

# NESNE YÖNELİMLİ PROGRAMLAMA(Object Oriented Programming/OOP)

**Öğr. Gör. Celil ÖZTÜRK**

Marmara Üniversitesi

Teknik Bilimler Meslek Yüksekokulu

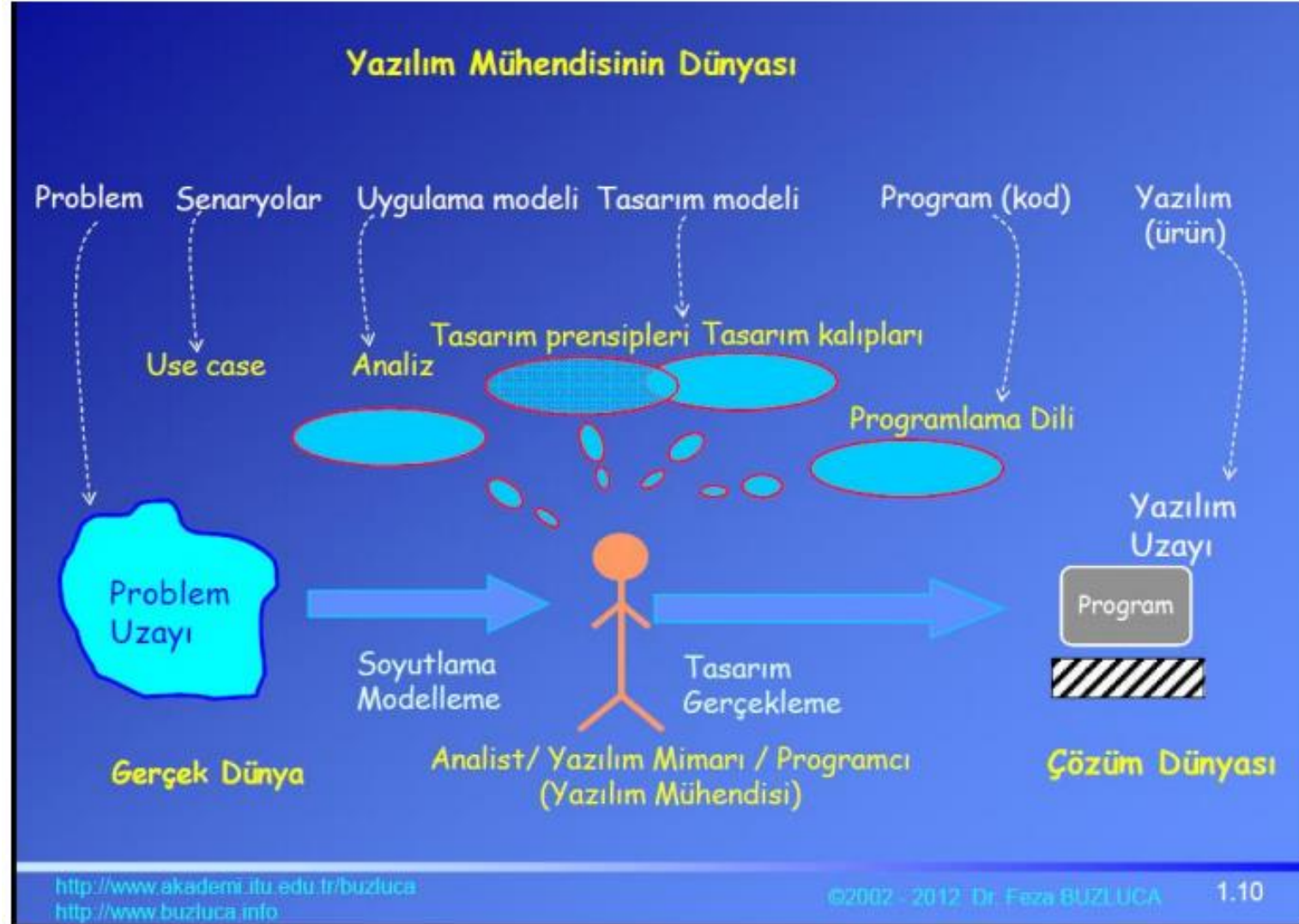
# İçerik

- ✓ Dersin Hedefi
- ✓ Yazılım Geliştirme Süreci
- ✓ Nesne Yönelimli Programlama Kavramı
- ✓ Java Dili Temel Bilgiler
- ✓ Java Geliştirme Ortamları


# Dersin Hedefi

- Nesne yönelimli programlama kavram ve yöntemlerini tanır.
- Java ile Kaliteli ve profesyonel yazılım üretmek için gereken metodolojileri öğrenir.
- Bu dersi başarıyla tamamlayanlar, nesne yönelimli programlama mantığı ile bir problemi modelleme, nesne tabanlı programlamayı Java dili ile kullanabilme yeteneklerini kazanacaklardır.

# Yazılım Geliştirme Süreci



# Yazılım Geliştirme Süreci/Yazılım Geliştirme Etkinlikleri

- Problem Tanımlama
  - Gereksinim Analizi
  - Planlama
  - Mimari (Üst-düzey) Tasarım
  - Ayrıntılı Tasarım
  - Kodlama ve Hata Ayıklama
  - Birim Testi
  - Tümlleştirme Testi
  - Sistem Testi
  - İşletme
  - Bakım
- 
- Implementation

# Yazılım Geliştirme Süreci/Yazılım Geliştirmenin Zorlukları

- Karmaşıklık (Complexity)
- Uzun Ömür ve Evrim (Longevity and Evolution)
  - Yazılımların yaşayan varlıklar olması ve sürekli gelişmesi gerekliliği.
  - Gereksinimler yazılım geliştiricilerinin kontrolünün dışındadır
- Yüksek Kullanıcı Beklentileri (High User Expectations)
  - Büyük ölçekli (large-scale) yazılım sistemlerinde son derece fazla ayrıntı vardır.

# Yazılım Geliştirme Süreci/Yazılım Geliştirmenin Zorlukları

## Karmaşıklığı Yönetmek ve Azaltmak

-Karmaşıklığın Yönetimi, yazılım geliştirmedeki **en önemli teknik konudur.**

-Komplike ve tek parça yerine basit parçalar halindeki bilgiyi anlamak daha kolaydır.

