

MTK467 Nesneye Yönelik Programlama

Ders 1

Zümra Kavafoğlu

Dersin Amacı

Nesneye Yönelik Programlama Kavramını Java Dili Üzerinden Öğrenmek

Nesneye yönelik programlama ile ilgili nesne, sınıf, kalıtım, çok biçimlilik, kapsülleme, arayüzler gibi temel kavramları tanıtmak

Bu kavramları kullanarak Java programlama diliyle program yazma becerisi kazanmak

Puanlama

- 5 ödev projesi (% 30) - En yüksek puanlı 4 ödev
- 1 arasınay (%30)
- Final (%40)
- Derslere düzenli katılım beklenmektedir.

İletişim

- e-posta adresi : zdemir@hacettepe.edu.tr

Bu derste neler öğreneceğiz?

- Programlama Dilleri ve Seviyeleri
- Programlama Dillerinin Tarihçesi
- Tipik bir Java geliştirme Ortamı
- JRE ve JDK
- Örnek bir Java kodu
- IDE (Tümleşik Geliştirme Ortamı)

Programlama Dilleri ve Seviyeleri

- **Programlama dili**, bir makineye özellikle de bilgisayara bir takım talimatlar vermek için tasarlanmış yapay bir dildir.
- **Makine Dili**, işlemcinin verilen komutlar doğrultusunda çalıştırılmasını sağlayan ve işlemci mimarisine göre değişen en alt seviyedeki programlama dilidir. Bu dil sadece 0 Ve 1 ikililerinin anlamlı kombinasyonlarından meydana gelmektedir. Bu nedenle, makine dilinin anlaşılması çok güçtür.



Python, JavaScript,...

Java, C#,...

C++, C,...

Assembly

Makine Dili



Programlama Dilleri ve Seviyeleri

- **Assembly(Çevirme Dili)**, belirli bir bilgisayar mimarisine özelleşmiş düşük seviye bir programlama dilidir. Makine dilinden daha anlaşılırdır ama yüksek seviye dillere göre öğrenmesi ve yazması çok zordur.

Assembly

```
title    Hello World Program

dosseg
.model small
.stack 100h

.data
hello_message db 'Hello World!',0dh,0ah,'$'

.code
main proc
    mov     ax,@data
    mov     ds,ax

    mov     ah,9
    mov     dx,offset hello_message
    int     21h

    mov     ax,4C00h
    int     21h
main endp
end main
```

C

```
#include<stdio.h>

int main()
{
    printf("Hello World\n");
    return 0;
}
```



Python, JavaScript,...

Java, C#,...

C++, C,...

Assembly

Makine Dili



Programlama Dilleri ve Seviyeleri

- **Yüksek Seviye Diller**, gerçek dile(İngilizce) daha yakın komutlara sahip, bilindik matematik işlemlerini kapsayan, yazması ve öğrenmesi daha kolay programlama dilleridir.
- Günümüzde sıklıkla kullanılan C, C++, Java, Python gibi diller yüksek seviye programlama dilleridir.
- Bu dilleri makine diline çevirmek için derleyiciler kullanılır.



Python, JavaScript,...

Java, C#,...

C++, C,...

Assembly

Makine Dili



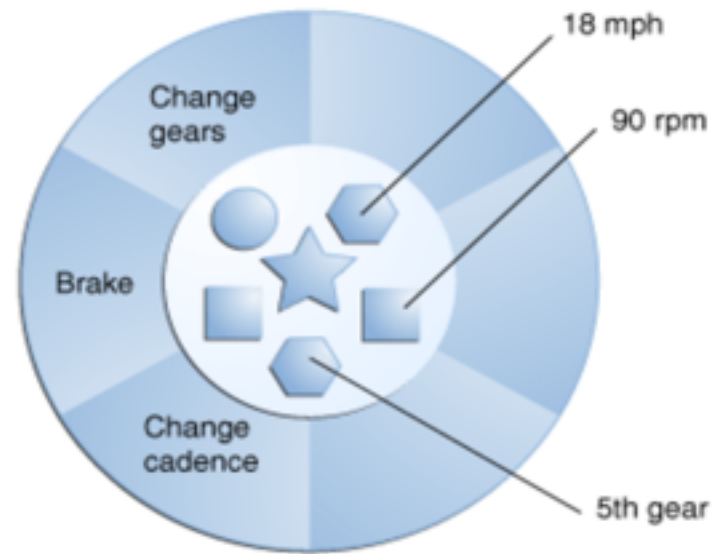
Yapısal ve Nesne Yönelimli Programlama Dili Farkları

