

1. HAFTA

TEMEL KURULUM İŞLEMLERİ ve NESNE TABANLI PROGRAMLAMA KAVRAMLARI

TMYO

**Bilgsayar Teknolojileri Uzaktan Eğitim
Bölümü**

Nesne Tabanlı Programlama

Zafer YAVUZ

zaferyavuz@ktu.edu.tr

Sunum İçeriği

- **NESNE TABANLI PROGRAMLAMA**
- **TEMEL KURULUM İŞLEMLERİ**
- **JAVA HAKKINDA TEMEL BİLGİLER**
- **TEMEL BİR UYGULAMASI GELİŞTİRMEK**
 - Dilin genel yazım kuralları
 - Basit gramer hataları
 - Merhaba Dünya uygulaması
 - Sayılar arasında 4 işlem
 - Basit string işlemleri

Nesne Tabanlı Programlama Nedir?

- Nesne tabanlı programlamada esas olan, gerçek hayatta var olan olguların programlamaya aktarılmasındaki yeni yaklaşımdır.
- Klasik prosedürel programlamada verilerimiz ve fonksiyonlarımız vardı.
- Yani her şey veri ve bu veriyi işleyen metotlar etrafında dönüyordu.
- Nesne tabanlı programlamada da iki önemli birim; veri ve veriyi işleyip mantıklı sonuçlar üreten metotlar bulunur.
- Ama buradaki fark gerçek hayattaki olguların da daha iyi gözlenip programlama dünyasına aktarılmasındadır.

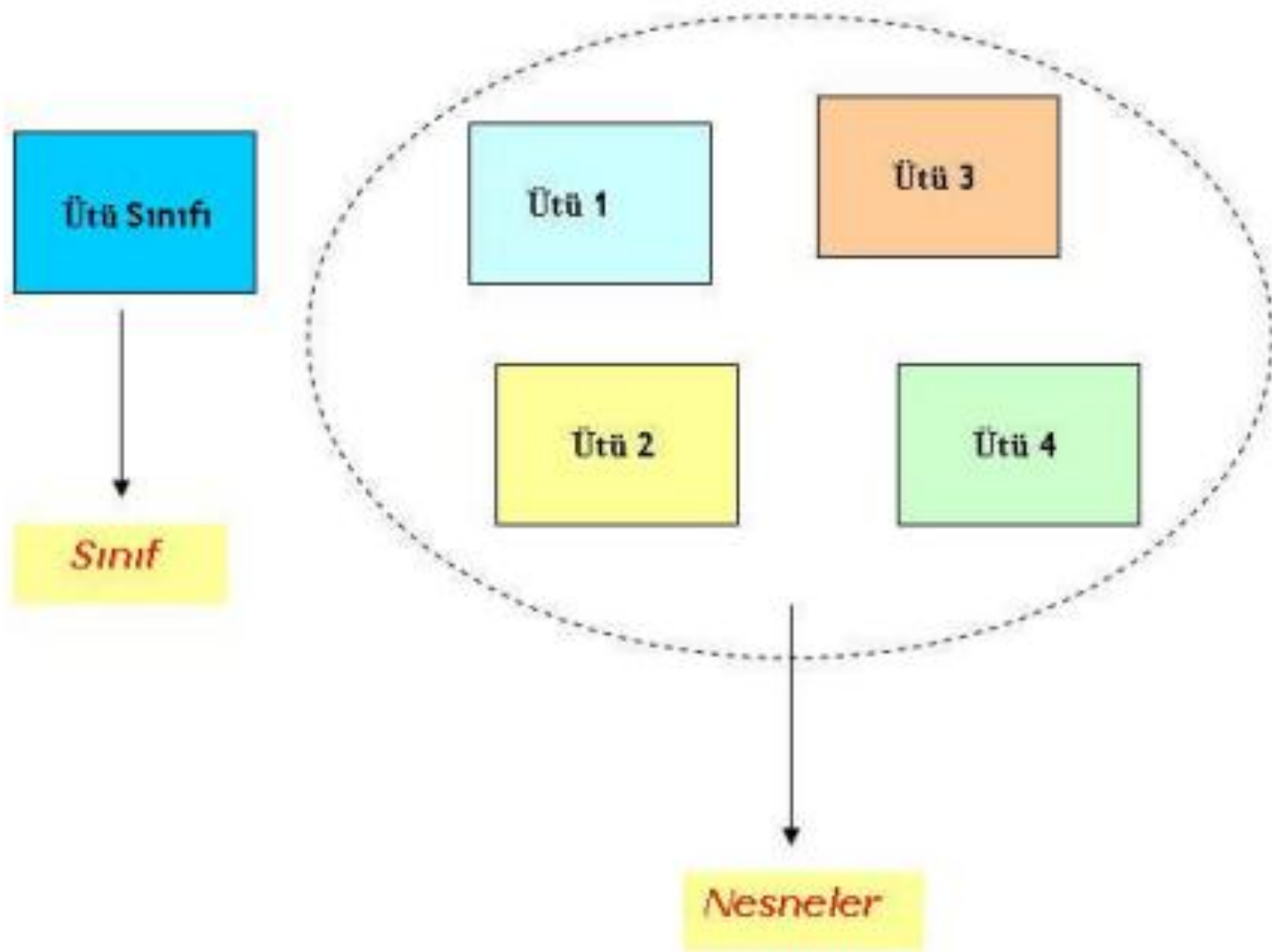
Nesne tabanlı programlama modeli



Nesne Tabanlı Programlama

- Mesela elimizde bir ütümüz olsun.
- Ütünün markası, modeli, rengi, çalıştığı elektrik voltajı, ne tür kumaşları ütüleyebildiği bu ütüye ait özelliklerdir (veri).
- Aynı zamanda ütümüzü ısıtabiliriz, ütüleme işinde kullanabiliriz ve soğumaya bırakabiliriz. Bunlar ise ütünün fonksiyonlarıdır (metot).

Sınıf (Class) ve nesne (Object) kavramı



- Eğer ütü ile ilgili bir program yapmış olsak ve nesne tabanlı programlama tekniğini kullansak hemen bir ütü sınıfı (class) oluştururduk.
- Bu sınıfta ütüye ait bilgiler (veriler) ve ütü ile yapabileceğimiz işler (metot) bulunurdu.
- O zaman nesne tabanlı programlamada bir sınıfta, sınıfa ait veriler ve bu verileri işleyip bir takım faydalı sonuçlar üreten fonksiyonlar / metotlar bulunur.
- Dahası, biz bir tane ütü sınıfı tasarlırsak bu sınıftan istediğimiz sayıda değişik ütüler (Object veya instance) yapabiliriz.
- Nesne tabanlı programlamanın bazı temel kavramları vardır.