- ❖ Basit ve Sade (Simple)
- ❖ Okunaklı (Readable)
- ❖ Anlaşılabilir (Understandable)
- ❖ Bakımı kolay (yeni özellik eklemek, hata ayıklamak vs.) (Easy to maintain (adding feature, fixing bug, etc.))

-Tüm yazılım tasarım tekniklerinin temel hedefi komplike bir problemi basit parçalara ayırmaktır.

# Yazılım Geliştirme Süreci/Yazılım Geliştirmenin Zorlukları

## Basit ve Sade Kod Yazmak

- ❑ Kodu anlamlı **küçük** bloklara ayırın (packages, components, classes, methods, functions, routines, etc.)
- ❑ **Kod tekrarından kaçının**
- ❑ Sürekli yeniden düzenleyin (Refactor continuously)
- ❑ Mümkün olan en az sayıda sınıf ve metod oluşturun
- ❑ Amacı olmayan herşeyi silin
- ❑ **Kod tekrarından kaçının. Yine!**
- ❑ Derin kalıtım hiyerarşilerinden uzak durun
- ❑ Çok derin iç içe döngü ve şart ifadelerinden uzak durun



# Yazılım Geliştirme Süreci/Yazılım Geliştirmenin Zorlukları



(1)



(2)

## *Temel amaç*

karmaşıklığı çözebilmek,  
değişimlerle baş edebilmek,  
*esnek ve tekrar*  
*kullanılabilir* yazılım  
sistemleri geliştirerek,  
maliyetleri düşürmek.

(1) <http://www.photoeverywhere.co.uk>

(2) <http://www.cafepress.co.uk/>

# Programlama Dilleri Evrimi



Şekil 1.1 Programlama Dillerinin Gelişimi

# Programlama Dilleri Evrimi

- **Birinci Nesil Diller:** **Makine düzeyinde** kodlama yani temeli 1 ve 0'lardan oluşan sayı gruplarına dayanan ve ortaya ilk çıkan programlama dilidir. İnsanların daha rahat okuyabileceği komut ve işlenen birimden oluşan kısa ifadeler halindeki **Assembly** daha sonra ortaya çıkan birinci derece nesil dillerden olup, her bilgisayarın farklı merkezi işlem türü için farklı olabilir.
- **İkinci Nesil Diller:** Modern dillerin temelini oluşturan dillerdir. Zengin kütüphaneleri vardır. Özellikle matematik ağırlıklı kütüphaneleri nedeniyle problem çözme alanında ağırlıklı olarak **Fortran**, ticari alanlarda veri işleme amaçlı olarak kullanılan **Cobol**, üçüncü nesil dillere öncülük eden zengin veri tipleri ve yordamsal yapılara sahip **Algol** ve kişisel bilgisayarlarda programlama kolaylığı sağlayan **Basic** sayılabilir.

# Programlama Dilleri Evrimi

- **Üçüncü Nesil Diller:** Modern - yapısal diller olarak üç temel bölüme ayrılırlar.
  - **Genel amaçlı yüksek düzey diller:** Algol'u model alan, mühendislik, bilim, gömülü sistemler ve kontrol sistemleri geliştirilmesinde yaygın olarak kullanılan PL/1, Pascal, Modula-2, C ve ADA sayılabilir.
  - **Nesneye yönelik yüksek düzey diller:** Nesneye yönelik çözümleme ve tasarımı desteklemek üzere geliştirilmiş bu dillerden C++ ve Java en yaygın olanlarıdır.
  - **Özel amaçlı diller:** Belirli bir uygulama tipi için tasarlanmış, alışılmış yapıların dışında yapılara ve sözdizimlerine sahip Prolog, Lisp ve APL gibi dillerdir.

# Programlama Dilleri Evrimi

- **Dördüncü Nesil Diller:** Yordamsal olan ve olmayan yapıları birleştirici özelliğe sahiptirler.

-**Sorgulama dilleri:** Özellikle veri tabanına erişmek, sorgulama ve raporlama yapmak üzere geliştirilen Dbase III/IV, Clipper, Delphi Paradox ve içlerinde en yaygın olanı SQL'dir.

-**Kod üreticiler:** Kullanıcının daha üst düzey yapı ve anahtar sözcüklerle daha az miktarda kod yazarak söz dizimi tam doğru olan üçüncü nesil bir dilde kod üretilmesini sağlarlar. Bilgisayar destekli yazılım mühendisliği –CASE araçları (UML,STP,SDL,MDA araçları gibi)

-**Diğer diller:** Karar destek, prototipleme, belirtim gibi farklı amaçlara hizmet eden diller bulunmaktadır. Ayrıca HTML, XML sayılabilir.

# Programlama Dilleri Evrimi

- **Beşinci Nesil Diller:** Araştırma halinde olan bu diller yeni oluşumları kapsayacaktır. Nesneye yönelik programlamadan sonra konuya yönelik (Subject Oriented) programlama üzerinde çalışılan konulardandır.

# PROGRAMLAMA YAKLAŞIMLARI

Programlama dilleri programlamaya yaklaşım açısından 4 gruba ayrılabilir:

- Prosedür yönelimli diller (örn. Fortran, Pascal, C)
- Fonksiyon yönelimli diller (örn. Lisp)
- Nesne yönelimli diller (örn. C++, C#, Java)
- Mantık yönelimli diller (örn. Prolog)

# PROSEDÜR YÖNELİMLİ (YAPISAL-Structured) YAKLAŞIM

- Prosedür yönelimli programlamada amaç problemi alt parçalara (subroutine, functions )bölerek bu parçaların çözümlerinin birleştirilmesidir.

**NOT !:**Nesne yönelimli programlama yaklaşımlarını kullanan dillerin neredeyse tamamı yapısal programlamayı da bünyelerinde barındırmaktadır.



