






## Hedefler



Bu üniteyi çalıştıktan sonra;

	Java programlama dilinin yeteneklerini keşfedecek,
	Java teknolojileri hakkında bilgi sahibi olacak,
	Java ile program geliştirme aşamalarını öğrenecek,
	Yaygın olarak kullanılan Java editörlerini tanıyacaksınız.
	Eclipse editörünün kurulum ve program yazım aşamalarını öğreneceksiniz.

# İçindekiler

## JAVAYA GİRİŞ

1. Giriş
2. Java dilinin özellikleri
3. Java'yı nereden ve nasıl indirebilirim?
4. Java Teknolojileri ve geliştirme araçları
5. Java ile Program Geliştirme Aşamaları
6. Java Programlama Dili ve Gramer Yapısı
7. Not defteri ile Java Programı Yazma
8. Bir Java Editörü ile Java Programı Yazma
9. Konu Özeti
10. Değerlendirme Soruları



# JAVAYA GİRİŞ

## 1. GİRİŞ

Java, Sun Microsystems mühendislerinden **James Gosling** tarafından geliştirilmeye başlanmış gerçek nesneye yönelik, platformdan bağımsız, yüksek performanslı, çok işlevli, yüksek seviye, adım adım işletilen bir dildir.

İlk sürümü 1996 yılında çıkmış olup, 1.1, 1.2, 1.3, 1.4, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0, 11.0(LTS) ..., 22.0 gibi sürümleri geliştirilmiştir. Oracle firması LTS (uzun süre destekli) Java sürümlerini her 3 yılda bir diğer ara sürümlerini ise 6 ayda bir çıkarmaktadır.

Java, ilk çıktığında daha çok küçük cihazlarda kullanılmak için tasarlanmış ortak bir platform dili olarak düşünülmüştü. Ancak platform bağımsızlığı özelliği ve standart kütüphane desteği C ve C++'tan çok daha üstün ve güvenli bir yazılım geliştirme ve işletme ortamı sunduğundan, hemen her yerde kullanılmaya başlanmıştır. Şu anda özellikle kurumsal alanda ve mobil cihazlarda son derece popüler olan Java özellikle J2SE 1.4 ve 1.5 sürümü ile masaüstü uygulamalarda da yaygınlaşmaya başlamıştır. Java'nın ilk sürümü olan Java 1.0 (1995) Java Platform 1 olarak adlandırıldı ve tasarlama amacına uygun olarak küçük boyutlu ve kısıtlı özelliklere sahipti. Daha sonra platformun gücü gözlemlendi ve tasarımında büyük değişiklikler ve eklemeler yapıldı. Bu büyük değişikliklerden dolayı geliştirilen yeni platforma Java Platform 2 adı verildi ama sürüm numarası 2 yapılmadı, 1.2 olarak devam etti. 2004 sonbaharında çıkan Java 5 ve 8.0, önceki 1.2, 1.3 ve 1.4 sürümlerinin ardından en çok gelişme ve değişikliği üzerinde barındıran sürüm oldu. Java SE 11.0 sürümü ise Oracle firmasının ilk LTS sürümü olarak 2018 tarihinde piyasa çıkmıştır. [13 Kasım 2006](#) da Java platformu [GPL](#) lisansı ile açık kodlu hale gelmiştir. Sürüm detaylarına <https://www.java.com/releases/fullmatrix/#note1> adresinden erişebilirsiniz.



Sun Microsystems'in, **Oracle tarafından** 2009 tarihinde satın alınması ile birlikte Java, artık Oracle firması tarafından geliştirilmektedir.

## 2. JAVA PROGRAMLAMA DİLİNİN ÖZELLİKLERİ

Java programlama dili, tam bir nesne tabanlı ortam sunması, platform bağımsızlığı, java sanal makinesinin (JVM) yüklü olduğu her ortamda çalışması ve ücretsiz olması gibi sebeplerle günümüzde en popüler programlama dillerinden birisi haline gelmiştir. Java dilinin temel özelliklerini maddeler halinde sıralarsak;

<b>Basit (Simple)</b>	Java dili kısa ve özlü komut yapısına sahip, öğrenmesi kolay bir dildir. Ayrıca C,C++ diline benzerliği nedeni ile bu dillere aşina olanlar için Java'da kod yazmak kolay olacaktır.
<b>Güvenli (Secure)</b>	Java programlama dili ile hazırlanan programlar güvenliğin çok önemli olduğu uygulamalarda (internet uygulamaları gibi) tercih edilmektedir. C, C++ gibi dillerden daha güvenlidir. Sadece derleme esnasında değil aynı zamanda yükleme(çalıştırma) esnasında da Byte-code doğrulaması yapılır.
<b>Taşınabilirlik (Portable)</b>	Java teknolojisinin en önemli özelliği her ortamda çalışabilmesidir. JVM (Java Sanal Makinesi) nin kurulduğu her ortamda (Java platformunu destekleyen her türlü ortamda) çalışır.
<b>Nesne Yönelimli (Object Orient)</b>	Nesne yönelimli bir programlama dilidir. Java'da her şey ya nesnedir ya da bir nesnenin parçasıdır.
<b>Sağlam (Robust)</b>	Java sağlam bir dildir. Programlamadaki hataların çoğu daha yazılım aşamasında anlaşılabilir. Yazılım aşamasında yakalanamayan hatalar, çalışma zamanında (run-time) düzeltiler.
<b>Çoklu kullanım (Multithreaded)</b>	Java dili çoklu kullanım (multithreaded) programlama desteği sağlar. Çoklu kullanım, program içinde birden fazla işlemin çalışabilmesi demektir.
<b>Mimariden bağımsız (Architecture-neutral)</b>	Bir belirli makine ya da işletim sistemi mimarisine Java programlama dili bağlı değildir.