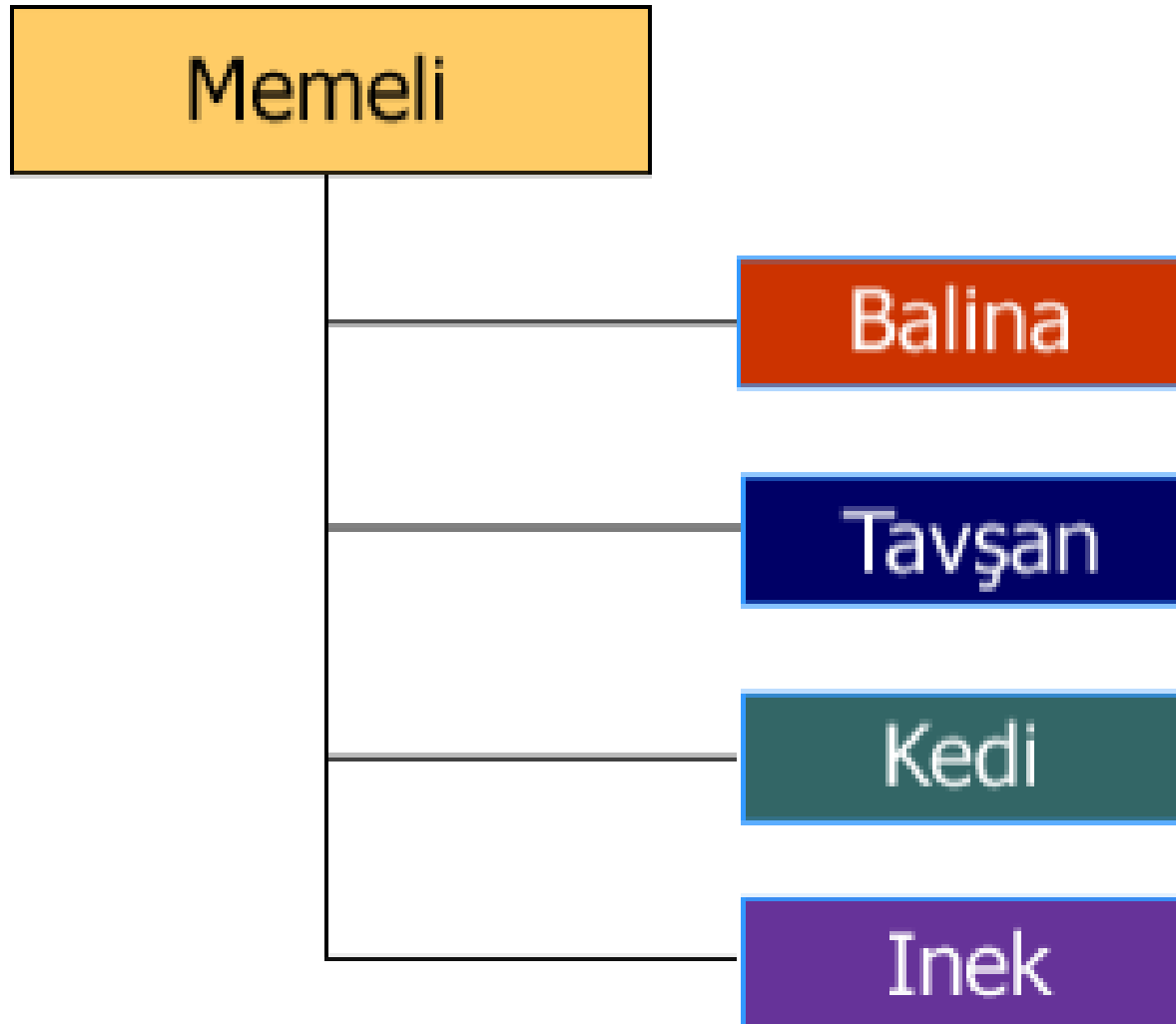
Kapsülleme (Encapsulation)

- İşlemlerin ve verilerin gizlenmesine veya bir grup içerisinde toplanmasına kapsülleme denir. (Arabanın elk. Sistemi, yakıt sistemi, vites sistemi gibi)
- Kapsülleme sayesinde, sınıfımız içinde kullandığımız alanların dışarıdan herhangi bir etki ile doğrudan değiştirilmelerini engellemiş oluyoruz.
- Bu durum bizim isteğimiz dışında oluşabilecek atamaların önüne geçmemize ve ille de bu alanların değerleri değişecekse bizim belirlediğimiz bir çizgide olmasına neden oluyor.
- Bu da programcılıkta ve özellikle nesne tabanlı programlamada çok büyük bir yere sahip olan kontrolün elimizde olmasını sağlıyor.

