

Java ile Programlamaya Giriş Eğitimi



<https://github.com/milikkan/acm-hacettepe-java-2021>

Eğitmen



Mustafa ILIKKAN

- Web Developer (Özgür Yazılım A.Ş.)
- Oracle Certified Java 11 Professional (OCP)
- ODTÜ Enformatik Enstitüsünde Yüksek Lisans (devam)
- İlgi Alanlarım:
 - * Web Tabanlı Yazılım Geliştirme
 - * Çevik Yazılım Geliştirme Metodolojileri
 - * Amazon Web Servisleri (AWS)
 - * Algoritma Geliştirme ve Problem Çözümü
 - * Java, JavaScript, Ruby
 - * Spring, Ruby on Rails, Node.js
- **Twitter:** @milikkan



Eğitimin Tanıtımı

- **Programlamaya giriş**
- **Java programlama dilinin temelleri**
- **Nesne merkezli programlama**
- **Spring Boot ile web programlamaya giriş**
- **Yeri geldikçe ilave faydalı bilgiler**

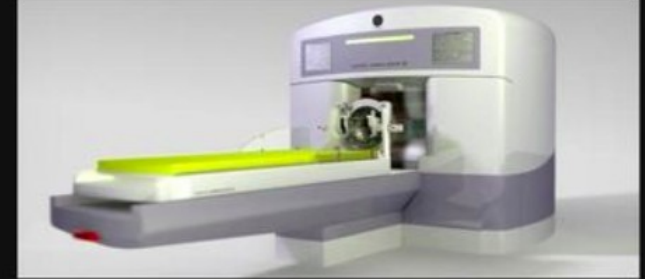
Hedef Kitlesi

- **Hiç programlama bilmeyenler**
- **Başka bir dilde programlama tecrübesi olup Java öğrenmek isteyenler**
- **Java bilip de temel konularda bilgisini artırmak veya tazelemek isteyenler**

Konu Başlıkları

- **Hafta-1:** Giriş ve Java'nın Temelleri
- **Hafta-2:** Karar yapıları, Döngüler ve Diziler
- **Hafta-3:** Nesne Merkezli Programlama-1
- **Hafta-4:** Nesne Merkezli Programlama-2
- **Hafta-5:** Temel Java Sınıfları (Collections)
- **Hafta-6:** Temel Java Sınıfları (Lambda ve Streams)
- **Hafta-7:** Spring Boot ve Web Programlamaya Giriş
- **Hafta-8:** Spring Boot ile Web Örnek Web Uygulaması

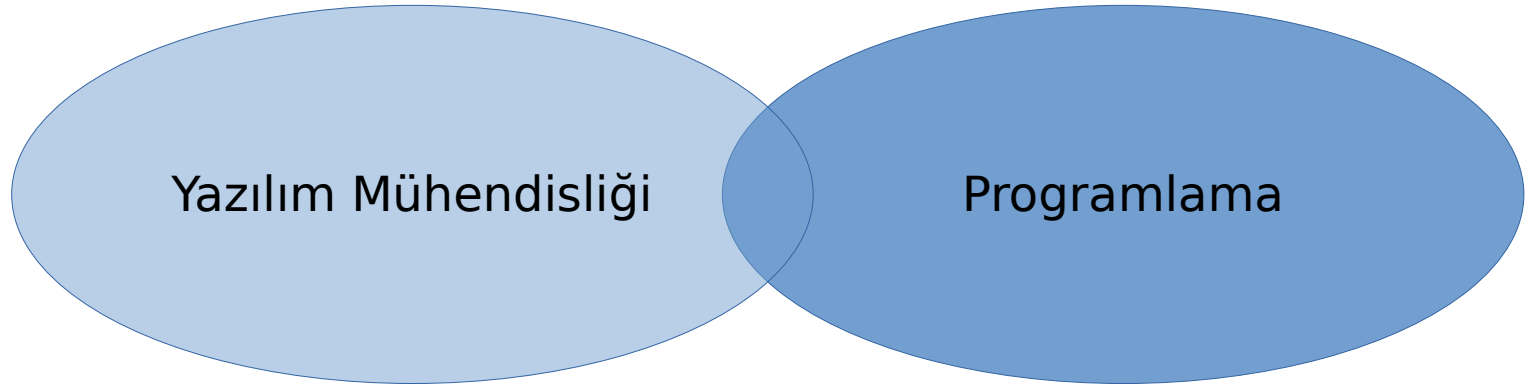
Yazılım Nedir?