<b>Yorumlayıcı ( Interpreter )</b>	Java dili hem derleyici hem yorumlayıcı kullanır.
<b>Yüksek performans (High performance)</b>	Java diğer dillere nazaran nispeten yavaş olmakla birlikte, Java bytecode'u çalışma hızı için en iyi şekilde optimize edilmektedir.
<b>Dağıtık ( Distributed )</b>	'Dağıtık' birden fazla bilgisayarda çalışan programların bir biriyle uyumlu çalışabilmesidir. Bir yazılım parçasının bir kısmının bir makinede diğerinin başka makinede aynı anda çalışması mümkündür. (Özellikle internet uygulamalarında Java tercih edilmektedir.
<b>Dinamik (Dynamic )</b>	Java da bir programın birimlerinin (kütüphaneler, modüller veya sınıfların) birbirine bağlanması çalışma zamanında (run-time) yapılır. Kullanılan birimlerin içyapısı değiştirildiğinde, bu birimleri kullanan programın değişmesi gerekmez
<b>Ücretsiz</b>	Java programını ve her türlü yazılımını internet adresinden ücretsiz edinebilirsiniz.

Java, kendini çok hızlı bir şekilde yenileyen bir dil, sürekli yeni sürümleri çıkmakta ve her çıkan sürümle birlikte dile yeni özellikler eklenmektedir. Bu programcılar için hem dezavantaj hem de avantaj sağlamaktadır. Dezavantajı, programcının sürekli yeni özellikleri öğrenmek zorunda kalması ama bu dezavantajı java dili geriye uyumluluk ile avantaja dönüştürmektedir. Yani 1.4 sürümündeki bir komut, 1.6 sürümünde de çalışmaktadır. **Java'nın mevcut sürümleri ve aralarındaki bazı farkları aşağıdaki listede görebilirsiniz.**

- **1995: Version 1.0 :**
  - o 8 paket ile 212 sınıfa(class) sahip.
  - o İlk sürüm
- **1997: Version 1.1:**
  - o 23 paket ile 504 sınıfa (class) sahip.
  - o JVM iyileştirildi, iç sınıflar ve daha etkili hata yakalama içermektedir.
  - o Çoğu web tarayıcı 1.1 ile uyumlu hale geldi

- **1999: Version 1.2:**
  - o 59 paket ile 1520 sınıfa (class) saip.
  - o Java 2 Platformu olarak adlandırıldı.
  - o Eski sürümlere oranla 10 kat daha hızlıdır.
  - o API'de önemli değişiklikler, reflection özelliği ve [JIT Derleyicisi](#) nin eklenmesi ancak dilin kendisinde önemli değişiklikler olmadı.
- **2000: Version 1.3:**
  - o 76 paket ile 1842 sınıfa (class) saip.
  - o Hotspot JVM sunuldu.
- **2002: Version 1.4:**
  - o 135 paket ile 2991 sınıfa (class) saip.
  - o XML desteği ve pek çok yeni API değişikliği
- **2004: Version 5.0 (1.5 sürümü):**
  - o 165 paket ile 3000 den fazla sınıfa (class) sahip.
  - o Grafik alt yapısında ve sanal makine başarımında gelişmeler oldu.
  - o For- each benzeri döngü yapısı, yeni komutların eklenmesi
- **2006: Version 6.0 (1.6 sürümü):**
  - o Java 1.5 göre daha fazla paket ve sınıfa sahip.
  - o Derleyici erişimi, program içerisinde javac derleyicisi kontrol edilebiliyor.
  - o Daha fazla güvenlik desteği
  - o Yeni masaüstü özellikleri ve pek çok yeni API değişikliği
- **2009: Version 7.0 (1.7 sürümü):**
  - o Ağ bağlantılarına yüksek performansta erişim için SDP- Sockets Direct Protocol geliştirildi.
  - o Dosya giriş/çıkış paketi yenilendi (java.nio.file paketi geliştirildi)

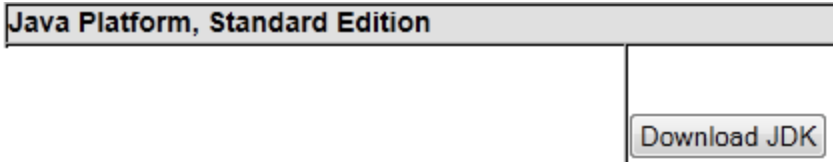
Çok dinamik bir şekilde kendini yenilen/güncelleyen Java programlama diline en son eklenen özellikleri aşağıdaki linkten takip edebilirsiniz.