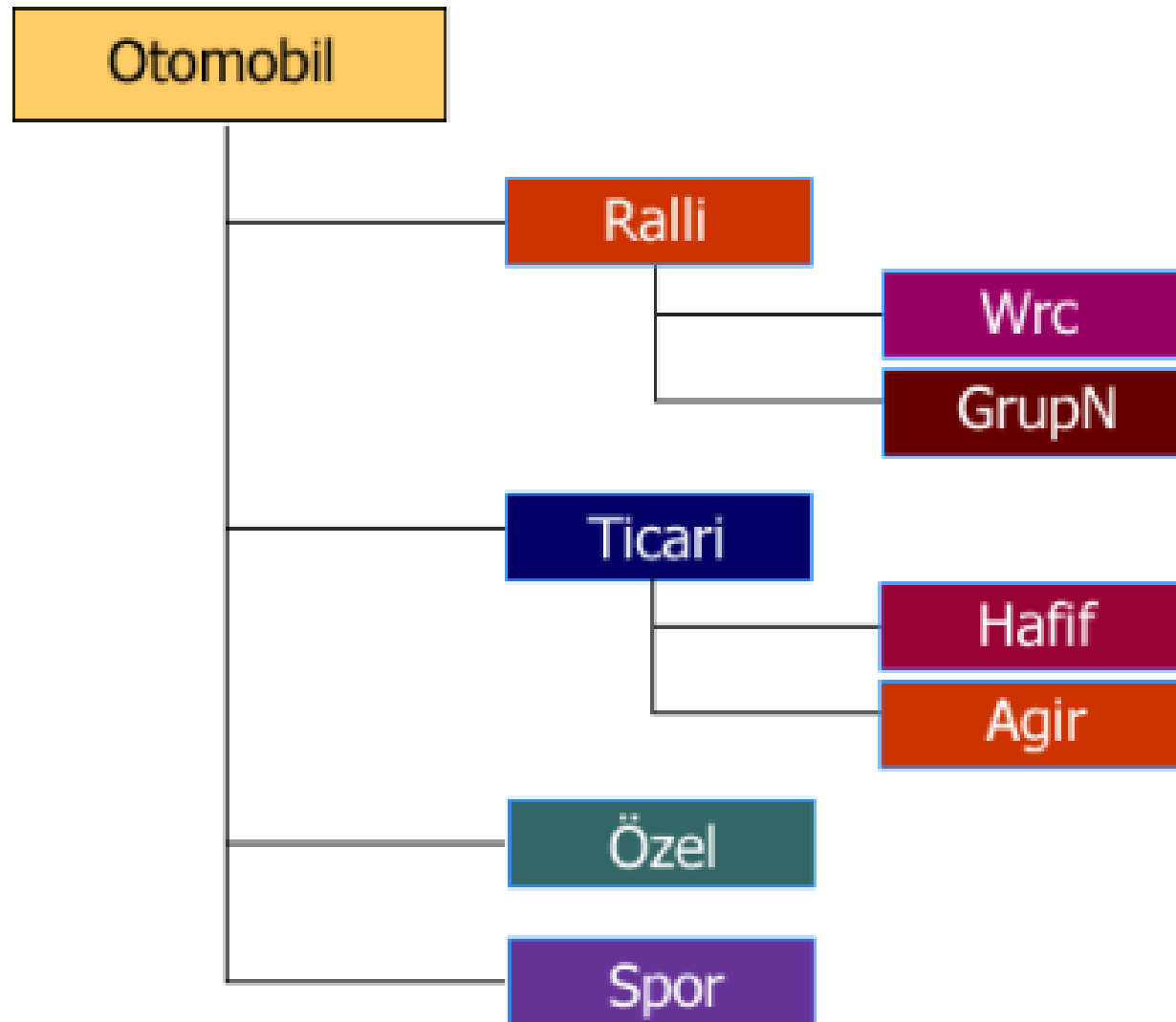
Kalıtım (Inheritance)

- Normalde bu kavramı gerçek hayattan biliyoruz.
- En basit anlamda, örneğin “Ben, annemin gözlerini almışım.” dediğimizde, tıp uzmanlarının buna getirdikleri yorum " siz annenizden kalıtımsal olarak şu özellikleri almışsınız." oluyor.
- Programlama dillerinde de kalıtımın rolünün aynı olduğunu söyleyebiliriz.
- Zaten nesne tabanlı programlama dillerini tasarlayan uzmanlar, gerçek hayat problemlerini, bilgisayar ortamına taşıyabilmek amacıyla en etkili modelleri geliştirmişler, bu model içerisine kalıtımı da katarak çok önemli bir özelliğin kullanılabilmesini sağlamışlardır.

Kalıtım (Inheritance) kavramı

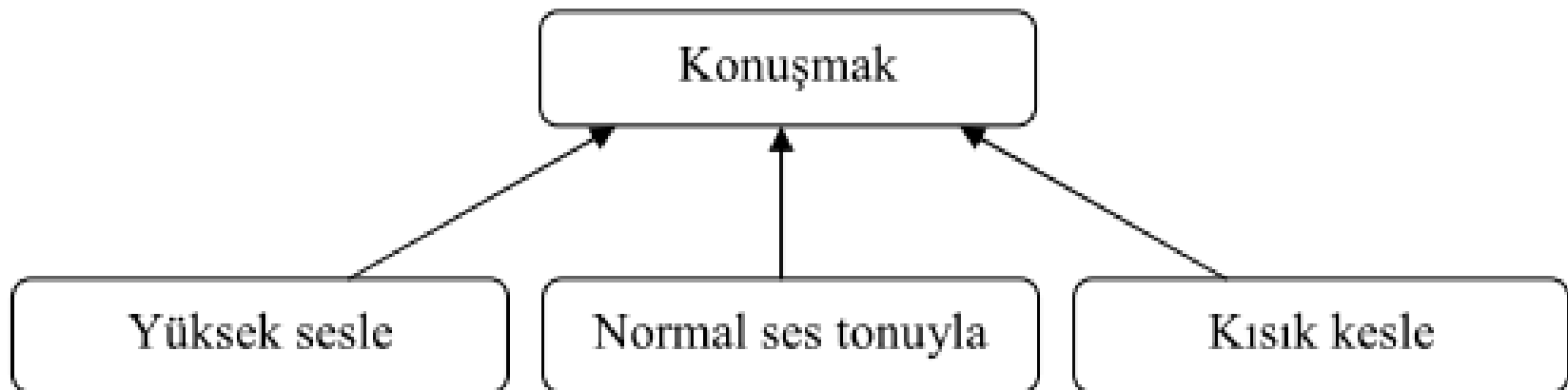


Kalıtım kavramına bir başka örnek



Çok Biçimlilik (Polymorphism)

- Bir metodun farklı nesnelerde veya farklı ortamlarda farklı sonuçlar üretmesidir.
- Mesela “konuşmak” insan sınıfına ait bir metottur.



JVM, JRE ve JDK

- JDK NEDİR?

Java, platform bağımsız ve yüksek performanslı uygulamaların geliştirilmesine olanak sağlayan basit, nesne tabanlı bir programlama dilidir. Java kullanılarak yazılan bir programın işletim sisteminden bağımsız olarak her bilgisayarda çalışabiliyor olması diğer hiçbir yazılım programında olmayan bir özelliktir.

Ayrıca Java tamamiyle ücretsizdir. Tabi Sun (Oracle) istediği bir zamanda bunu ücretli halede getirebilir.

Java ile program geliştirebilmek için önce Java geliştirme sistemini (JAVA 2 SOFTWARE DEVELOPMENT KIT - STANDART EDITION = J2SE) Sun Microsystem'in internet adresinden indirmemiz gerekiyor. JDK ücretsiz bir yazılımdır.

Java ile geliştirme (development) yapmak için Java Development Kit'e (JDK) ihtiyacınız var. Bu paket Java ile geliştirme yapmak için bütün araçları içeriyor..

JVM, JRE ve JDK

- JRE nedir?

JRE (Java Runtime Environment) Java Virtual Machine, Java Platform çekirdek class'larını ve destekleyici dosyalar gibi paketleri içerir. Yani java kodu geliştirmek için gerekli olan derleyici (compiler), debugger gibi paketleri içermez. JRE zaten JDK'nın içinde vardır. Böyle bir paketin oluşturulmasının nedeni ise basit: Yazdığınız ve derlediğiniz (compile) Java class'larını herhangi başka bir makinede çalıştırmak için bu pakete ihtiyacınız olabilir.