Yazılım Nedir?

- Çalıştırıldığında istenen fonksiyonların yerine getirilmesini sağlayan bilgisayar komutları, veriler vs.
- **Temel Özellikleri**
 - Nispeten genç bir alan
 - Soyut (hacmi ve ağırlığı yok)
 - Karmaşık
 - ...

Programlama Nedir?



| Yazılım Mühendisliği | Programlama |
|--|--|
| Ekip işidir. | Kişiseldir |
| Bir planlama dahilinde belirli bir süreci kapsar | Yazılım geliştirme sürecinin bir aşamasıdır |
| Tam bir çözüm sunma amacıdadır. (veritabanı, bakım, testler, karlılık vs) | Hedef üzerinde çalışılan programın doğru bir şekilde yazılmasıdır. |

Yazılım Üzerine Bazı Yorumlarım

- Herkes yazılım öğrenebilir. Yaş, cinsiyet, fiziki özellikler fark etmez.
- Yazılım öğrenmeye başlangıç için derin bir matematik bilgisi gerekmez.
- Yazılım öğrenmek için üniversite mezunu olmak şart değildir. *(Faydası yok denemez)*
- Yazılımcı olmak için Bilgisayar Müh.okumak şart değildir. *(Faydası yok denemez)*

Yazılım Öğreneceklere Tavsiyelerim

- **Uzun süreli, istikrarlı bir çalışma.**
 - Sadece 2 haftalık kurs ile, Udemy eğitimi olmaz
 - Bol tekrar, yemek yapmayı öğrenmek veya enstrüman çalmak gibi.
 - Kod yazmayınca çabuk unutuluyor, yabancı dil gibi.
- **Meraklı olmak, bozmaktan korkmamak.**
- **İngilizce düzeyi**
- **Temelleri iyi öğrenmek**
 - Temeller uzun süre değişmez.
 - Temelleri iyi bilen başka dillere çabuk geçer.
- **Detaycı olmak, sabırlı olmak**
- **Her modaya kapılıp gitmemek** (Dil ve framework milliyetçiliğinden kaçınmak)

Programlama Dili Dışında Bunları da Öğrenmelisiniz

- **Git (versiyon kontrolü)**
- **Komut Satırı (shell)**
- **Metin Editörü, IDE**

Java ile Neler Yapılabilir?

- **TL; DR**
 - HER ŞEY
- **Günümüzde en popüler**
 - Web uygulamaları
 - Mobil uygulamalar
 - Masaüstü yazılımları
 - IoT uygulamaları
 - Oyunlar
 - Mikroservisler
 - Bulut Uygulamaları

“Star 7” Prototype

(*Green Project*, Oak Language)