<http://download.oracle.com/javase/tutorial/information/history.html>

### 3. JAVA Programını Nereden ve Nasıl İndirebilirim?

Bir Java programını çalıştırabilmek için Oracle web adresinden

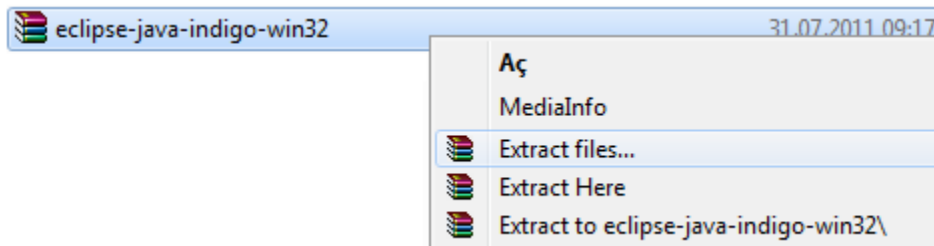
<http://www.oracle.com/technetwork/java/javase/downloads/index.html>



en son **JDK** sürümünü indirebilirsiniz (download edebilirsiniz).

Java programlarını kodlayıp, geliştirebilmek için **JDK** (Java Geliştirme Kiti) ne ihtiyaç vardır. JDK içerisinde Java derleyicisi “**javac.exe**”, derlenmiş programı çalıştıracak olan “**java.exe**”, applet uygulamalarını test etmemizi sağlayan “**appletviewer.exe**” gibi dosyaları/programları ve java sanal makinesini (JVM) barındırır. Ayrıca JDK içerisinde **JRE**(Java Çalıştırma ortamı)’nı barındırdığı için ayrıca indirmenize **gerek yoktur**.

JDK kurulumunu tamamladıktan sonra [www.eclipse.org/downloads/](http://www.eclipse.org/downloads/) adresinden program yazımında kullanacağımız Eclipse java editörünü yani “**eclipse**” programını bilgisayarımıza indiriyoruz. İndirdiğimiz zipli dosyanın üzerinde iken sağ tık yaparak bilgisayarımızda herhangi bir yere açıyoruz(extract ediyoruz). Artık Eclipse programını çalıştırabiliriz.



### 4. JAVA TEKNOLOJİLERİ VE GELİŞTİRME ORTAMLARI

Java teknolojisi, esnekliği, verimliliği, platformlar arasında taşınabilmesi ve güvenliği sayesinde günümüzde milyarlarca cihaz / aygıtta ( Bilgisayar, Internet, cep telefonu, akıllı kart, kod çözücü, yazıcı, web kameraları, yol bilgisayarları, tıbbi cihazlar v.s. ) kullanılmaktadır. Bugün Java teknolojisini dizüstü bilgisayarlardan veri merkezlerine, oyun konsollarından bilimsel amaçlı süper bilgisayarlara, cep telefonlarından Internet'e kadar her yerde bulabilirsiniz. Java'ya yeni başlayanlar için bazı terim ve kısaltmaların açıklamalarını yapmakta fayda var.



**JDK(Java Development Kit – Java Geliştirme Aracı):** Java geliştirme kiti olarak Türkçeye çevirebileceğimiz JDK, içerisinde hem java derleyicisini (javac) hem de java yorumlayıcısını (JVM) barındıran bir pakettir. **Bu paket, java editorü hariç Java ile geliştirme yapmak için bütün araçları içerir.** JDK;

*“Java Derleyicisi (javac) + JVM + Java Sınıf Paketleri ( util, math, lang, awt,swing v.b) + çalışma zamanı kütüphaneleri”*

Gibi birçok paket içerir.

**JRE( Java Runtime Engine- Java Çalışma Zamanı Motoru):** JRE, java uygulamalarını çalıştırabilmek için sistemde yüklü olması gereken temel bir pakettir. Bu paketle birlikte java uygulamalarını çalıştırabilirsiniz. JRE ise;

*“JVM + Java Sınıf Paketleri ( util, math, lang, awt, swing v.b) + çalışma zamanı kütüphaneleri”,*

gibi paketleri içerir.

**JRE,** bilgisayar kullanıcılarına, **JDK** ise yazılım geliştiricilerine hitap etmektedir. JDK kurulduğunda JRE'de kurulmuş olur, ayrıca kurulum yapmanıza gerek yoktur.

**JVM (Java Virtual Machine- Java Sanal Makinesi):** JVM'nin sanal makine diline **bytecode** denmektedir. Bütün bilgisayarlar Java bytecode için yorumlayıcıya (interpreter) ihtiyaç duyarlar. JVM işletim sistemi veya makinenin donanımsal mimarisinden bağımsızdır. Bir java uygulamasının çalışabilmesi için o makinede JVM nin yüklü olması gerekir. Zaten çoğu web tarayıcı programı (Internet Explorer, Netscape, Firefox, gibi) içerisinde JVM barındırır.

*Java'nın farklı platformlarda çalışan birçok geliştirme aracı vardır. Bunlardan bazıları;*

**Java ME(Java Micro Edition):** Mobil uygulamalar (cep telefonu, PDA gibi sınırlı kaynaklara sahip cihazlar üzerinde çalışan) geliştirmek için kullanılan platformdur. J2ME olarak da isimlendirilmektedir.

**Java SE(Standart Edition):** Programcıların java uygulamaları geliştirmek için kullandığı genel platformdur. Applet ve masaüstü uygulamalar geliştirmek için kullanılır. J2SE olarak da isimlendirilmektedir.