JVM, JRE ve JDK

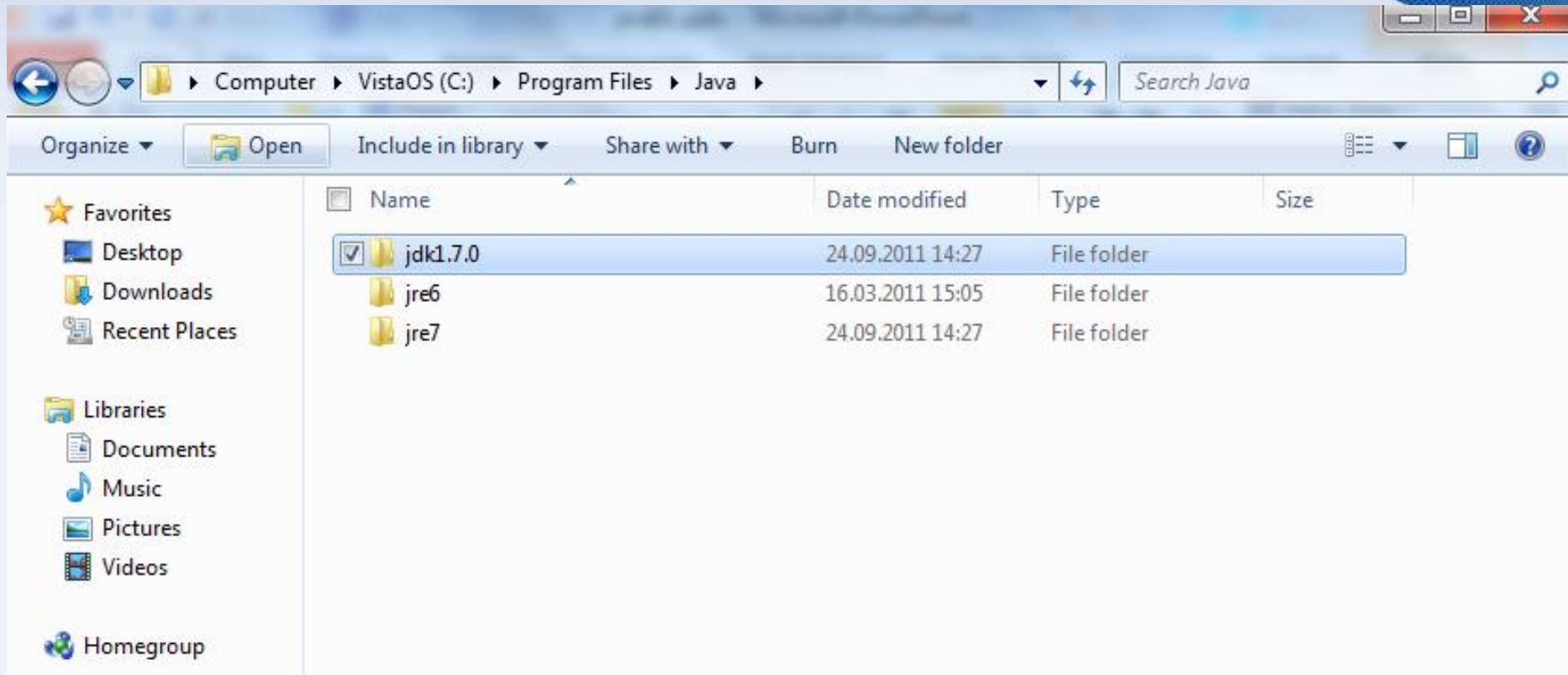
- JVM NEDİR?

jvm , java öncelikli derlenen programların çalıştırıldığı açık bir bilgisayardır.küçük ve basit olarak dizayn edilmiştir çünkü düşünülen onu her yere indirmektir . bu basitlik onu tüm varolan platformlara taşımayı mümkün kiliyor , böylece hepsine ortak bir yazılım sağlıyor ve her karışık ağ sisteminde (internet gibi) büyük ölçüde taşınabilirlik sorununu ortadan kaldırır.hayal edilebildiği gibi bu, durdurulamayan teknoloji için yaklaşık bir nedendir. bununla beraber sanal makine bir yazılım tabakası olarak görülmemelidir, fakat isminin belirttiği gibi, bu dolu bir bilgisayar sistemi olmaya çalışıyor,yavas yavas gerçek java makinelerini , kredi kartları ,tv dekodeerlerini ve benzerlerini destekleyen java Çiplerini marketlerde görüyoruz.

JVM, JRE ve JDK

- JDK indirmek için
 - <http://www.oracle.com/technetwork/java/javase/downloads/java-se-jdk-7-download-432154.html>

JDK Kurulumu



Editör Kurulumu

- Java programları geliştirmek için NetBeans editör programını kurmak gerekir.
- Başka editörler de kullanabilirsiniz. Hatta not defteri uygulaması, java programları geliştirmek için yeterlidir.
- Ancak bir çok kütüphaneyi desteklemesi ve editörünün güçlü olması nedeniyle Netbeans editörü tercih edilmiştir.
- Bu editör ücretsiz olarak şu adresten indirilebilir.

NetBeans Web Sitesi

netbeans.org/index.html

Sayfa KTU-Ceng Çalışmalarım İncelenecek Toolox Wavelets - A ... Wavelet Based Meth... folk.uio.no/oyvindry... My Opera Mail PDF converter, creat...

NetBeans Choose page language Search: Google™ Custom Search GO!

Home IDE Platform Plugins Docs & Support Community Partners



NetBeans IDE 7.0.1

Develop desktop, mobile and web applications with Java, PHP, C/C++ and more.

Runs on Windows, Linux, Mac OS X and Solaris. NetBeans IDE is open-source and free.