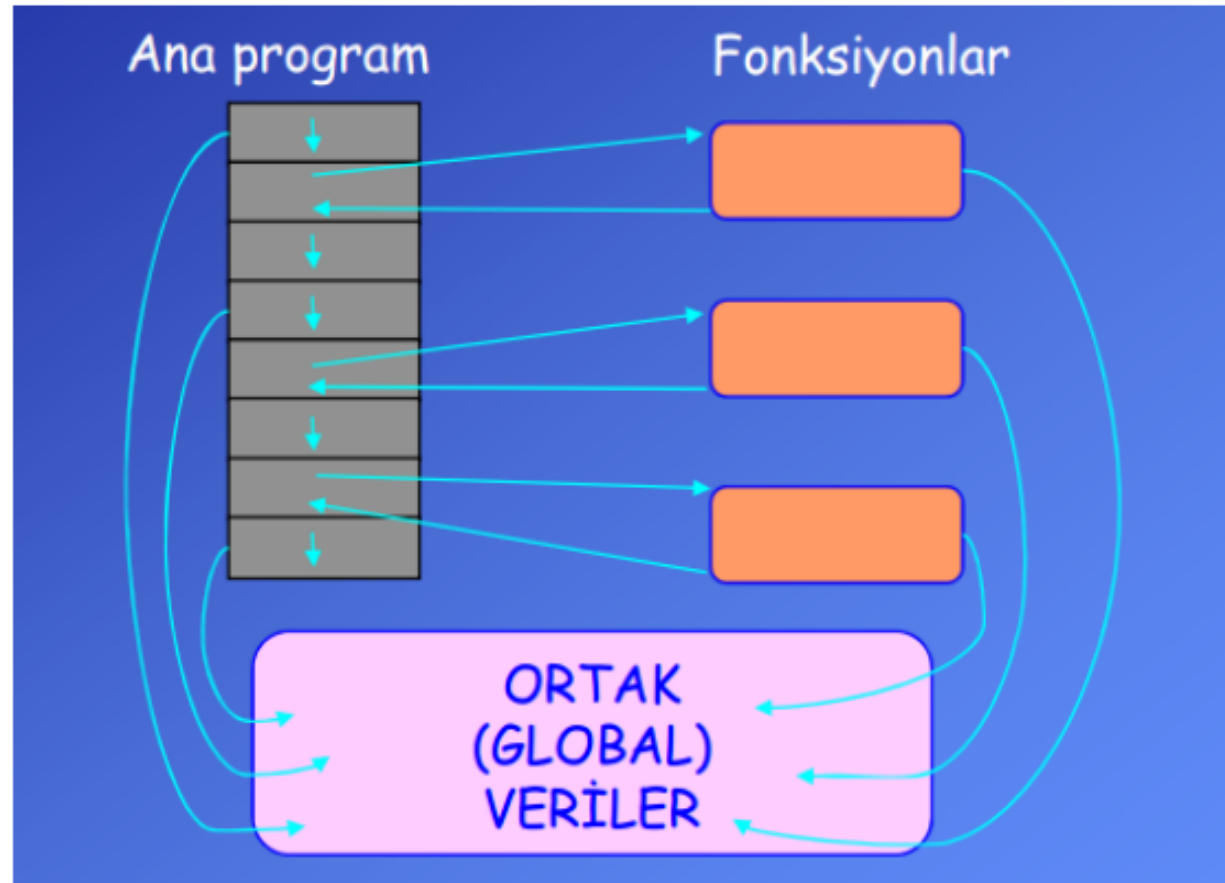
# PROSEDÜR YÖNELİMLİ (YAPISAL-Structured) YAKLAŞIM

Prosedürel programlama yaklaşımında,

- **İlk adım:** Programdan beklenen işlevselliği belirlemektir. (Yanıtlanması gereken soru: “Bu program ne yapacak?”)
- **İkinci adım:** İstenileni gerçekleştirmesi için programın atması gereken temel adımlar yüksek-düzeyli “pseudo” kodlar ya da akış diyagramları yardımıyla belirlenir.
- **Üçüncü adım:** Her temel adım daha küçük adımlara bölünerek tasarım daha rafine hale getirilir.

Bu yaklaşıma, prosedürel ayrıştırma (“procedural decomposition”) denir.