James Gosling



<https://www.youtube.com/watch?v=1CsTH9S79qI>

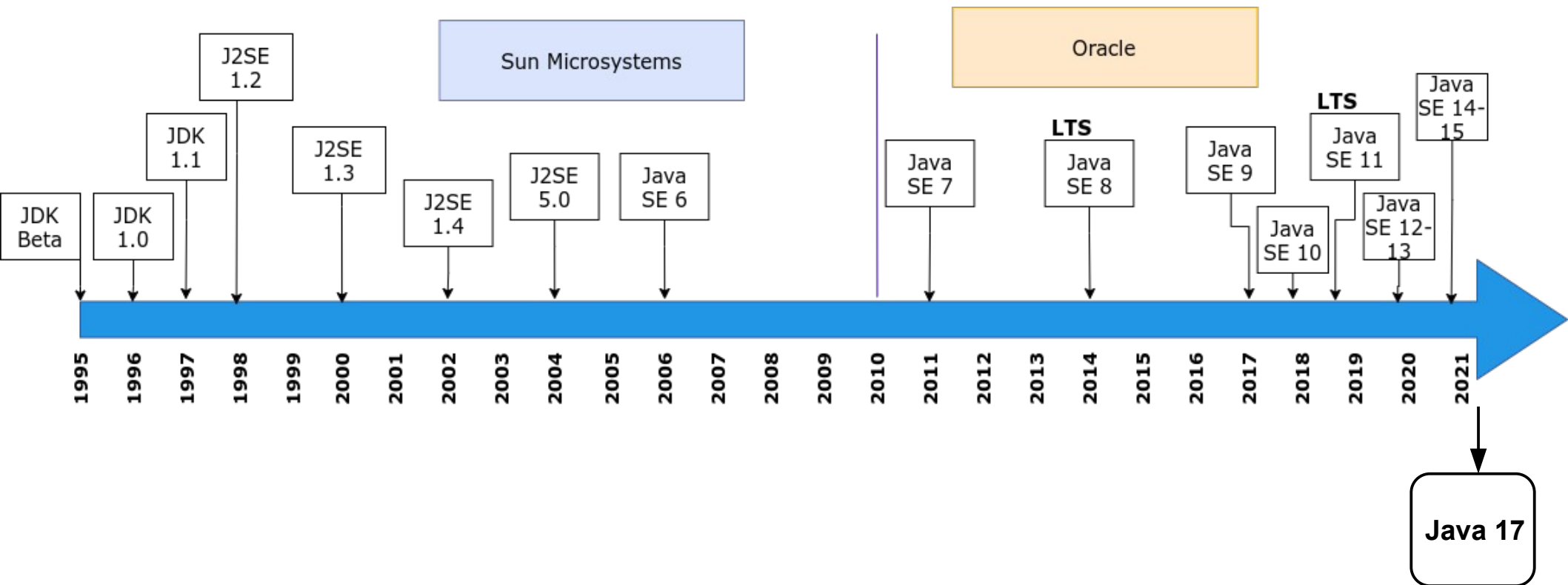
Java'nın Tarihçesi

James Gosling - 1996



- **JDK 1.0 (1996)**
- **J2SE 5.0 (2003)**
- **Java SE6 (2005)**
- **Java SE7 (2011)**
- **Java SE8 (2014)**
- **Java SE9 (2017)**
- **...**
- **Java SE16 (2021 Mart)**
- **Java SE17 (2021 Ekim)**

Java Version History



Tiobe Index

(<https://www.tiobe.com/tiobe-index/java/>)