[Download FREE NetBeans IDE 7.0.1](#)

[Learn More about NetBeans IDE](#)



NetBeans @ JavaOne 2011

Experience the Smarter Way to Code

San Francisco October 2 - 6, 2011

Featured News: [NetBeans IDE 7.0.1 Patch 1 Now Available](#) [See All News](#)

Tutorials and Demos

[General Java Development](#)

[Java GUI Applications](#)

[Java EE & Java Web Applications](#)

Plugins

[Switch File Read/Write Access plugin](#)

[PHP Smarty Framework](#)

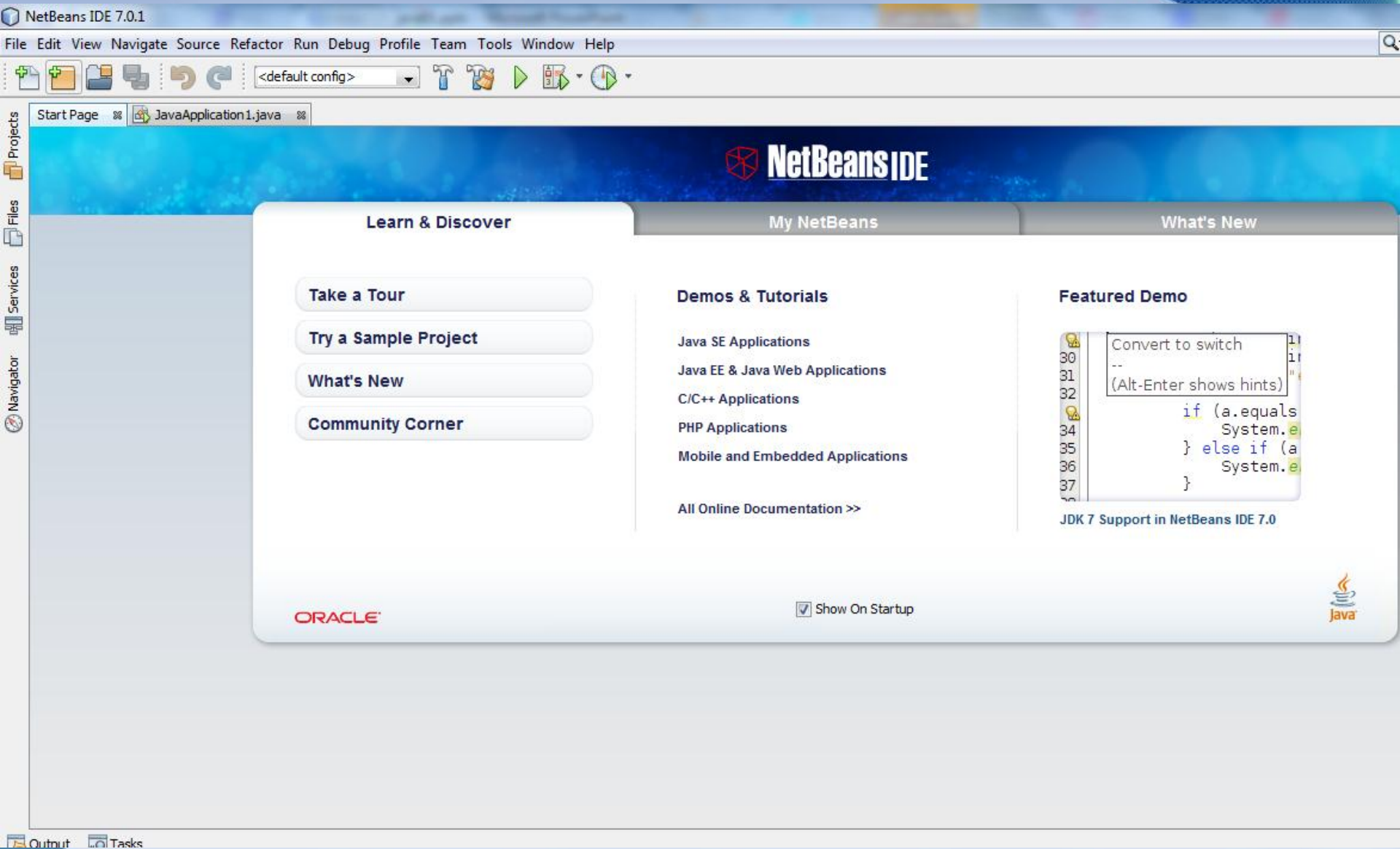
[DTrace GUI plugin](#)

NetBeans Community Poll

How many open source projects do you contribute to?