Yapısal Programlama Dili	Nesne Yönelimli Programlama Dili
Program fonksiyon adı verilen küçük parçalara bölünmüştür.	Program nesne adı verilen küçük parçalara bölünmüştür.
Top-down yaklaşım	Bottom-up yaklaşım
Yeni veri ve fonksiyon eklemek zordur, diğer bir deyişle genişletilebilirliği kolay değildir.	Yeni veri ve fonksiyon eklemeyi kolaylaştıran bir yapıdadır.
Genellikle tüm veriler fonksiyonlar tarafından erişilebilirdir.	Veriler özel(private) veya genel(public) olarak tanımlanabilir ve böylece verinin erişilebilirliği kontrol altında tutulabilir.

Nesne nedir

- Yazılım nesneleri, gerçek hayattaki nesnelere benzer özellikler taşıyan yapılardır.
- Gerçek nesnelerin ortak iki özelliği vardır: bir duruma ve bir davranışa sahiptirler.
- Örneğin masa lambanızın iki farklı durumu vardır, “Açık” ve “Kapalı”, ve iki mümkün davranışı vardır “Aç” ve “Kapa”.
- Yazılım nesneleri de benzer şekilde durumlara ve davranışlara sahiptirler. Bir yazılım nesnesi durumunu değişkenler içinde tutarken, davranışını belirleyen metodlara sahiptir.
- Metodlar ait oldukları nesnenin durumu üzerinde işlem yapar ve nesneler arası iletişimde temel mekanizmayı oluşturur.
- Bir nesnenin iç durumunu saklamaya ve tüm iletişimleri metodlar aracılığıyla sağlamasına veri kapsüllemesi denir ve bu nesne yönelimli yazılımın temel prensiplerindendir.

Nesne nedir



A bicycle modeled as a software object.

- Örneğin bir bisikleti ele alalım. Bisikletin durumunu belirleyerek (hız, pedal ritmi ve vites) ve bu durumu değiştirecek metodlar tanımlayarak, dış dünyanın onu nasıl kullanabileceğinin kontrolünü elde etmiş oluruz. Örneğin bisiklet yalnızca 6 vitesse sahipse, vites değiştirmek için tanımlanmış bir metod 1'den az veya 6'dan büyük bir değeri kabul etmeyecektir.

Nesne nedir

- Nesnelerle kod yazmanın bir çok faydası vardır
 - Modülerlik: Bir nesnenin kodu, sistemdeki diğer nesnelerin kodundan bağımsız olarak yazılabilir ve değiştirilebilir. Bir kere oluşturulduğunda sistemin istenilen yerinde rahatça kullanılabilir.
 - Bilgi saklama: Bir nesneyle yalnızca metodları aracılığıyla ilişki kurularak, onun iç yapısı dış dünyadan gizli hale getirilebilir.
 - Kodun yeniden kullanılması: Eğer bir nesne halihazırda mevcutsa (başka bir yazılımcı tarafından yazılmış da olabilir), bu nesneyi kendi programınızda kullanabilirsiniz. Böylelikle uzman yazılımcıların yazdığı ve test ettiği karmaşık ve göreve-özel nesneleri kendi programınızda güvenle kullanabilirsiniz.
 - Ekleme ve hata ayıklama kolaylığı: Eğer herhangi bir nesne problem çıkarıyorsa onu uygulamanızdan çıkarabilir ve yerine yeni bir nesne ekleyebilirsiniz. Bu gerçek dünyadaki mekanik problemleri çözmeye benzerdir. Eğer bir civata bozulursa, bütün makineyi değil onu değiştirirsiniz.