**Java EE (Enterprise Edition):** Standart Edition (J2SE) 'den çok daha fazla kütüphaneye sahiptir. Özellikle dağıtık, sunucu tabanlı ve çok katmanlı web uygulamaları geliştirmek için kullanılan bir platformdur. J2EE olarak da isimlendirilmektedir.

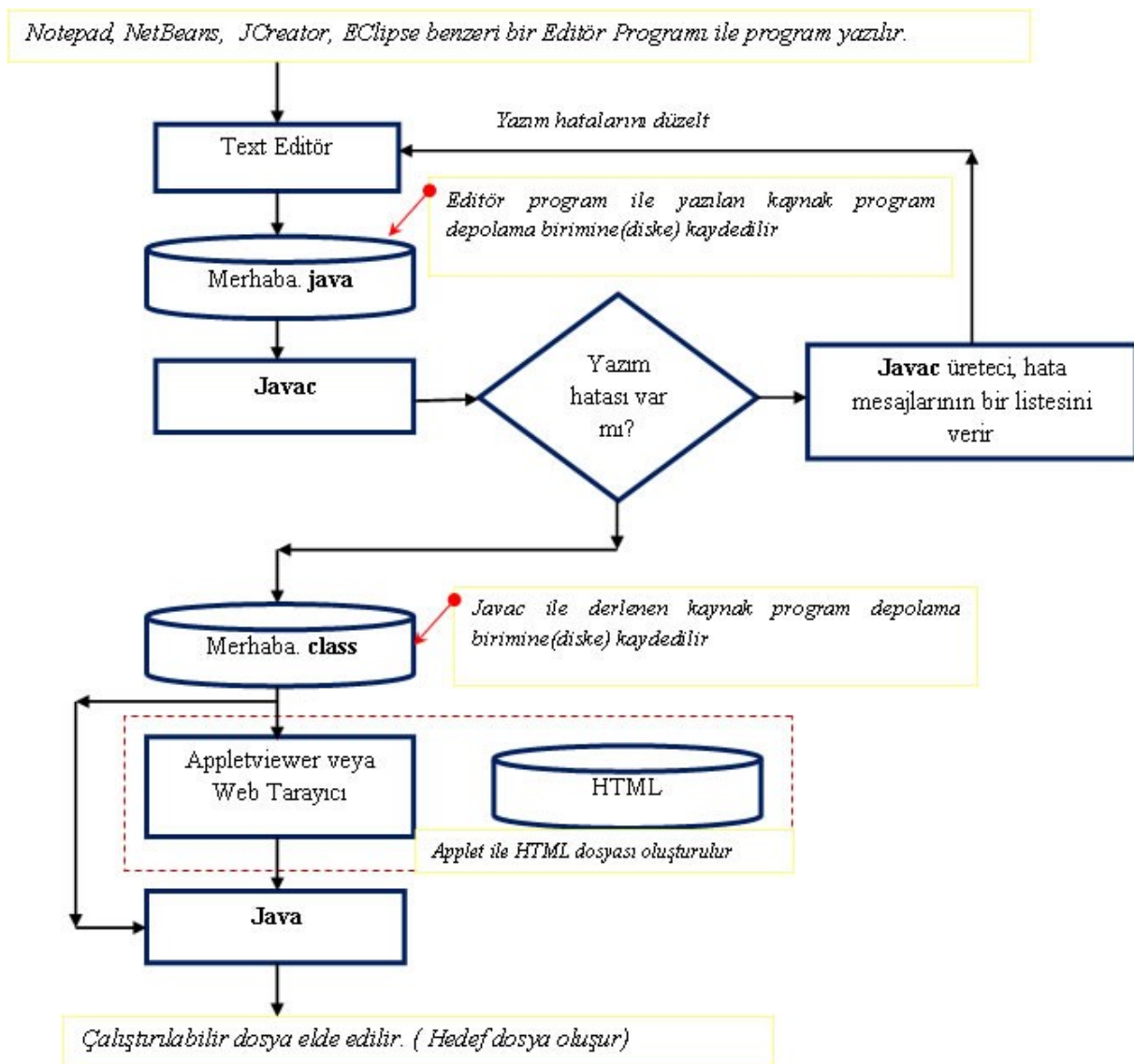
**Java FX:** Java'nın 2007 yılından itibaren geliştirdiği, zengin internet uygulamaları (Rich Internet Applications - RIA) geliştirmek için kullanılan bir script dilidir. JavaFX, script dilinin yanında bu dil kullanılarak geliştirilmiş bileşenler ve yine bu bileşenleri destekleyen kütüphaneleri içeren yeni java teknolojisidir. Java FX ile masaüstü, web tarayıcı ve mobil uygulamalar geliştirilebilir. Yazılım geliştiricilerin zengin içerikli, etkileşimli ve etkileyici uygulamalara Web tabanlı vektör grafiği, animasyon, ses ve video gibi öğeler yerleştirmelerine imkân verir.

**Java Card:** Java Card teknolojisi ile akıllı kartlar ve güvenlik noktaları için açık kaynaklı ve birlikte çalışabilir bir platform sunulmaktadır. Java Card platformu programcılara, ağ yönelimli yeni özelliklerin yanında yeni bir çalışma ortamı ve sanal makine sunmaktadır.