# PROSEDÜR YÖNELİMLİ (YAPISAL-Structured) YAKLAŞIM

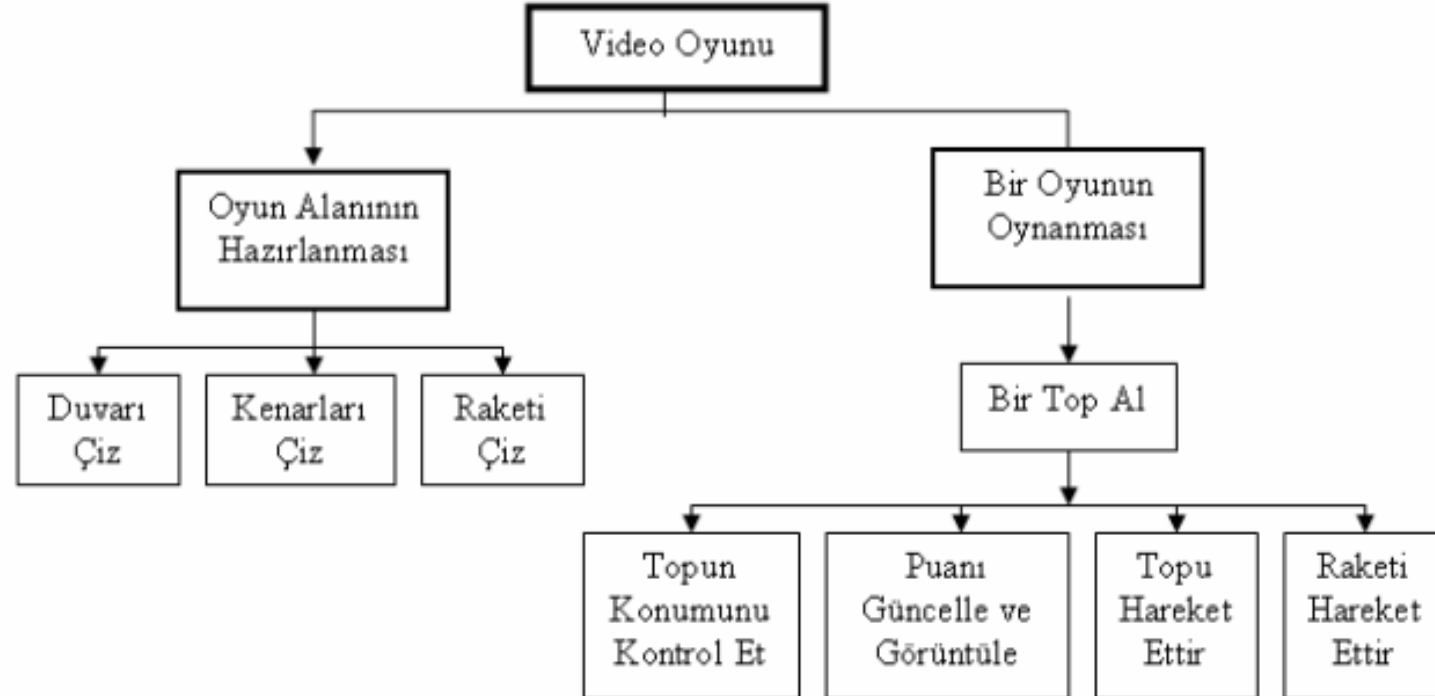


# PROSEDÜR YÖNELİMLİ (YAPISAL-Structured) YAKLAŞIM

- Yapısal programlama dillerinde, veriler ve metotlar ayrı ayrı tutulur. Veriler veri yapılarına gönderilir; kod blokları metotlara ve/veya fonksiyonlara gönderilir.
- Dolayısıyla, yapısal programlama geliştirmede, fonksiyonlara bağımlılık ve kodları takip etme zorluğu gibi dezavantajlar ortaya çıkar.
- Yapısal programlama geliştirme sürecinde:
  - Fonksiyonlar belirlenir.
  - Fonksiyonlar arasındaki ilişkiler belirlenir.
  - Fonksiyonların özellikleri belirlenir.

# PROSEDÜR YÖNELİMLİ (YAPISAL-Structured) YAKLAŞIM

Örnek:



# FONKSİYONEL YÖNELİMLİ YAKLAŞIM

Fonksiyonel yönelimli yaklaşımda, bir veri işleme işinde yapılacak işler sadece fonksiyon çağrıları ile ifade edilir.

- Kod tekrarını engellemekte ve aynı kodun farklı şartlar için tekrar tekrar çalışmasını sağlamaktadır.
- Kodun okunabilirliğini arttırmakta ve kod analizini daha kolay hale getirmektedir.
- Programın tasarlanması aşamasında tasarımcıya modüler yaklaşım yapmasını sağlamaktadır.

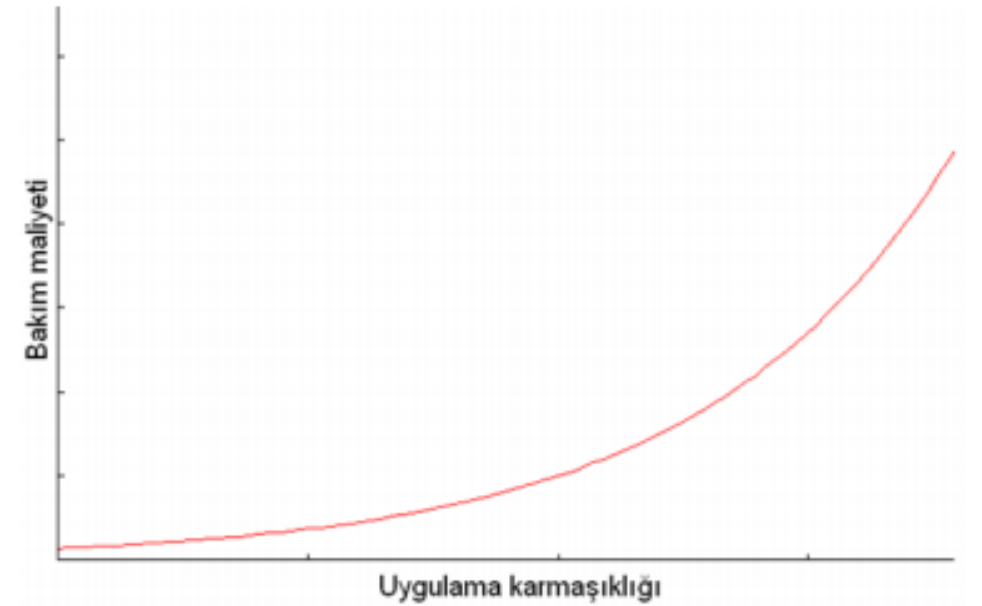
# MANTIK YÖNELİMLİ YAKLAŞIM

Mantık yönelimli yaklaşımda;

- Bir problemin çözümünde uygun mantıksal terimler kullanılarak bir algoritma kurulabileceği görülmektedir.
- Burada problemin çözümünün nasıl bulunacağı değil, problemin ne olduğunun tanımlaması yapılır.

# NESNE YÖNELİMLİ YAKLAŞIM

- Nesneye yönelik yaklaşım 1960'lı yılların sonuna doğru, yazılım dünyasında beliren bir bunalımın sonucu olarak ortaya çıkmıştır. Yazılımların karmaşıklığı ve boyutları sürekli artıyor, ancak belli bir nitelik düzeyi korumak için gereken bakımın maliyeti zaman ve çaba olarak daha da hızlı artıyordu.



# Nesne Yönelimli Programlama ?

- Nesne Yönelimli Programlama, mantıksal işlemlerden ziyade, nesnelere ve nesneler üzerine işlemlere odaklanan programlama dilidir.
- Nesne tabanlı programlamada kodlar **Sınıflar** içinde yazılmaktadır.
- Nesne tabanlı programlamayı etkin kullanabilmek için 3 ana prensip vardır.
  - Kalıtım(Inheritance)
  - Çok Biçimlilik(Polymorphism)
  - Erişim Kontrolü/Kapsülleme(Encapsulation)
  - \*Soyutlama (Abstraction)