C ve C++'ın tarihçesi

- **C**
 - Yapısal bir programlama dilidir.
 - 1972'de Unix işletim sistemi için geliştirilmeye başlanmıştır.
- **C++**
 - Nesne Yönelimli bir programlama dilidir.
 - 1979'da Danimarkalı Bilgisayar Bilimcisi Bjarne Stroustrup'un doktora tezi için "Sınıf içeren C" üzerine çalışmasıyla geliştirilmeye başlanmıştır.
 - 1985'de ilk ticari sürümü yayımlanmıştır.

Java'nın Tarihçesi

- **Java**

- Nesne Yönelimli bir programlama dilidir.
- 1991'de James Gosling, Mike Sheridan, ve Patrick Naughton tarafından ticari elektronik cihazlar için geliştirilmeye başlanmıştır.
- 1995'de ilk sürümü "Write Once, Run Everywhere" mottosuyla Sun Microsystems tarafından yayınlanmıştır.
- Yazılımcılara tanıdık gelmesi için C/C++ benzeri bir syntax stiliyle tasarlanmıştır.

Tipik bir Java Geliştirme Ortamı

- **Java Programları aşağıdaki 5 aşamadan oluşur**

- **Düzenleme**

- ▶ Programcı bir editör kullanarak kaynak kodu yazar ve kodun olduğu dosyayı .java uzantısıyla diske kaydeder.

- **Derleme**

- ▶ javac(Java Derleyicisi) kullanılarak kaynak kodundan bytecode üretilir, bytecode .class dosyasında saklanır.

- **Yükleme**

- ▶ Sınıf yükleyici .class dosyasından bytecodeları hafızaya okur.