**Java TV:** Oracle firması ile birçok sayısal TV üreticisi bir araya gelerek Java TV API'yi geliştirdiler. Java TV, TV üzerinde çalışabilecek uygulamalar geliştirmek için kullanılan bir platformdur. Doğal olarak Java TV API'yi destekleyen üreticiler ürettikleri TV'lerin içine JVM yi yerleştirmektedirler.

## 5. JAVA PROGRAMLAMA DİLİNİN YAPISI

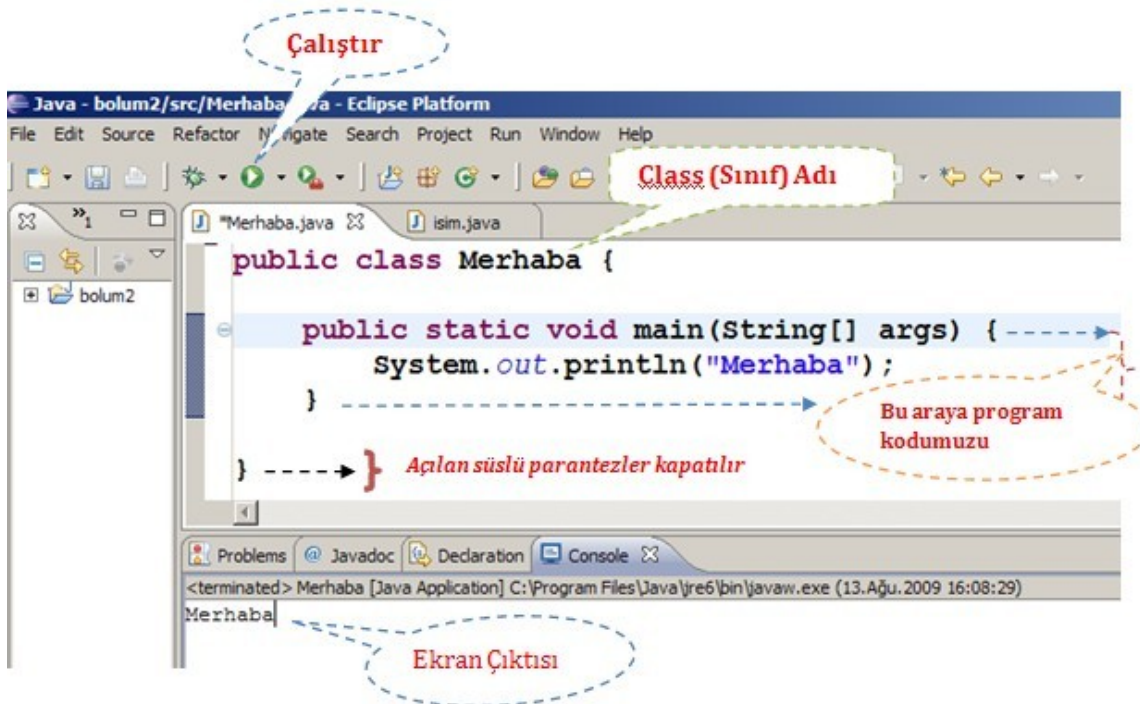
Aşağıdaki akış şeması Java programlama dilinde yazılan bir programın derleme aşamalarını göstermektedir. Dikkat edilirse bir editör programı veya not defteri ile yazılan Java programı **.java** uzantılı olarak kaydedilir, **javac** (java derleyicisi) ile **.class** uzantısına dönüştürülür ve java.exe ile çalıştırılır.



Her programlama dilinde olduğu gibi Java programlama dilinde de komutlar, metotlar, sabitler, mesajlar, değişkenler, operatörler ve açıklama satırları vardır. Java programlama dilinin yapısını basit bir program (Merhaba.java) üzerinde aşağıdaki şekilde açıklayabiliriz.

Ekrana “Merhaba” yazan örnek program kodu:

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



## 5.1. Java Dili Gramer Yapısı

Java programlama dili C/C++ dilinin gramer yapısını kullanır. Aynı C dilinde olduğu gibi her komut satırı ‘;’ karakteri ile sonlandırılmalıdır. Bütün komutlar küçük harflerle yazılmalıdır.



**Dikkat :** class deyimini Class şeklinde yazdığınızda program hata verir.

Açıklama (Yorum) satırları içinde;

- Eğer tek satırlık bir açıklama yapılacaksa ‘//’ karakteri,

*// Bu karakterlerden sonra yazılanlar ekranda görünmez*

- Birden fazla satırlık açıklama yapılacaksa ‘ /\* ..... \*/ ’ karakterleri kullanılır.

*/\* Bu karakterlerden sonra yazılanlar ekranda görünmez ve*

*Derleyici tarafından dikkate alınmazlar \*/*

## 5.2. Java ile Program Geliştirme Araçları

Bir Java programını (örneğinizde **Merhaba.java** isimli) ya Java dili için özel olarak geliştirilen bir **Java editör** programı yada bilgisayarınızda zaten kurulu olan “**not defteri**” programı ile yazabiliriz. Java editör programı kullanırsanız, kodu yazarken yazım hatalarınızı görebilir ve aynı platform üzerinde kodu çalıştırıp, sonuçları test edebilirsiniz. Not defteri ile yazdığınız java uzantılı programı çalıştırmak için ise komut istemini kullanabilirsiniz. Yazım hatalarınızı ise ancak kodu derledikten sonra görebilirsiniz.