☐ 1

Netbeans



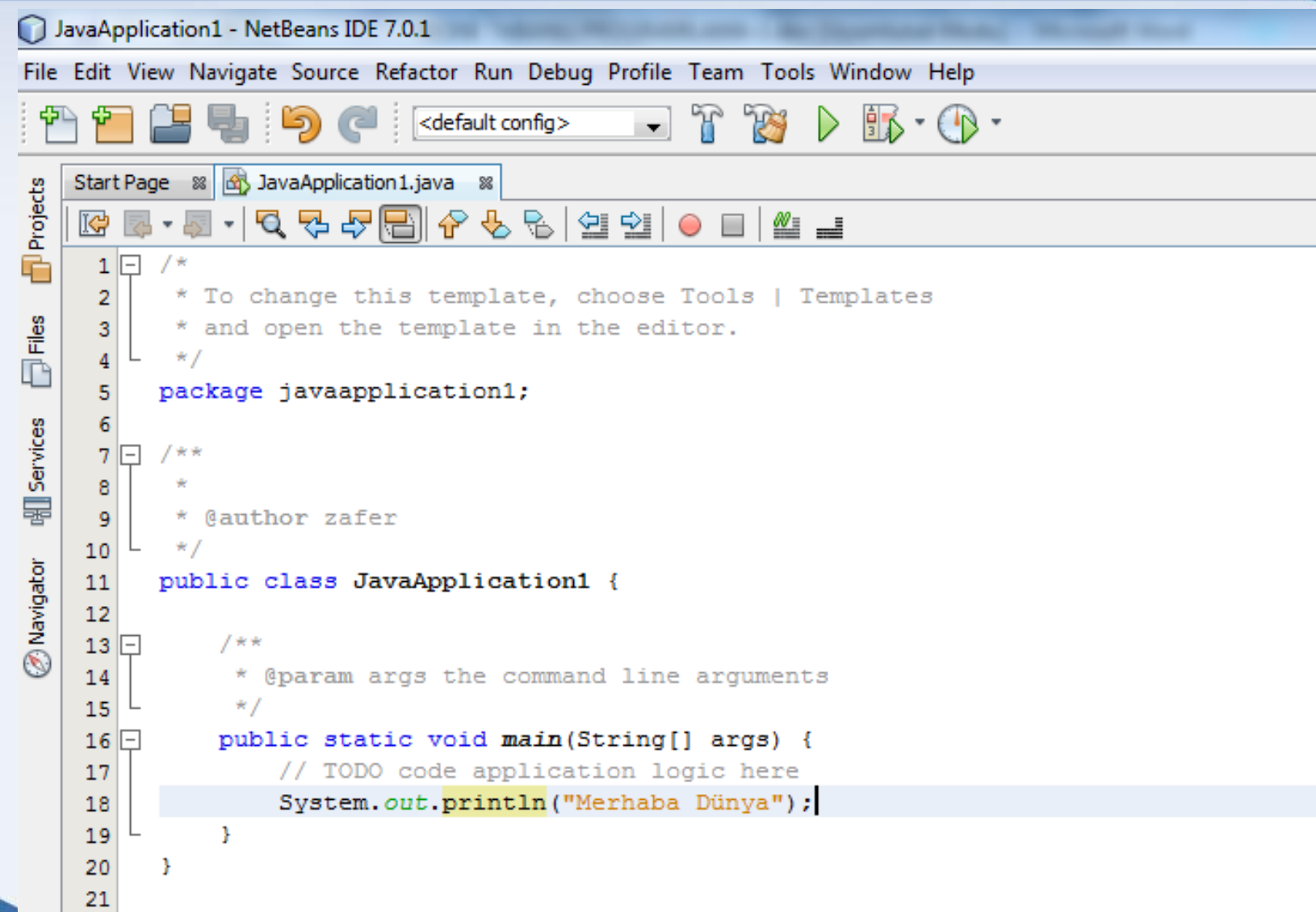
Java Hakkında

- Java, [Sun icrosystems](#) mühendislerinden [James Gosling](#) tarafından geliştirilmeye başlanmış açık kodlu, nesneye yönelik, zeminden bağımsız, yüksek verimli, çok işlevli, yüksek seviye, adım adım işletilen (interpreted) bir [dildir](#).
- Yaygın olarak proje geliştirmede kullanılan bir dildir.

Java Programlama Dili

- Java Sun Microsystems'den James Gosling tarafından geliştirilen bir programlama dilidir (Sun Microsystem'in şu anda Oracle Corporation ile bağlı ortaklığı bulunmaktadır) ve 1995 yılında Sun Microsystems'in çekirdek bileşeni olarak piyasaya sürülmüştür.
- Bu dil C ve C++'dan birçok sözdizim türetmesine rağmen bu türevler daha basit nesne modeli ve daha az düşük level olanaklar içerir.
- Java uygulamaları bilgisayar mimarisine bağlı olmadan herhangi bir Java Virtual Machine(JVM)'de çalışabilen tipik bytecode(sınıf dosyası)dır.

İlk Program



public

public anahtar sözcüğü bir erişim belirleyici olup, programcıya her bir değişkenin ya da yöntemin kontrolünü verir. Burada public anahtar sözcüğü herhangi bir sınıfın main yöntemini görebildiğini göstermektedir.

private

Bu tip bir sınıfa yalnızca paket içerisinde ulaşılabilir. Bir sınıfın tanımlanması sırasında public ya da private olduğu belirtilmediği zaman, sınıfın otomatik olarak private olduğu kabul edilir.

abstract

Soyut yöntemlerin yer aldığı sınıfları belirtir. Soyut yöntemler program kodu olmayan anlamına gelmektedir. Soyut sınıflar diğer sınıflar için yapı oluşturur.

static

Bir sınıfa uygulanan yöntem ve değişkenlerin static olduğu bildirilir. Yöntem, sınıfın özel bir örneğinin çalıştırılmasına gerek kalmadan çağrılabilir. Main durumunda örneklerden önce yorumlayıcı tarafından çağırıldığı için, static olduğunu bildirmek gerekir. Static olduğu bildirilen yöntemler, statik ve yerel değişkenlerden başka herhangi bir şeye doğrudan başvuramayabilir. Nesneye yönelik programlama dillerinde statik değişken ve yöntemlere sınıf değişkenleri ve sınıf yöntemleri adı verilmektedir.

void

Genellikle birtakım özel türlerin değerleri için yöntemler kullanılır. Örneğin tamsayı değerler için int, gerçel sayı değerleri için float ya da daha karmaşık değerler için bir sınıfın özel adları kullanılmaktadır.

Temel Değişken Türleri

Degisken Türü	Türkçe karsiligi	Bit büyüklüğü	Sinir Degerleri
boolean	Mantik degiskeni	1	true(dogru) , false(yanlis)
char	harf degiskeni	16	'\u0000' den '\uFFFF'
byte	tam sayi degiskeni	8	-128 den 127 e
short	tam sayi degiskeni	16	-32768 den 32767 e
int	tam sayi degiskeni	32	-2157483648 den 2147483647 e
long	tam sayi degiskeni	64	-9223372036854775808 den 9223372036854775808 e
float	Gerçek sayi degiskeni	32	-3.40292347e+38 den 3.40292347e+38 e
double	Gerçek sayi degiskeni	64	-1.7976931348623157e+308 den 1.7976931348623157e+308 e

`\n` alt satıra geç
`\t` bir sonraki tab'a atla
`\r` gir tusu bas
`\\` \ isaretini yaz
`\"` " isaretini yaz

- DEMO
 - Değişken türleri
 - Tür Dönüşümleri
- Dört İşlem Uygulaması
- Kullanıcıdan girdi alma
- String nesnesinin kullanımı

Aritmetik Operatörler

- + toplama (string birleştirme)
- - çıkarma
- * çarpma
- / bölme
- % mod alma
- +=, -=, *=, /=
- Atama Operatörü
=

İlişkisel Operatörler

- == eşit mi
- != eşit değil
- > büyük
- >= büyük eşit
- < küçük
- <= küçük eşit

Operatörler (Diğer)

- **Birim Operatörler**
 - ++ (1 arttır)
 - -- (1 azalt)
- **Şart Operatörleri**
 - && Conditional-AND
 - || Conditional-OR
 - ?: Şart operatörü
 - ! Değil operatörü

Operatör Öncelikleri

Operator Precedence	
Operators	Precedence
postfix	<code>expr++ expr--</code>
unary	<code>++expr --expr +expr -expr ~ !</code>
multiplicative	<code>* / %</code>
additive	<code>+ -</code>
shift	<code><< >> >>></code>
relational	<code>< > <= >= instanceof</code>
equality	<code>== !=</code>
bitwise AND	<code>&</code>
bitwise exclusive OR	<code>^</code>
bitwise inclusive OR	<code> </code>
logical AND	<code>&&</code>
logical OR	<code> </code>
ternary	<code>? :</code>
assignment	<code>= += -= *= /= %= &= ^= = <<= >>= >>>=</code>

- DEMO
 - Operatörler ve öncelikleri

DENETİM DEYİMLERİ

- Denetim deyimleri, koşulun mantıksal durumuna göre programın akışını değiştiren yapılardır.
 - If Deyimi
 - If-Else Deyimi
 - Switch Case Deyimi
 - For
 - While

