

JAVA PROGRAMLAMA DİLİ ÖZELLİKLERİ

Java programlama dili, Java SUN bilgisayar şirketi tarafından 1991 yılında geliştiril

- Program bloklar halinde yazılır ve bloklar { } ile oluşturulur.
- Komutlar aynı satıra veya alt alta yazılabilir.
- Tüm komutlar noktalı virgül (;) ile sonlanır. Blok başlatan ifadelerden sonra kullanılmaz.
- Kullanılan tüm değişkenlerin veri tipleri belirtilir.
- Programda kullanılacak olan komutların bulunduğu kütüphaneler programın başında belirtilir.
- Günümüzde nesneye yönelik programlama dilleri (C++, Java) ve script dilleri (JavaScript,, PHP) gibi programlama dilleri C Programlama Dili'nden esinlenmiştir.
- Büyük-küçük harf duyarlıdır. main ile Main farklı değerlendirilir.

Java

- En popüler programlama dilidir,
- Sanal bir arayüzde çalışır (JVM),
- Birçok programa göre daha komplekstir (örn. Python)
- Daha basittir (C++ vb)

Program yazma ve çözüm adımları

1. Problemin analizi
2. Çözüm geliştirme
3. Algoritma ve akış diyagramının oluşturulması
4. Programın kodlarının yazımı
5. Programın test edilmesi

ÖRNEK 1:

```
class Hosgeldiniz
{
public static void main(String args[])
{
System.out.println("Java Sinifina Hos Geldiniz!");
}
}
```

- ✓ // işaretiyle başlayan yazılar bilgisayar tarafından kullanılamaz sadece programı okuyanı bilgilendirmek amacı taşır /* */ işaretleriyle de verilebilir.
- ✓ Örneğin : /* Bu kısımda yazılanlarla bilgisayar ilgilenmez Sadece insanların programı anlaması içindir */
- ✓ class Hosgeldiniz Class (sınıf) nesne kökenli programlama yapısının temel taşıdır. Her Class'ın kendine özgü bir ismi olması gerekir.
- ✓ Daha önce yazılmış Class'lar **import** komutuyla programa çağırılabilirler. Yeniden yazılmaları gerekmez.
- ✓ { } başlatma ve bitirme : Sınıf tanımlaması da dahil olmak üzere Tüm Program parçacıkları { işaretiyle başlar ve } işaretiyle son bulur.
- ✓ ";" işareti java komutunun bittiğini belirtir. Cümle yağısında kullanılan nokta "." işareti gibi işlev görür.
- ✓ Java programlarını istersek ";" ve "{}" ayrıçalarını kullanarak tek bir satır olarak yazabiliriz.

```
class Hosgeldiniz
{
public static void main(String args[])
{
System.out.println("Java Sinifina Hos Geldiniz!");
}
}
```

- ✓ `public static void main(String args[])` diğer dillerde fonksiyon (function), subroutine, procedure gibi adlarla anılan Program alt parçacığının başlığı. Bu Program parçacıkları javada Metot (Method) adıyla anılır.
- ✓ Metotların girdileri ve çıktıları olabilir. Bu metotun adı `main` (ana Program) dır. `String` değişken türü `args[]` değişkeni girdi olarak girebilir.
- ✓ Bu metotta `void` çıktı türü kullanılmıştır. Void kelimesi bu metotdan hiçbir çıktı alınmayacağını belirtir. `public` metota dışarıdan ulaşılabilme izni verir.
- ✓ `System.out.println("Java Sinifina Hos Geldiniz!");` Bu satır `System.out` sınıfından `println` metotunu çalıştırır.
- ✓ Bu metot method girişinde gönderilen `String` tipi değişkeni konsol çıktısına aktarır.
- ✓ "Java Sinifina Hos Geldiniz!" stringi aynen aktarılırken içinde `c` dilinde kullanılan temel kontrol terimleri yer alabilir ve aynen `c` dilindeki anlamlarıyla kullanılabilir.

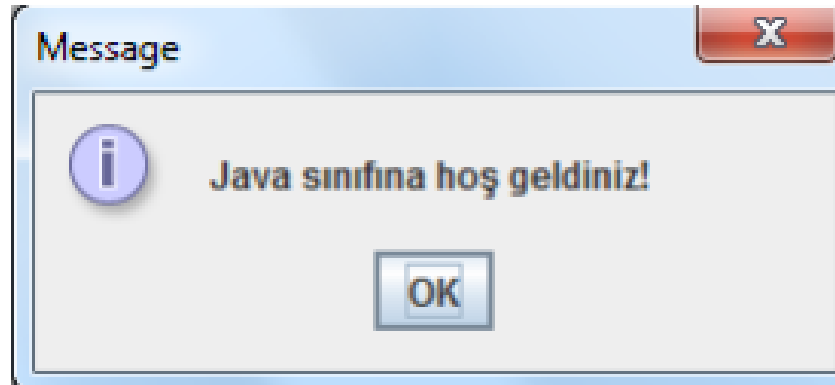
ÖRNEK 2:

```
import javax.swing.JOptionPane;

class Hosgeldiniz1 {
    public static void main (String args[])
    {JOptionPane.showMessageDialog(null,"Java sınıfına hoş geldiniz!");}
}
```

SWING kütüphanesinde yer alan **JOptionPane.showMessageDialog(...)** metodunu kullanılarak...