Tabi burada şunu unutmayalım. Gerek not defteri gerekse bir java editörü ile kod yazalım öncesine mutlaka [JDK programını](#) bilgisayarımıza kurmamız gerekmektedir.

### 5.2.1. Not Defteri ile Java programı yazıp çalıştırma aşamaları

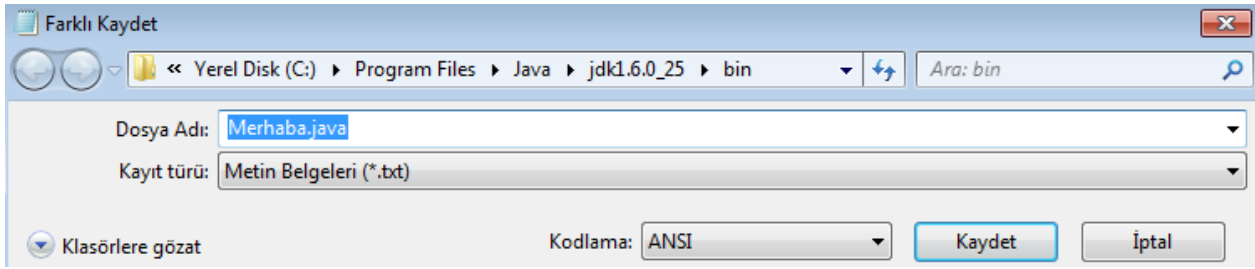
1. Bir metin editörünü (Not defterini) kullanarak ilk Java programımızı aşağıdaki gibi yazalım.


```
class Merhaba {
    static public void main( String args[] ) {
        System.out.println( "Merhaba bu ilk Java Programim!" );
    }
}
```

1. Yazdığımız programı **javac** derleyicisinin bulunduğu klasör (JDK programının kurulduğu klasör) içerisine “**.java**” uzantılı olarak kaydediyoruz. Bunun için izleyeceğimiz yol;

**Dosya → Farklı Kaydet → “C:\Program Files \ Java\jdk1.6.0\_25\bin”**

Klasörünü seçiyoruz ve **Merhaba.java** ismiyle kaydediyoruz.



2. MS-DOS komut istemine geçiyoruz (Windows kullanıcıları için, **Klavyedeki Windows**  simgesi + R tuşu ile “**Çalıştır**” menüsünden **CMD** yazarak geçebiliriz.) Programı derlemek için **javac** isimli derleyicinin bulunduğu yolu yazıyoruz. (Örnek olarak “javac” ın bulunduğu “C:\Program Files \ Java\jdk1.6.0\_25\bin” klasörünün içerisine **CD klasör\_ismi** konutu ile aşağıdaki gibi geçiyoruz)

Sonrasında ise CD komutları ile **javac** dosyasının bulunduğu klasöre geçilir.

```
C:\>cd program files
C:\Program Files>cd java
```

```
C:\Program Files\Java>cd jdk1.6.0_25
C:\Program Files\Java\jdk1.6.0_25>cd bin
C:\Program Files\Java\jdk1.6.0_25\bin>
```

3. MS-DOS komut ortamında **javac Merhaba.java** komut satırı ile **java** uzantılı programı derliyoruz.

```
C:\Program Files\Java\jdk1.6.0_25\bin>javac Merhaba.java
C:\Program Files\Java\jdk1.6.0_25\bin>
```

Programda bir yazım hatası yoksa derlenecektir. Bu aşamada “**Merhaba.class**” dosyası oluşmuş olur.

4.adım: Derlenmiş olan program dosyasını (class uzantılı dosya) **java** komutu ile (**java Merhaba** ) çalıştırabiliriz.

```
C:\Program Files\Java\jdk1.6.0_25\bin>java Merhaba
Merhaba bu ilk Java Programım!
C:\Program Files\Java\jdk1.6.0_25\bin>
```

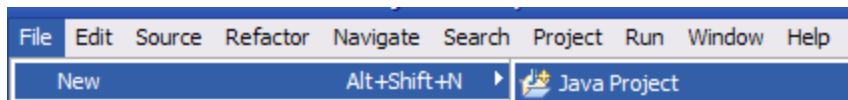
Programı çalıştırdığımızda ekrana “**Merhaba, bu ilk JAVA Programım!**” mesajını aldık.