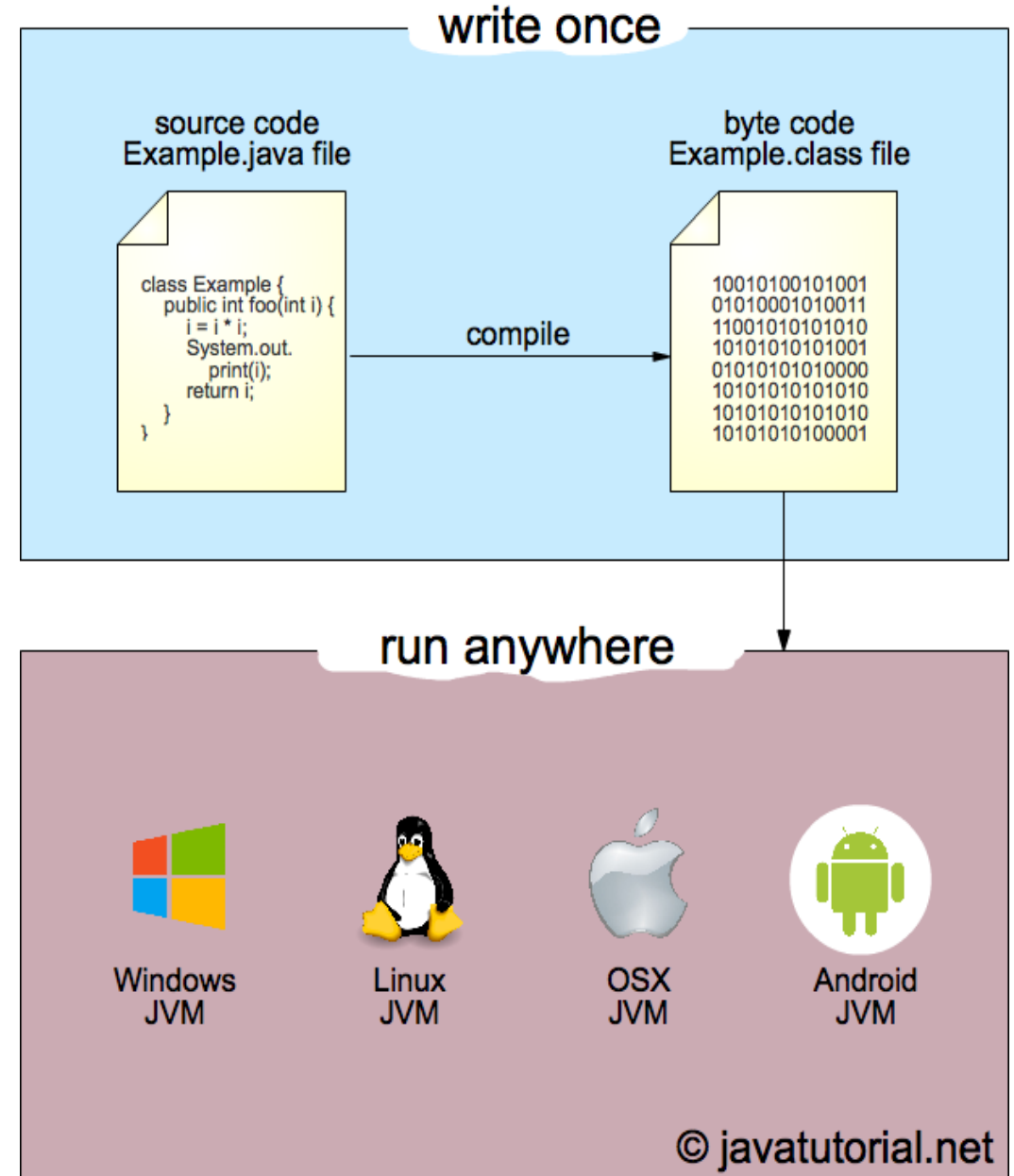


# Java Dili 1/2

- Java Sun Microsystems James Gosling tarafından geliştirilmeye başlandı.
- Günümüzde geliştiricisi **ORACLE**.
- Oracle, Sun Microsystems'i 2010'da satın aldı.
- Açık kaynak kodlu(GNU).( \*Community (Topluluk))
- C# ve C++ dillerine syntax olarak benzerdir.
- Java uygulamaları direkt olarak işletim sistemi üzerine çalışmaz. Java Sanal Makinesi / Java Virtual Machine(JVM) üzerine çalışır.
- Platform bağımsızdır, JVM olan sistemlerde çalışır.

# Java Dili 2/2

**Write Once, Run Anywhere !!**

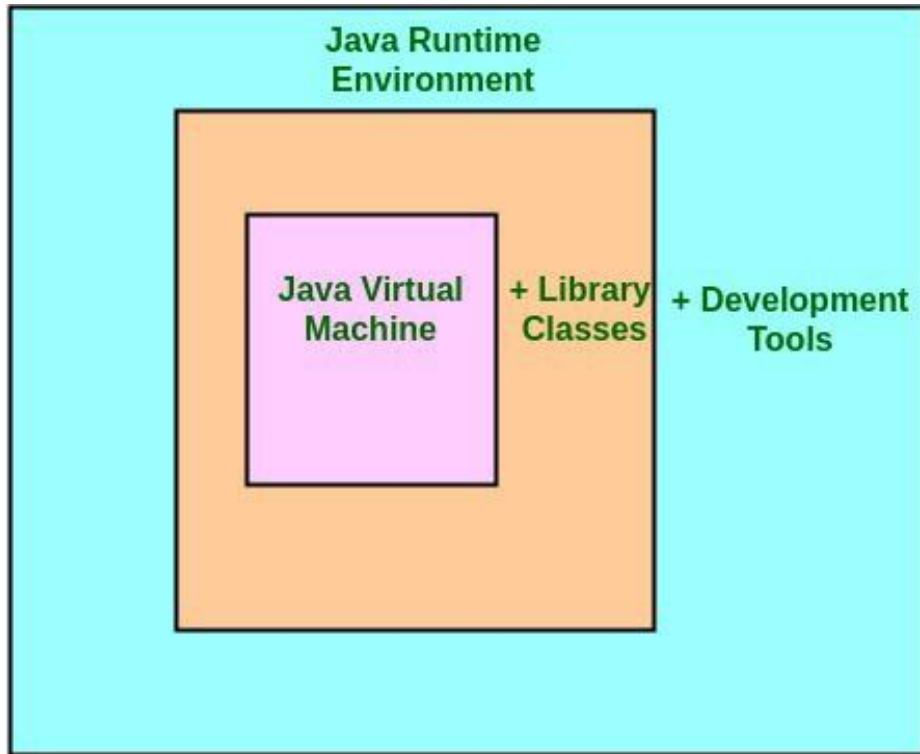


# JDK (Java Development Kit), JRE (Java Runtime Environment), JVM (Java Virtual Machine) 1/2

- JDK, Java kodlarını derleyip, çalıştırmaya yarayan kütüphaneleri içeren bir bileşendir.(SDK-Software Development Kit denebilir.)
- JRE, Java dili ile yazılmış olan uygulamaların çalışmasını sağlayan kütüphaneleri barındıran bileşendir.
- JVM, derlenmiş bytecode'lar yorumlanıp makine diline çevrilmesini sağlar.