Hosgeldiniz1 dosyasının konsol **SWING pop-up** grafik çıktısı kullanarak görülmesi



CPU Talimatları

$$Z = x + y$$

x konumu oku,

y konumu oku,

topla

z konumuna yaz

Binary	Hexadecimal	Binary	Hexadecimal	Binary	Hexadecimal
0 0 0 0	0	0 1 0 1	5	1 0 1 0	A
0 0 0 1	1	0 1 1 0	6	1 0 1 1	B
0 0 1 0	2	0 1 1 1	7	1 1 0 0	C
0 0 1 1	3	1 0 0 0	8	1 1 0 1	D
0 1 0 0	4	1 0 0 1	9	1 1 1 0	E
				1 1 1 1	F



Programlama Dilleri

- Kolay anlaşılabilir olmalı
- Girilen bilgilerin anlaşılabilir olması için CPU nun anlayacağı dile çevrilmeli

1: Types, Variables, Operators

- **Hedef**

- Yeteri kadar Java öğrenerek faydalı birşeyler yapmak

Örnek:

- Bir mühendislik problemi çözmek,
- Veri tabanları kontrol etmek
- Grafik çizmek vb.

NetBeans- New Project

The screenshot displays the NetBeans IDE 8.1 interface during the creation of a new project. The main window shows a project named 'Ornek1' with a package 'ornek1' and a class 'Ornek1.java'. The 'New Project' dialog is open, showing the 'Choose Project' step. The 'New Java Application' dialog is also open, showing the 'Name and Location' step.

NetBeans IDE 8.1
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

New Project

Steps

1. Choose Project
2. ...

Choose Project

Filter:

Categories:

- Java
- JavaFX
- Java Web
- Java EE

Projects:

- Java Application
- Java Class Library
- Java Project with Existing Sources
- Java Free-Form Project

Ornek1 - NetBeans IDE 8.1

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

New Java Application

Steps

1. Choose Project
2. Name and Location

Name and Location

Project Name:

Project Location:

Project Folder:

Use Dedicated Folder for Storing Libraries

Libraries Folder:

Different users and projects can share the same compilation libraries (see Help for details).

Create Main Class

```
1  /**  
2  * To change this license header, choose License Headers in Project Properties.  
3  * To change this template file, edit the template files in the source/ folder.  
4  * and open the template files in the editor.  
5  */  
6  package ornek1;  
7  
8  /**  
9  *  
10 * @author Caner Koç  
11 */  
12 public class Ornek1 {  
13  
14     /**  
15     * @param args the command line arguments  
16     */  
17     public static void main(String[] args) {  
18         // TODO code a  
19     }  
20  
21 }  
22
```

```
package ornek1;
```

```
public class Ornek1 {
```

```
    public static void main(String[] args) {
```

```
        System.out.println("selam dünya");
```

```
    }
```

```
}
```

Program Yapısı

```
public class Ornek1 {  
  
    public static void main(String[] args) {  
        KOMUTLAR (CODES) ....  
    }  
  
}
```

Output

`System.out.println(some String)` outputs to the console

Example:

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

İkinci Program

```
package ornek2;
```

```
public class Ornek2 {
```

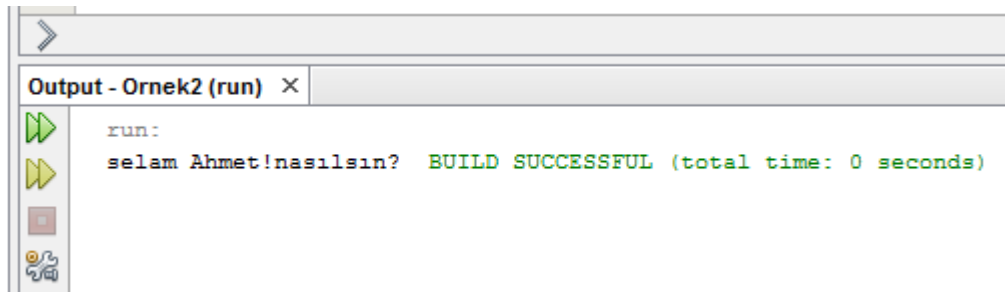
```
    public static void main(String[] args) {
```

```
        System.out.print("selam Ahmet!");
```

```
        System.out.print("nasılsın? ");
```

```
    }
```

```
}
```



The screenshot shows an IDE output window titled "Output - Ornek2 (run) x". The output text is as follows:

```
run:  
selam Ahmet!nasılsın? BUILD SUCCESSFUL (total time: 0 seconds)
```

The output window includes standard IDE icons for running, stopping, and refreshing.

Veri Tipleri (Types)

Farklı tiplerdeki veriler, farklı şekillerde depolanır ve değiştirilebilirler.