IF Deyimi

- If deyimi belirli bir koşul sağlandığında yapılacak işleri belirlememizi sağlar.
 - Örnek

```
if (<koşul>
<deyim>;
If (a<3)
a=a+1;
```

If-else yapısı

- Koşul sağlandığında ya da sağlanmadığında yapılacak işleri belirlememizi sağlayan yapıdır.
 - Örnek

```
if (<koşul>
<deyim1>;
else
<deyim2>;
if (a<5)
a=a+1;
else
a=a-1;
```

If-else-if yapısı

- Birden fazla koşulu kontrol edeceğimiz durumlarda kullandığımız yapılardır.
 - Örnek

```
if (<koşul 1>) <deyim 1>;  
else if (<koşul 2>) <deyim 2>;  
else if (<koşul 3>) <deyim 3>;  
....  
else <deyim 4>;  
if (a==1) {a=a+1;}  
else if (a==2) {a=a+2;}  
else if (a==3) {a=a+3;}  
else a{=a+4;}
```

if-else Demo

- Vergi İadesi hesabı

Switch-case yapısı

- Bir ifadeye ait birden fazla koşulu kontrol etmemizi sağlayan yapılardır.
 - Örnek

```
switch (<ifade>) {  
  case <sabit 1>: <deyim 1>; break;  
  case <sabit 2>: <deyim 2>; break;  
  ....  
  case <sabit n>: <deyim n>; break;  
  default: <deyim>;  
}  
  
switch(a){  
  case 1: a=a+1;  
  break;  
  case 2: a=a+4;  
  break;  
  case 3: a=a+7;  
  break;  
  default: a=a+9;  
  break;  
}
```

Switch-case DEMO

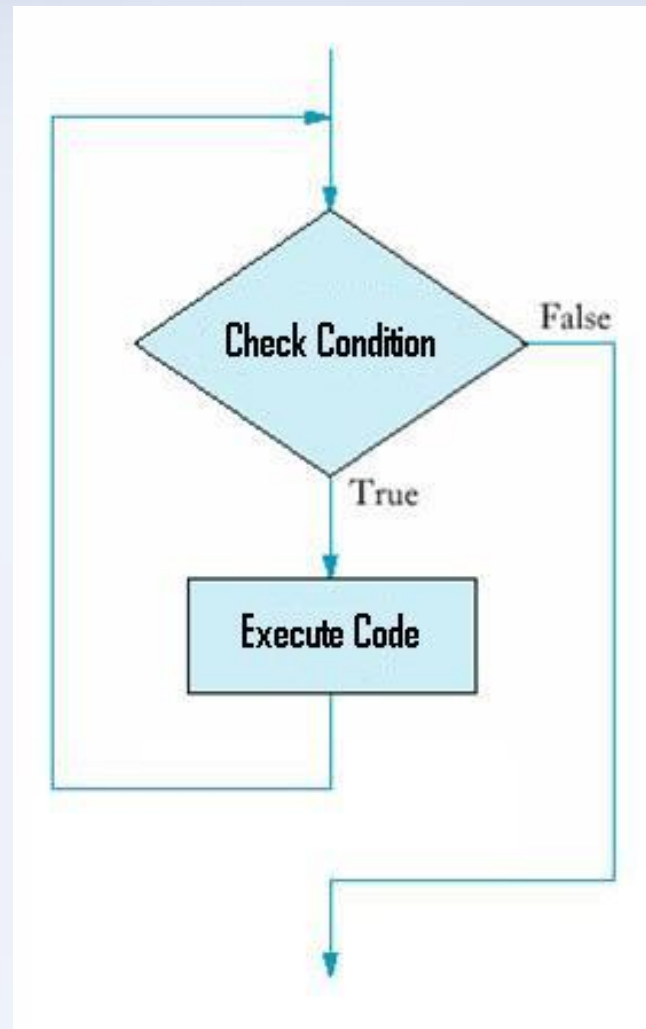
- Öğrenci notlarının harfli sistemdeki karşılığının bulunması

While döngüsü

- Java'da while döngüsü, çok basit bir temel üzerinde tasarlanmıştır. while komutu içerisindeki expression (ifade) “true” değerini döndürdüğü sürece while döngüsü bloğu içerisindeki komutlar işleme alınır. Eğer expression, bir sonraki döngüde “false” değerini döndürür ise döngü bloğunun bittiği yerden program çalışmaya devam eder.

```
while(ifade)
{
    //döngü kodları
}
```

While döngüsü



While döngüsü - örnek

```
/**
Program 1 ile 100 arasındaki tüm çift sayıları
ekrana sırasıyla yazdırır
*/
package org.javablog;

public class Test
{
    public static void main(String[] args) {
        int i = 0;
        while(i < 100)
        {
            if(i % 2 == 0)
            {
                System.out.printf("%d ", i);
            }
            i++;
        } // end of while loop
    } // end of main
} // end of class
```

While - DEMO

- *Program 1 ile 100 arasındaki tüm tek sayıları ekrana sırasıyla yazdırır*
- *Program 1 ile 100 arasındaki tüm çift sayıların toplamı*
- *Program 1 ile 100 arasındaki tüm tek sayıların toplamı*
- *Ekranda * karakterleri ile üçgen çizme*