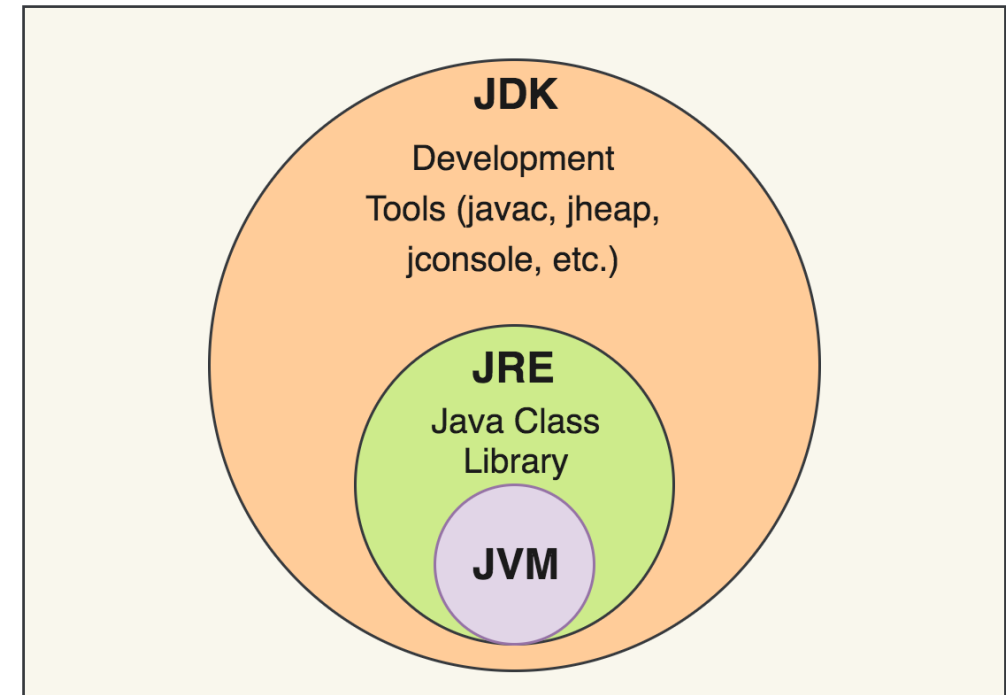
**NOT:JVM, JRE'nin bir alt bileşenidir. JRE ,içerisinde bir JVM barındırır.**

# JDK (Java Development Kit), JRE (Java Runtime Environment), JVM (Java Virtual Machine) 2/2



JDK = JRE + Development Tool  
JRE = JVM + Library Classes

<https://www.geeksforgeeks.org/differences-jdk-jre-jvm/>



<https://medium.com/@choudajharshata/difference-between-jdk-jre-and-jvm-189a786e2855>

# Java IDE's(Integrated Development Environment)1/2

- IDE, tümleşik geliştirme ortamı anlamındadır.
- Yazılan kodların otomatik olarak tamamlanması,
- Kodun syntax renklendirmesi yapabilmesi ve **Syntax** hatalarını algılayıp, derlemeden hataları göstermesi,
- Bir değişken,sınıf veya nesnenin değiştirilmesini kolaylaştırması,
- Kolay, hızlı ve temiz kod yazmak için çeşitli araçları bulunduran yapılar.

# Java IDE's(Integrated Development Environment)2/2

- Netbeans(Oracle)



- Eclipse(IBM)



- IntelliJ Idea(Jetbrains)