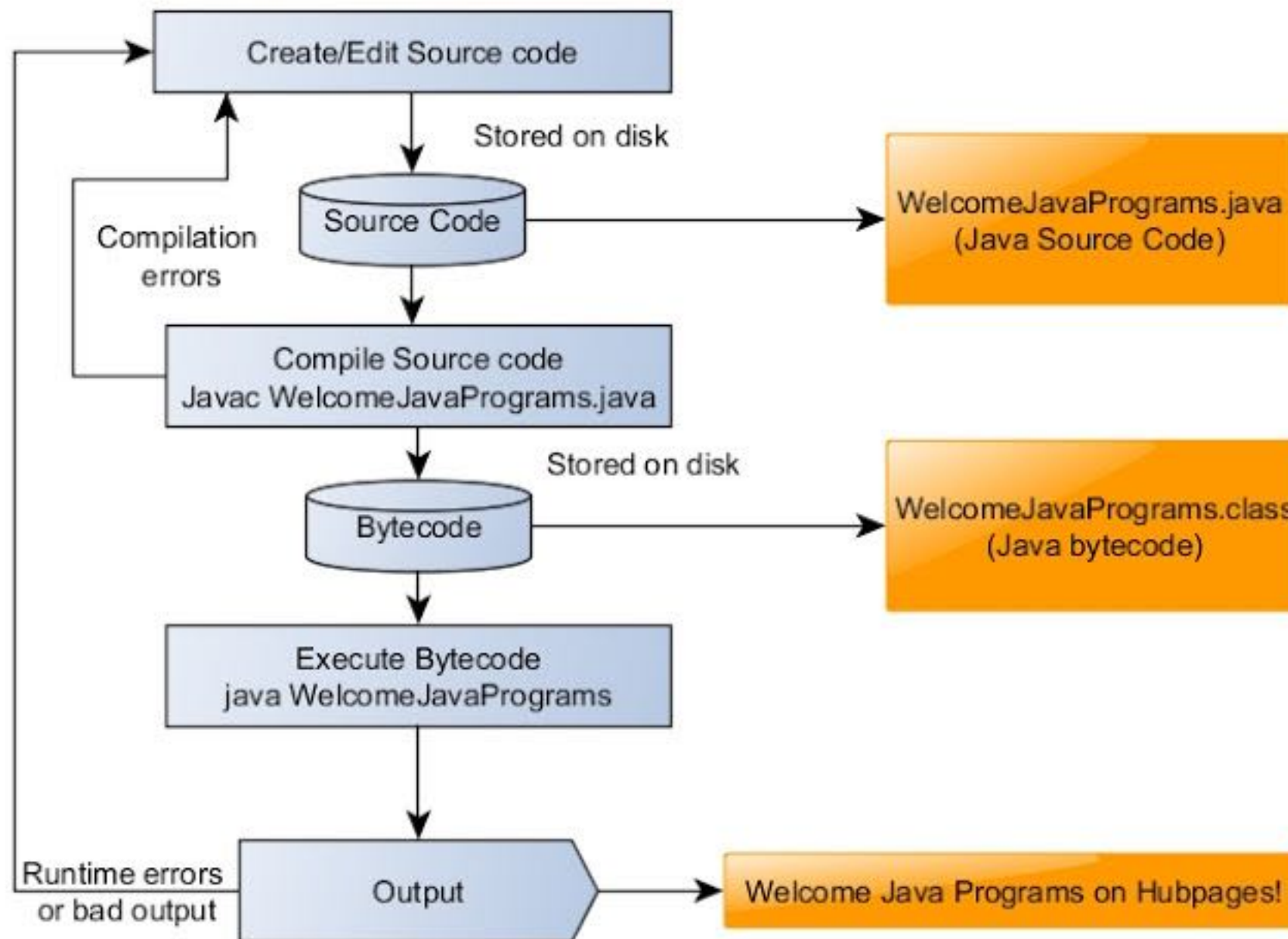
- **Doğrulama**

- ▶ Bytecode doğrulayıcı geçerli olduklarından ve güvenlik kısıtlamalarını ihlal etmediklerinden emin olmak için bytecodeları inceler.

- **Çalıştırma**

- ▶ Java Sanal Makinesi(Java Virtual Machine/JVM) bytecodeları makine diline çevirir.