do-while Döngüsü

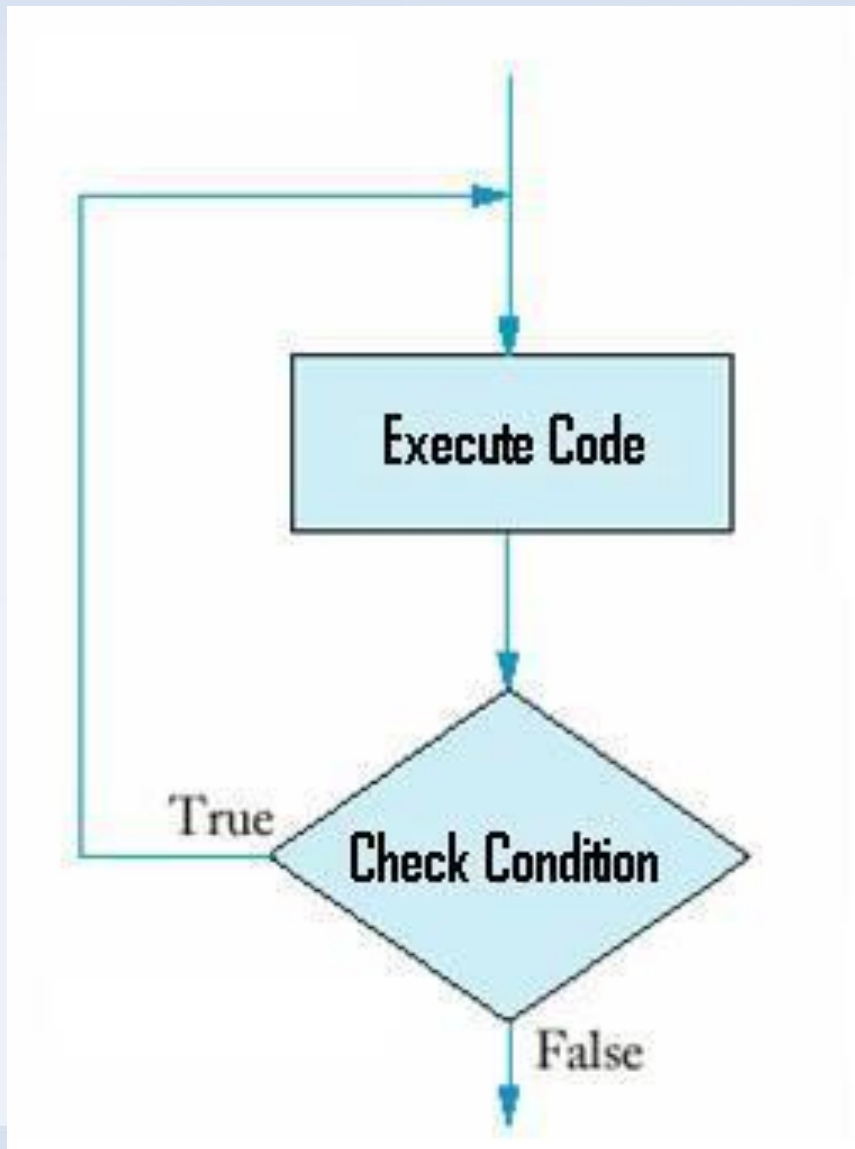
- do-while döngüsü aslında while döngüsü ile aynı şekilde çalışır.
- Tek fark, programın içerisinde eğer bir while döngüsü var ise bu döngüye girilip girilmeyeceği while komutu içerisindeki ifadenin true olup olmamasına bağlıdır.
- Eğer ifade true değerini döndürmüyorsa, döngüye girilmeyerek, döngü bloğu bitimindeki komutlardan program devam eder.

do-while Döngüsü

- Ancak, do-while döngü yapısında bu böyle değildir.
- Döngü, döngü koşulundaki değere bakılmadan mutlaka 1 kez çalıştırılır.
- Döngünün 2. veya daha fazla çalışıp çalışmayacağı, döngü koşuluna bakılarak eğer true dönüyor ise belirlenir.

```
do
{
    //döngü kodları
} while(ifade);
```

Do-while



Do-While - DEMO

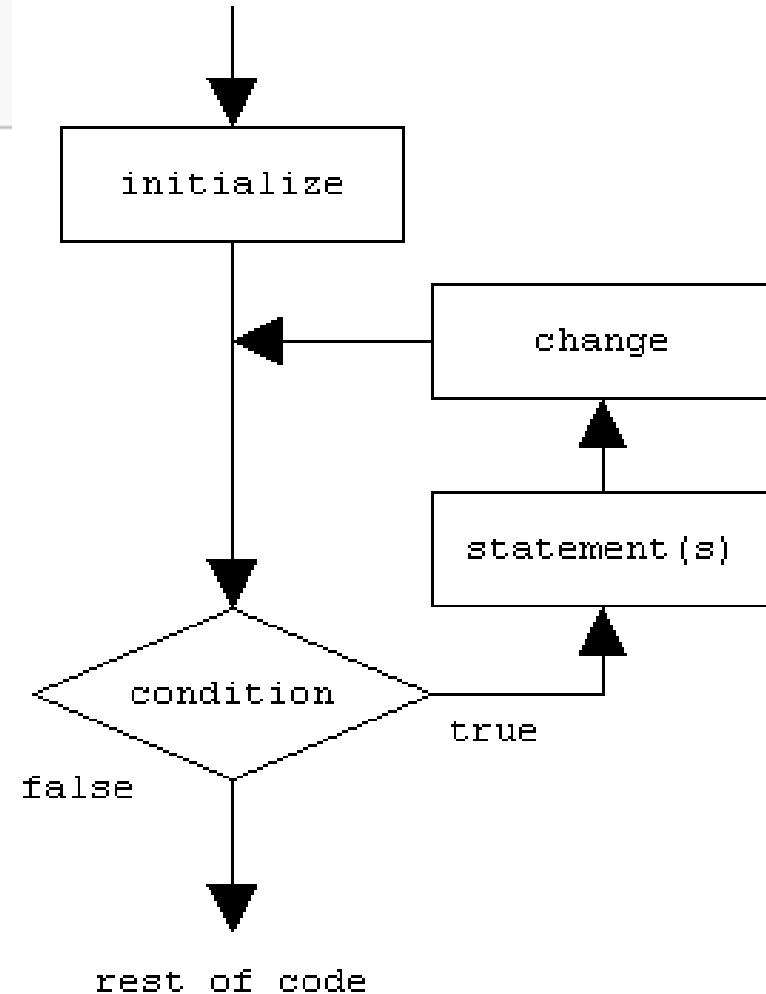
- *Parola giriş ekranı ile kullanıcıdan şifre alma*

for Döngüsü

- for döngüsü, while ve do-while döngülerine göre daha farklı bir kullanım alanına sahiptir.
- Programcılar, kendi belirledikleri koşul etrafında döngünün kabaca kaç defa döneceğine karar verirler.
- for döngüsü 3 farklı yapı ile oluşturulur.
- Bunlar, döngüyü kullanıma hazırlama (initialization), koşul (condition) ve arttırma (increment).
- Arttırma bölümünde, döngü ifadesini oluşturan değişkenler güncellenerek döngünün çalışma yapısı güncellenir.

for Döngüsü

```
for(initialization; condition; increment)
{
    //döngü kodları
}
```



döngüler

- while ve do-while döngüleri ile yapılan her türlü döngü işlemleri for ile de yapılabilir.
- for döngüsü döngüyü oluşturan ifadeleri tek bir satırda toplayarak, döngü yapısının okunmasını ve değişkenlerin güncellenmesi gibi komutların unutulmasında ortaya çıkabilecek hataların önüne geçebilmek için tasarlanan gelişmiş bir döngü yapısıdır.

For - DEMO

- Faktöryel
- Üslü sayı hesabı
- Fibonacci sayıları

?? SORULAR ??

TEŞEKKÜRLER