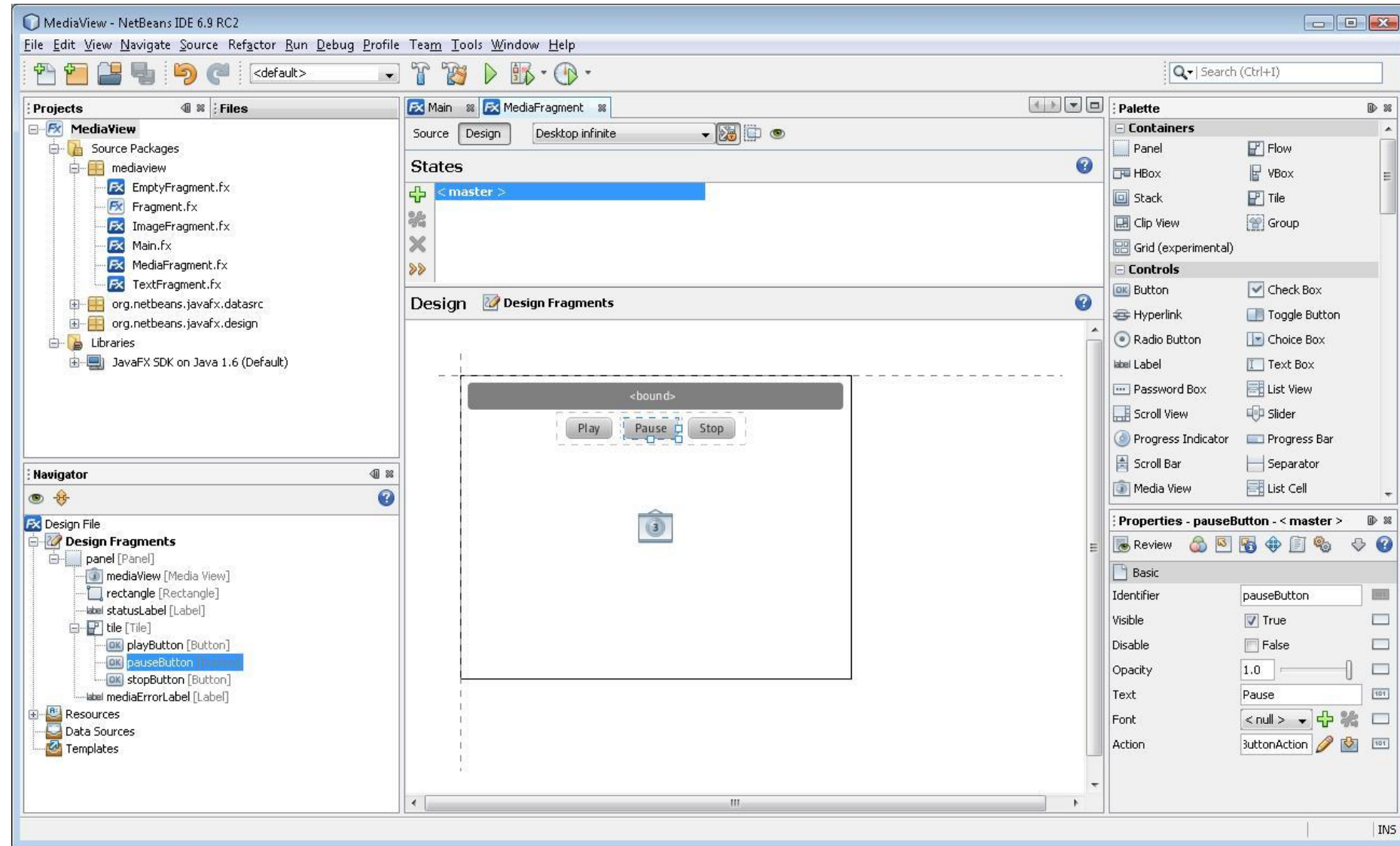


- JGrasp



# NETBEANS

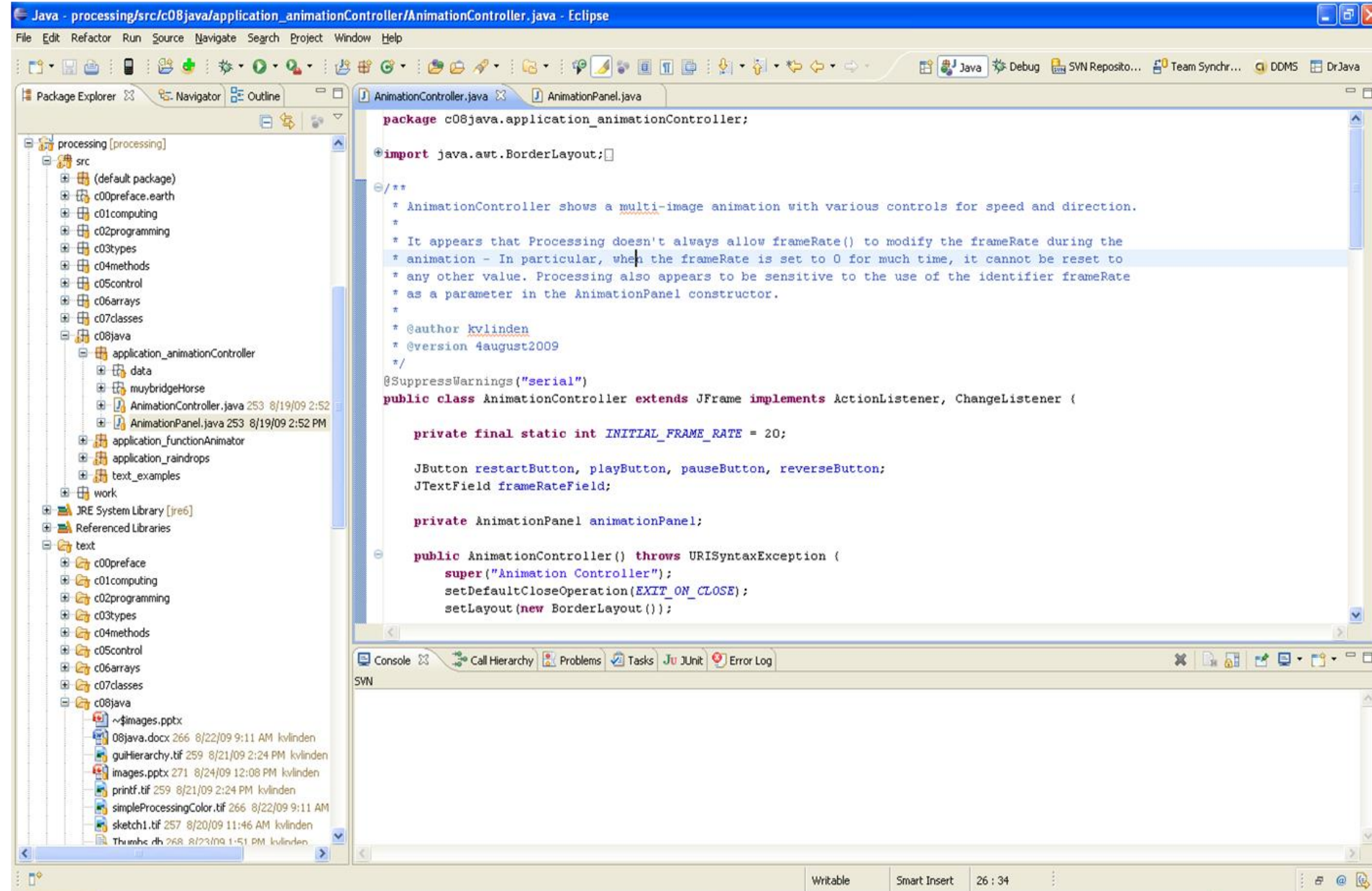
- Oracle tarafından geliştiriliyor.
- Açık kaynak kodlu.