Tipik bir Java geliştirme ortamı

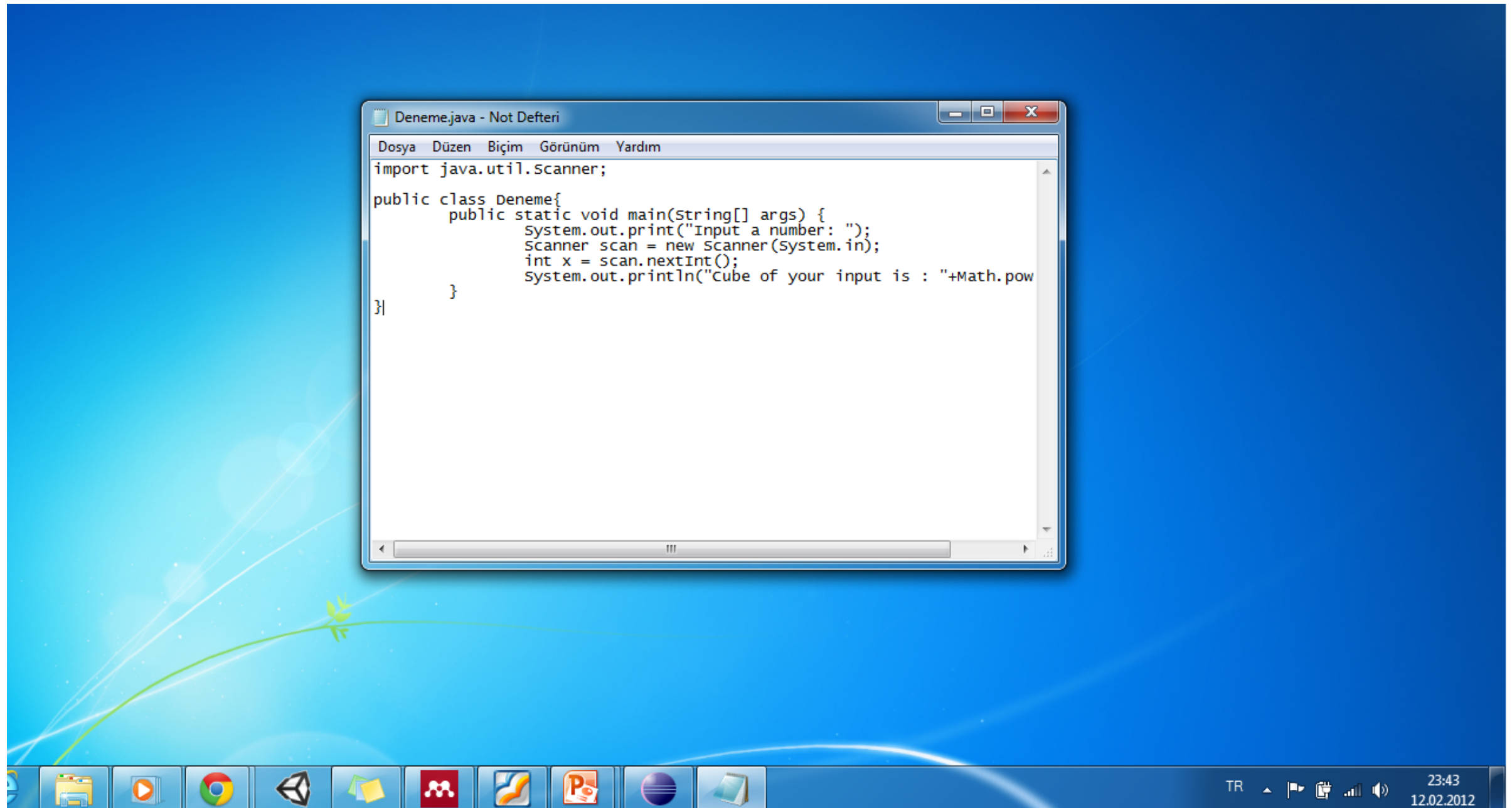


JRE ve JDK

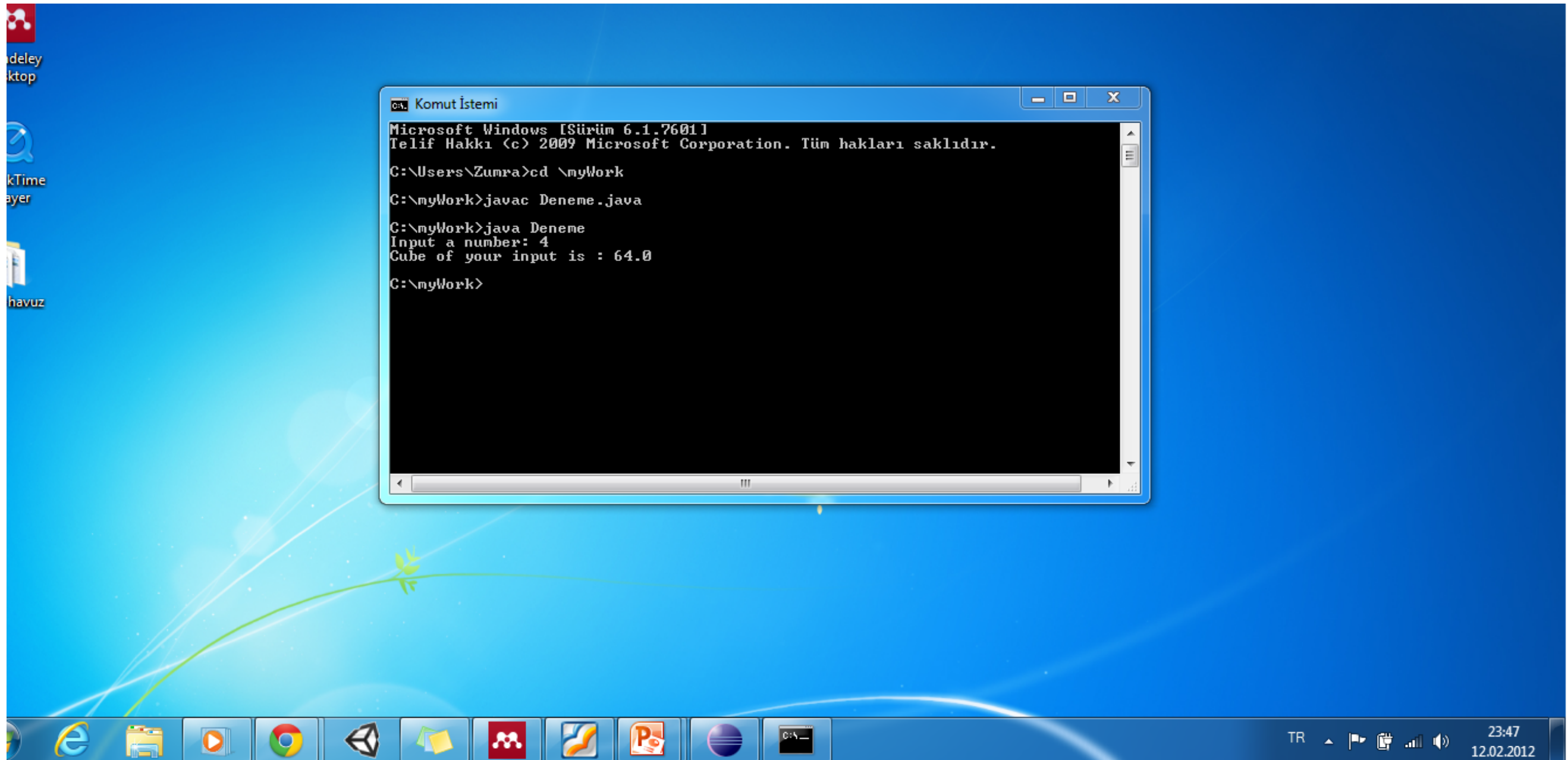
- **JRE (Java Runtime Environment)** Java ile yazılmış bir programın çalıştırılması için gerekli bileşenleri içeren bir yazılım paketidir (Bir Java Sanal Makinesi içerir)
- **JDK (Java Development Kit)** Java ile uygulama geliştirmek için gerekli araçları içeren bir yazılım paketidir. (javac derleyicisini içerir).
 - JRE, JDK'nın içinde yer alır.
 - Java ile uygulama geliştirmek için JDK kurulumu şarttır.
 - <http://www.oracle.com/technetwork/java/javase/downloads/index.html> adresinden indirip kurabilirsiniz.

Java kodu nasıl çalıştırılır?

Java kodunu çalıştırmak için bir metin editörü (örneğin notepad) ve komut penceresi yeterli.



Java kodu nasıl çalıştırılır?



IDE(Integrated Development Environment / Tümleşik Geliştirme Ortamı)

- **Tümleşik geliştirme ortamı (IDE)** , bilgisayar programcılarının hızlı ve rahat bir şekilde yazılım geliştirebilmesini sağlayan araçlara sahip bir yazılım uygulamasıdır.
- Bir IDE genel olarak,
 - Programlama diline göre sözdizimi renklendirmesi yapabilen kod yazım editörü ve
 - Tümleşik bir derleyici, yorumlayıcı ve hata ayıklayıcıya sahiptir, böylece kod yazım sürecini hızlandırır.
- Belli başlı Java ideleri
 - IntelliJ
 - Eclipse
 - Netbeans
 - JCreator
- **Bir sonraki derse kadar kişisel bilgisayarlarınıza JDK ve bir IDE kurunuz.**