## 5.2.2. Bir Java Editör programı ile program geliştirme

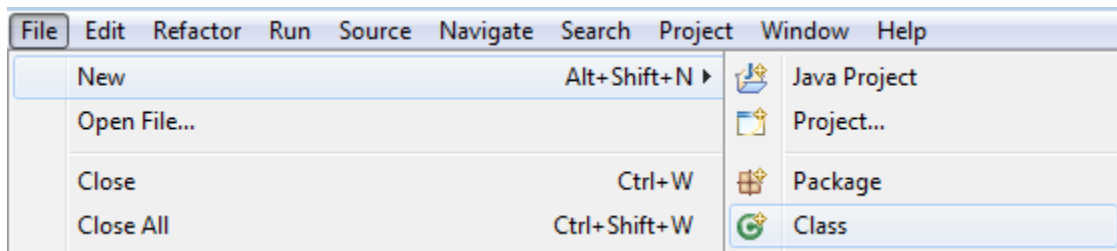
Eğer MS-DOS komut istemi ile uğraşmak istemiyorsanız, yazdığım kod direkt çalışsın diyorsanız, bir Java editörü kullanmanızı tavsiye ederiz. [Sık kullanılan bazı Java editörleri ve ücretsiz indirebileceğiniz web adresleri aşağıda verilmiştir.](#)

- Eclipse (<http://www.eclipse.org/downloads>)
- jGRASP (<http://www.jgrasp.org/>)
- NetBeans (<http://www.netbeans.info/downloads>)
- JCreator (<http://www.jcreator.com/download.htm>)
- BlueJ (<http://www.bluej.org/download/download.html>)
- Borland JBuilder (<http://cc.codegear.com/free/jbuilder>)
- Oracle JDeveloper ( <http://www.oracle.com> )

1. **Adım:** Bir Java editör programını örneğin Eclipse programının son sürümünü yukarıda belirtilen web adresinden indirdik ve bilgisayarımıza kurduk.
2. **Adım:** “File - - > New - - > Java Project” menüsünden bir proje klasörü oluşturduk. (Örneğin **JavayaGiris** isimli)



3. **Adım:** Oluşturulan proje klasörüne sağ tuş yapılır ve “New - - > Class” menülerinden



yeni bir sınıf adı seçilerek “Name” kısmına programımızın adı (“Merhaba”) yazılır.

**New Java Class**

The use of the default package is discouraged.

Source folder: JavayaGiris/src **Browse...**

Package: (default) **Browse...**

☐ Enclosing type: **Class isminin yazıldığı yer** **Browse...**

**Name:** Merhaba

Modifiers: ☒ public ☐ default ☐ private ☐ protected  
☐ abstract ☐ final ☐ static

Superclass: java.lang.Object **Browse...**

Interfaces: **Add...**  
**Remove**

Which method stubs would you like to create?

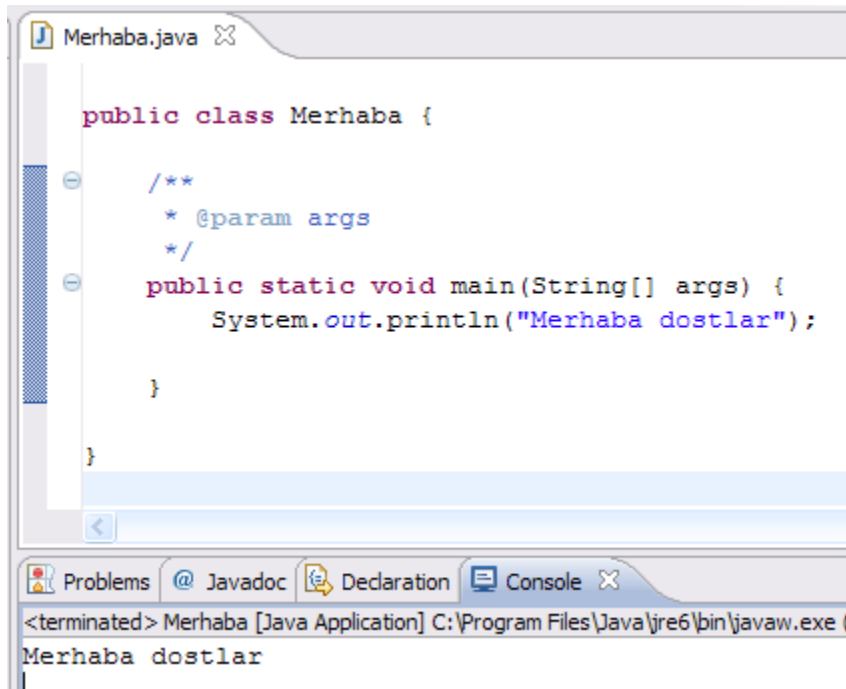
☒ public static void main(String[] args)  
☐ Constructors from superclass  
☒ Inherited abstract methods

Do you want to add comments? (Configure templates and default value [here](#))  
☐ Generate comments

**Finish** **Cancel**

**4. adım:** Programı yazıp, **Run** menüsünden “Run (Çalıştır)” veya F5 tuşu ile çalıştırıyoruz.





### 5.2.3. Applet ile bir Java Programı Geliştirme (Not. Bu başlık Java 8.0 ve öncesi sürümler için geçerlidir.)

Bir Java appleti, **appletviewer** programı veya html (web) dosyası ile çalıştırılabilir. Appletler sayesinde görsel araçlar (buton, label, gibi) kullanılabilir. Yazılan java kodları derlenerek önce **.class** dosyasına dönüştürülür, daha sonrada bir **html** dosyası içinde kullanıcıya sunulur.

Java applet program kodunun başlangıcında iki önemli paketi **import** etmek (çağırarak) gerekir. Grafiksel ara yüz için **java.awt** ve Applet sınıfı için gerekli olan **java.applet** paketini çağırmanız gerekir.