# ECLIPSE

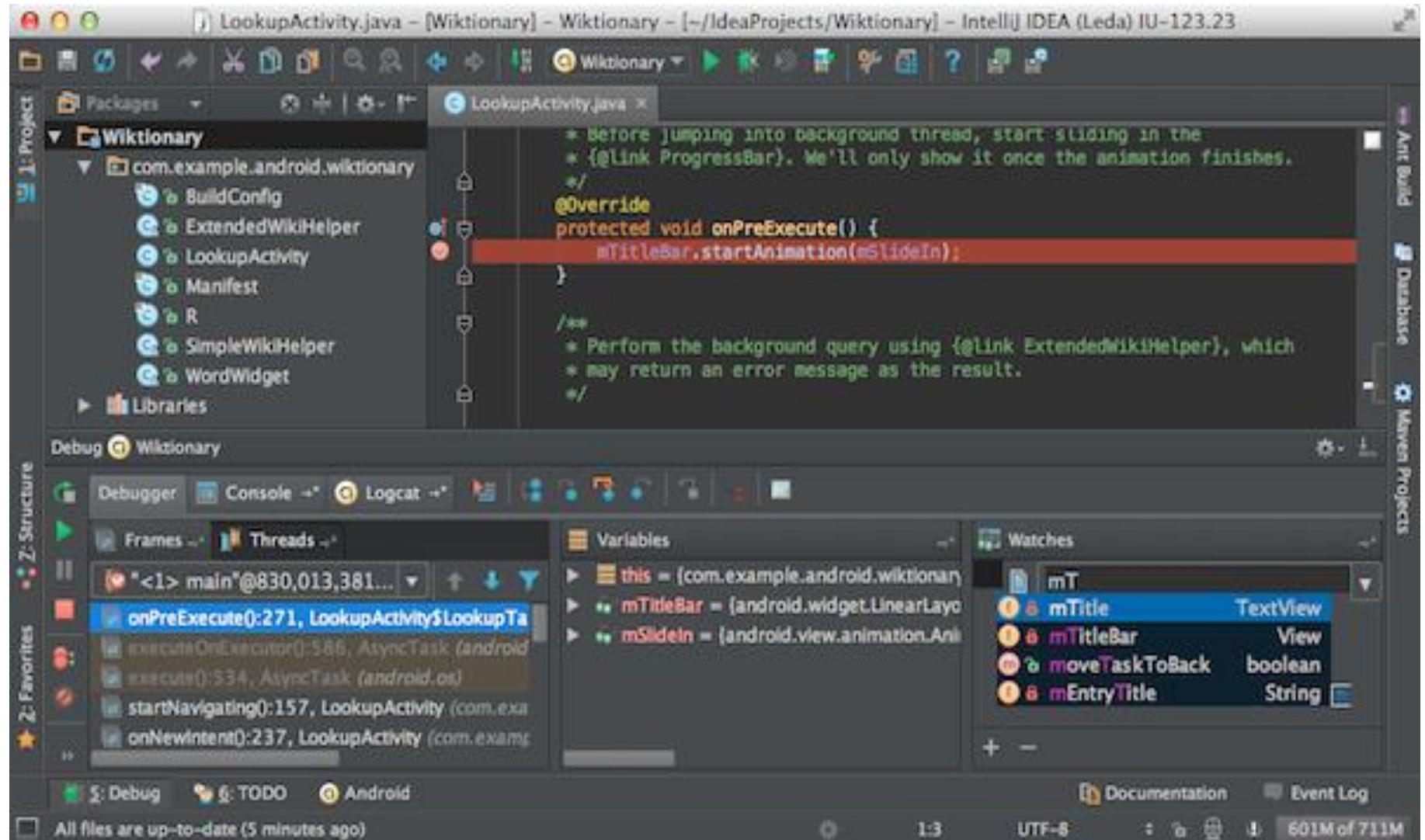
- IBM tarafından geliştiriliyor.
- Açık kaynak kodlu.





## INTELLIJ IDEA

- Jet Brains tarafından geliştiriliyor.
- Şirketler için Per User, first year 499\$,
- Kişiler için first year 149\$...



# DERS KAYNAK ÖNERİLERİ

Java ve Java Teknolojileri – Tevfik KIZILÖREN KODLAB

[www.javayaz.com](http://www.javayaz.com)

Java ile Temel Programlama, Bora Güngören, Seçkin Yayıncılık

Java Uygulamaları, David Flanagan, Pusula Yayıncılık

Just Java, Peter van der Linden, Prentice Hall

Java in a Nutshell, David Flanagan, O'Reilly

<http://tr.sun.com/training/courses>

Oracle Academy Eğitim Materyalleri

Java Software Solutions, Foundations of Program Design, Lewis, J., Loftus, W., Addison Wesley, 6th ed. 2008

<http://www.oracle.com/technetwork/documentation/index.html#java>

Java ile Nesne Programlama, Karaçay T., Seçkin Yayıncılık, ISBN 9789750219603, 2012

Java ve Yazılım Tasarımı, Altuğ B. Altıntaş, Papatya Yayıncılık, ISBN 9756797401, 2005

Java How to Program, Harvey M. Deitel- Paul J. Deitel, Prentice Hall, Sixth Edition, 0131483986, 2005

Nesneye Yönelik Programlama Ders Notları, Aydın Ö., Trakya Üniversitesi <http://ozlemaydin.trakya.edu.tr/d1.html>

Java Programlama Dili, Çoban T., [www.cgimarket.com/~turhan/Java](http://www.cgimarket.com/~turhan/Java)

Nesneye Dayalı Yazılım Geliştirme, 2008, <http://web.itu.edu.tr/~buzluca/ndyg/>

# Kaynaklar

- <https://www.oracle.com/tr/sun/>
- <https://bidb.itu.edu.tr/sevir-defteri/blog/2019/02/05/object-oriented-programming>
- Java ve Java Teknolojileri, *Tevfik KIZILÖREN* – Kodlab Yayınları
- <https://medium.com/@kplnosmn94/jvm-jre-ve-jdk-nedir-6cfee2727812>
- Yrd. Doç. Dr. Deniz KILINÇ-Celal Bayar Üniversitesi / Nesneye Yönelik Programlama Ders Notları
- Yrd. Doç. Dr. Erbil Akbay –Marmara Üniversitesi / Object Oriented Programming 1-2 Ders notları
- Dr. Öğrt. Üyesi Aysun ALTIKARDEŞ Nesne Yönelimli Programlama 1 Ders Notları
- Dr. Öğretim Üyesi Emin Borandağ- Celal Bayar Üniversitesi / Yazılım Yapımı Ders Notları
- <http://www.yazilimcildunyasi.com/>