Boolean: Gerçek değer (True or False).

İnt (integer): Tam sayılar (0, 1, -47)

Double : Gerçek sayılar (3.14, 1.0, -2.21)

String : Yazı ('selam ', 'naber').

Bit(binary digit): En küçük depolama büyüklüğüdür. (1 veya 0, doğru veya yanlış)

Veri Tipi	Alan Büyüklüğü	Kategori
byte	8 bit	Tamsayı Tipleri
short	16 bit	
int	32 bit	
long	64 bit	
float	32 bit	Kesirli Sayı Tipleri
double	64 bit	
char	16 bit	Karakter Tipi
boolean	-	Mantıksal Tip (ture/false)

Değişkenler

Saklayabilmek veya değer atayabilmek için özel bir İSİM VERİLİR.

Veri formu:

```
VERİ TİPİ ADI;
```

Örnek:

```
String Ad;
```

```
String Soyad;
```

```
int yas;
```


ÖRNEK :3

- AD VE SOYAD DEĞİŞKEN OLARAK TANIMLA
- YAŞI DEĞİŞKEN OLARAK TANIMLA
- YAZDIR.

ÖRNEK 3:

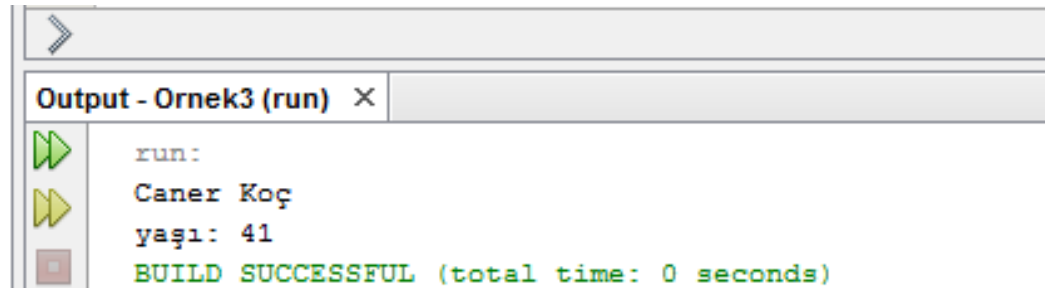
```
package ornek3;

public class Ornek3 {

    public static void main(String[] args) {
        String ad;
        int yas;

        ad = "Caner Koç";
        yas = 41;

        System.out.println(ad);
        System.out.println("yaşı: " + yas); // TODO code application logic here
    }
}
```



Output - Ornek3 (run) ×

```
run:
Caner Koç
yaşı: 41
BUILD SUCCESSFUL (total time: 0 seconds)
```

ÖRNEK 4:

- KESİRLİ SAYILARI DEĞİŞKEN OLARAK TANIMLA
- YAZDIR

```
package ornek4;
```

```
public class ORNEK4 {
```

```
    public static void main(String[] args) {
```

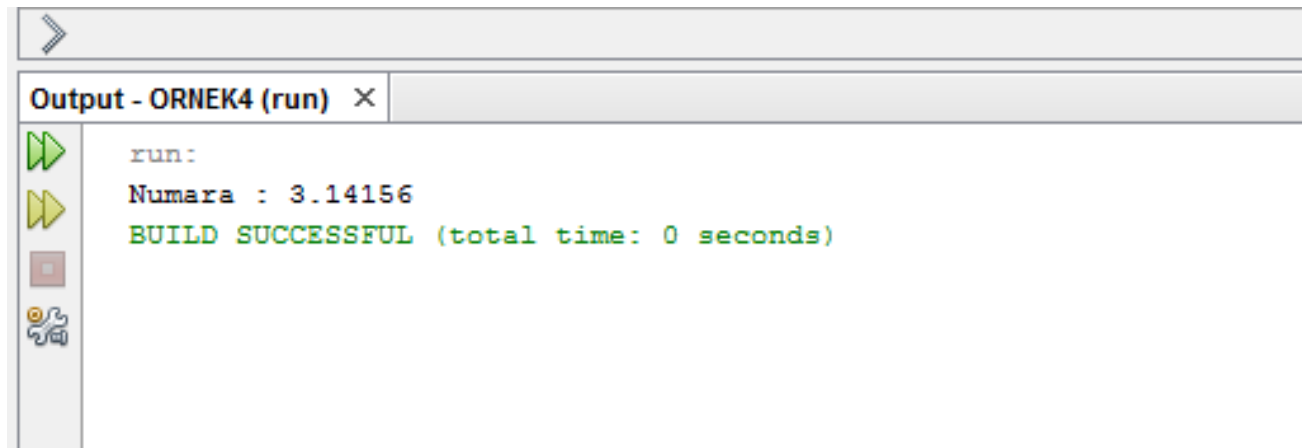
```
        Double numara;
```

```
        numara = 3.14156;
```

```
        System.out.println("Numara : "+ numara );// TODO code application logic here
```

```
    }
```

```
}
```



```
run:  
Numara : 3.14156  
BUILD SUCCESSFUL (total time: 0 seconds)
```