**Merhaba2.java** dosyasını aşağıdaki gibi oluşturup derliyoruz ( **Merhaba2.class** dosyası haline dönüştürüyoruz.

```

import java.awt.*;

public class Merhaba extends java.applet.Applet {

    public void paint(Graphics g)
    {
        g.drawString ("Merhabalar, bu ilk JAVA Programim", 10, 50);

    }

}

```

Daha sonra HTML tagları arasına applet kodumuzu yerleştiriyoruz. Bunun için Not defterini açıp aşağıdaki kodu yazalım ve

```
<html>
```

```
<applet code="Merhaba.class" width=450 height=150>
```

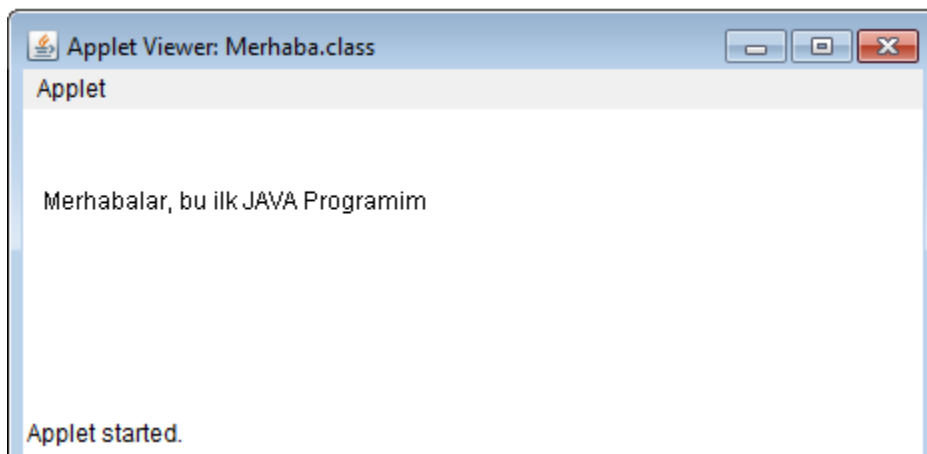
```
</applet>
```

```
</html>
```

Bu dosyamızı **Merhaba.html** olarak kaydedip, MS-DOS komut satırında **appletviewer** komutu ile

```
C:\Program Files\Java\jdk1.6.0_25\bin>appletviewer Merhaba.html
```

çalıştırırız. Aşağıdaki ekran görüntüsünü elde ederiz.



Eclipse editöründe bir applet kodunu yazdığımızda otomatik olarak çalıştırma seçeneklerinde **appletviewer** seçeneği çıkar.

## 6. Çoktan Seçmeli Değerlendirme Soruları

1. Bir java editörü ile yazdığımız kaynak dosyamızı (.java uzantılı) derlediğimizde (javac ile) dosya uzantımız ne olur?  
a) javac                      **c) class**  
b) j                              d) exe
2. Java programlama dili hangi dilin gramer yapısını kullanır?  
**a) C**                              b) Basic  
c) Pascal                      d) Linux
3. Aşağıdakilerden hangisi bir Java editörü değildir?  
a) Eclipse b) NetBeans c) Not Defteri **d) Paint**
4. Hangisi Java dilinde açıklama (Yorum) satırları için kullanılan bir operatördür?  
a) ?                              b) \*                              **c) //**                              d)%
5. Aşağıdakilerden hangisi Java dilinin özelliklerindendir?  
a) Ücretsiz olması                      c) Dağıtık uygulamalar geliştirilebilmesi  
b) Nesne yönelimli olması                      **d) Hepsi**
6. Java dili günümüzde hangi firma tarafından geliştirilmektedir?  
a) Microchip                      b) Microsoft                      **c) Oracle**                      d) Sun System
7. Java programlama dili ile ilgili olarak hangisi söylenebilir?  
a) Mobil uygulamalar geliştirilebilir.  
b) Masaüstü uygulamalar geliştirilebilir.  
c) Grafiksel / Görsel uygulamalar geliştirilebilir.  
**d) Hepsi**

## 7. Pratik Alıştırmalar



- **Alıştırma 1:** Editorü açınız ve ekrana isminizi yazan bir programı yazıp “isim.java” olarak kaydediniz. {class ismi de dosya ismi ile aynı olsun}
- **Alıştırma 2:** “isim.java” dosyanızı derleyip, çalıştırınız (MSDOS ortamında)
- **Alıştırma 3:** “isim.java” dosyanızdaki class ismini “ad” olarak değiştiriniz. Ve isim.java dosyanızı derleyip, çalıştırınız (MS-DOS ortamında), Derleme aşamasında Alıştırma-2 den farkını belirtiniz.?
- **Alıştırma 4:** “isim.java” programınızda bir yazım hatası yapınız (örneğin satır sonundaki ; kaldırın) ve program yeniden derleyip çalıştırınız, aldığınız hata mesajını yazınız.

## Kaynaklar

1. [“Bülent Çobanoğlu”, Java ile Programlama ve Veri Yapıları, 4. Baskı, ISBN:978-9944-711-10-4, Sayfa 5-21, 377-394](#)
2. [“Bülent Çobanoğlu”, Yeni Başlayanlar için Java, 3. Baskı, Abaküs yayınları](#)