ile tanımlı başka bir dizginin son görüldüğü yeri bulur. Eğer `Bir_Dizgi` dizgisi kaynak dizgide yoksa `-1` değeri geri döner.

```
String iletı="Bilgisayar Mühendisliđi";  
int yer=iletı.indexOf("Müh");  
//yer deđişkeninin deđeri 11 olur.
```

- `.trim()`: Dizginin başındaki ve sonundaki boşlukların silindiđi bir dizgi verir.

```
String iletı= " Merhaba ";  
String boslukSil=iletı.trim();  
//boslukSil deđişkeninin deđeri "merhaba"
```

- `charAt(konum)`: Dizginin üzerinde yer alan konum numaralı karakteri geri verir.

```
String iletı="merhaba";  
Char konumdakiKarakter=iletı.charAt(3);  
// konumdakiKarakter deđişkeninin deđeri 'h' olur.
```


String Sinifinin Bazi Metotlari

- `.compareTo(Bir_dizgi)`: Bir_dizgi nesnesini kaynak dizgi ile alfabetik olarak karşılaştırır. Eğer metodu çağıran nesne daha küçük ise sıfırdan küçük bir değer, iki dizgi eşit ise 0 verir. Eğer metodu çağıran nesne metoda giren parametreden daha büyük ise sıfırdan büyük sonuç verir.

```
String iletı=«merhaba»;  
int esitlik=iletı.compareTo(«merhaba»);  
// esitlik deęişkeninin deęeri 0 olur.
```

- `.equals` ve `.equalsIgnoreCase`: `equals` komutu iki dizginin eşit olup olmadığını sınar. True veya false değeri dönderir.
- `equalsIgnoreCase` ise karşılaştırmada büyük küçük harf ayrımı yapmaz.

.equals Uygulama

```
public class Uygulama1 {  
    public static void main(String[] args) {  
        String birinci="BALIKESIR";  
        String ikinci="ÜNİVERSİTESİ";  
        String ucuncu="balikesir";  
        System.out.println(birinci.equals(birinci));  
        System.out.println(birinci.equals(ikinci));  
        System.out.println(birinci.equalsIgnoreCase(ucuncu));  
        System.out.println(birinci.equals(ucuncu.toUpperCase()));  
    }  
}
```

Çıktı:
true
false
true
true

Calisma Sorulari

- **Aşağıdaki değişkenlerden hangisi geçerlidir? Hangileri java anahtar kelimeleridir?**

applet, Applet, a++, —a, 4#R, \$4, #44, apps, class, public, int, x, y, radius

- **Aşağıdaki algoritmayı koda dönüştürün:**

Adım1: ismi mil olan double türünden bir değişkeni başlangıç değeri 100 olacak şekilde tanımlayın

Adım2: ismi donusum olan double türünden sabiti değeri 1.609 olacak şekilde tanımlayın

Adım3: Kilometre adında bir değişken tanımlayın ve değerine mil ile donuşumun çarpımını atayın.

Adım4: kilometre sonucunu ekranda gösterin.

Dördüncü adım sonunda kilometrenin değeri nedir?

Calisma Sorulari

- **Int a = 1 ve double d = 1.0 alındığını varsayarak her bir ifadeyi bağımsız olarak değerlendirin?**
- **Aşağıdaki ifadelerin sonuçları nedir?**
 - a) `a = 46 / 9;`
 - b) `a = 46 % 9 + 4 * 4 -2;`
 - c) `a = 45 + 43 % 5 * (23 * 3 % 2);`
 - d) `a %= 3 / a + 3;`
 - e) `d = 4 + d * d + 4;`
 - f) `d += 1.5 * 3 + (++a);`
 - g) `d -= 1.5 * 3 + a++;`
- **Eğer bugün Salı ise 100 gün sonra hangi gün olacaktır?**
- **25/4 ifadesinin sonucu nedir? Noktalı sonuç elde etmek için nasıl yazmak gerekir?**

Calisma Sorulari

- **Aşağıdaki ifadeler doğru mudur? Eğer doğru ise çıktıları yazınız?**

```
System.out.println("25 / 4 is " + 25 / 4);
```

```
System.out.println("25 / 4.0 is " + 25 / 4.0);
```

```
System.out.println("3 * 2 / 4 is " + 3 * 2 / 4);
```

```
System.out.println("3.0 * 2 / 4 is " + 3.0 * 2 / 4);
```

- **Aşağıdaki aritmetik ifadeyi Java'da nasıl yazarsınız?**

$$\frac{4}{3(r + 34)} - 9(a + bc) + \frac{3 + d(2 + a)}{a + bd}$$

Calisma Sorulari

Aşağıdaki programdaki hataları düzeltiniz.

```
1 public class Test {  
2     Public void main(string[] args){  
3         int i;  
4         int k = 100.0;  
5         int j = i+1;  
6         System.out.println("j:"+j+" ve k:"+k);  
7     }  
8 }
```

Calisma Sorulari

- **Aşağıdaki programın çıktısını veriniz:**

```
public class Test {  
    public static void main(String[] args) {  
        char x = 'a';  
        char y = 'c';  
        System.out.println(++x);  
        System.out.println(y++);  
        System.out.println(x -y);  
    }  
}
```

System.out.println ile System.out.print arasında ne fark vardır?

- **Aşağıdaki program parçasının çıktısı nedir?**

```
String cümle="Merhaba JaVa";  
Cumle=cümle.toUpperCase();  
System.out.println(cumle);
```

- **Aşağıdaki programın çıktısını veriniz?**

```
String harfler="klmnoprs";  
System.out.println(harfler.substring(3));
```

What will this program print out:

```
String language = "java";  
String feeling = "Love";  
feeling = feeling.toLowerCase();  
language = language.toUpperCase();  
System.out.println("I " + feeling + " " + language);
```

Print output:

- Ifeelinglanguage
- I Love java
- IJAVAlove
- I love JAVA

Which of these statements is wrong ?

Select all statements where the declaration type does NOT match the value !

- char** me = 'I';
- boolean** fact = true;
- boolean** number = 17;
- String** text = "text";
- double** price = 23.75;
- long** total = 100.1;

What do you think this program will print?

```
int x = 0;
int y = 4;
double z = 3;
x = x + 2;
z = x + y - 7;
y = x * 3;
System.out.println("x = "+x);
System.out.println("y = "+y);
System.out.println("z = "+z);
```



```
x = 2
y = 0
z = -3.0
```



```
x = 0
y = 0
z = 1.5
```



```
x = 0
y = 4
z = 3.0
```



```
x = 2
y = 6
z = -1.0
```