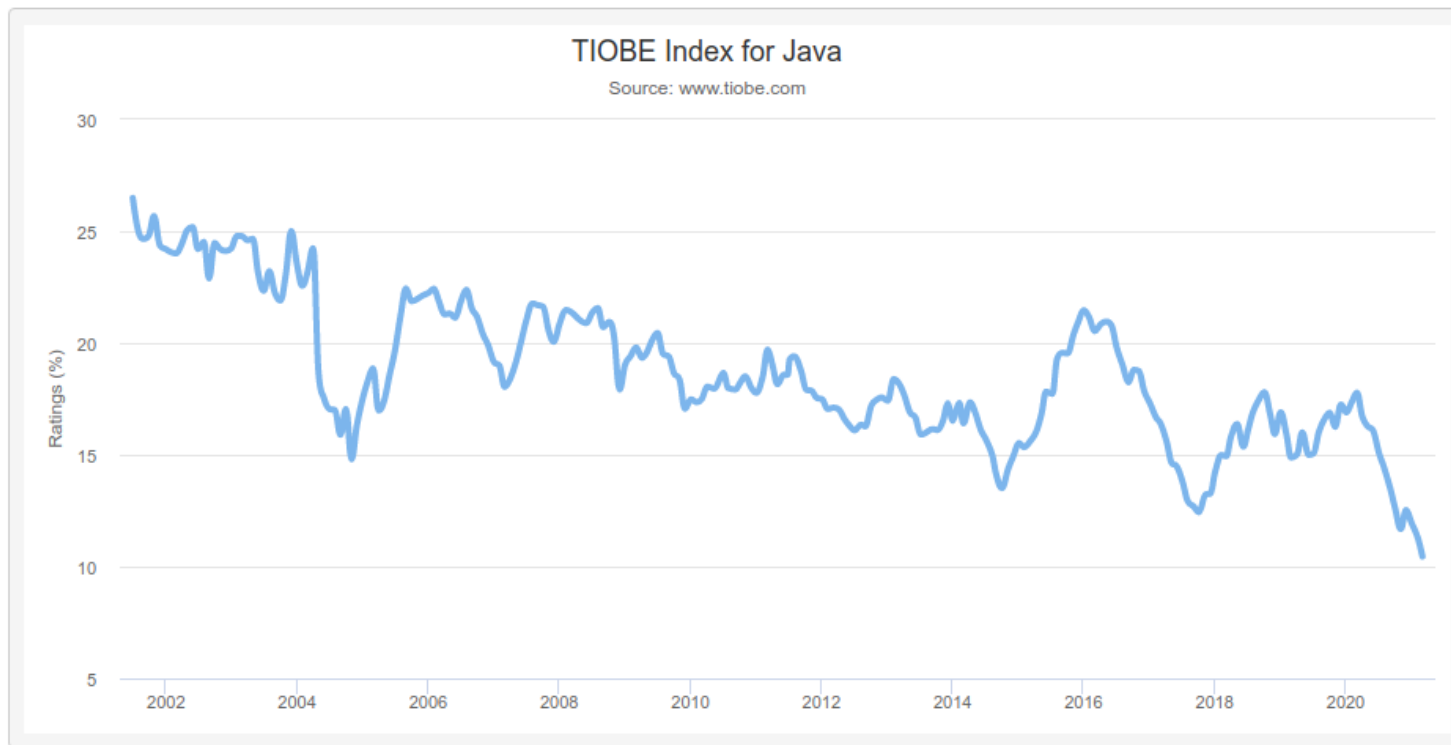
The Java Programming Language

Some information about Java:

📈 Highest Position (since 2001): #1 in Apr 2020

📉 Lowest Position (since 2001): #3 in Nov 2020

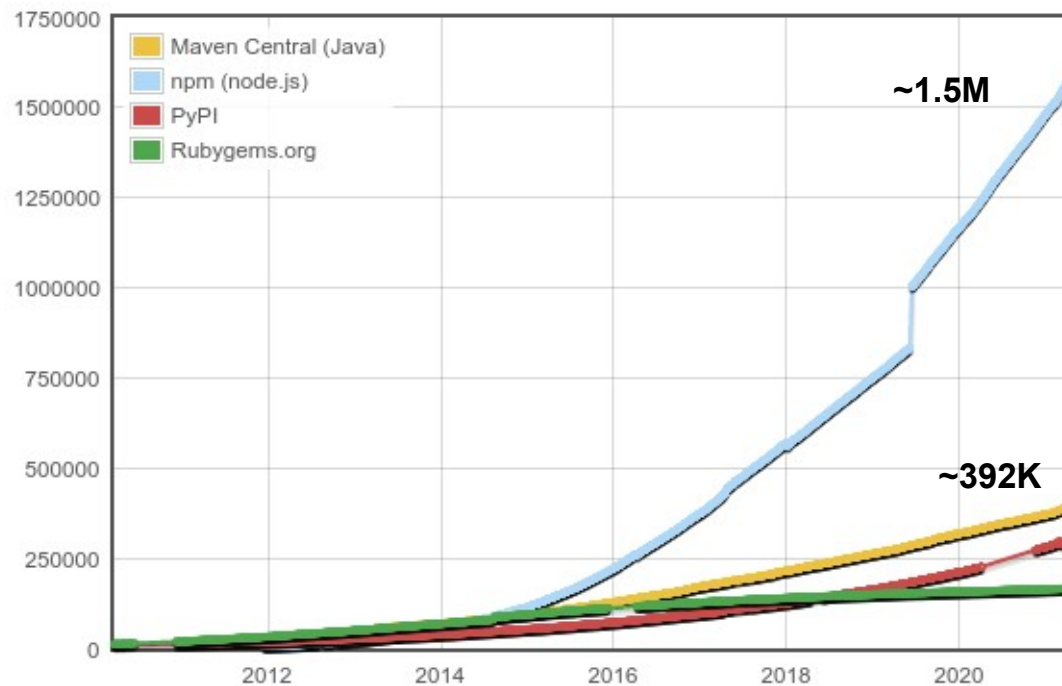
🏆 Language of the Year: 2005, 2015



Module Counts

(<http://www.modulecounts.com/>)

Module Counts



Include

- ☐ Clojars (Clojure)
- ☐ CPAN
- ☐ CPAN (search)
- ☐ CRAN (R)
- ☐ Crates.io (Rust)
- ☐ Crystal Shards
- ☐ Drupal (php)
- ☐ DUB (dlang)
- ☐ Gopm (go)
- ☐ Hackage (Haskell)
- ☐ Hex.pm (Elixir/Erlang)
- ☐ Julia
- ☐ LuaRocks (Lua)
- ☒ Maven Central (Java)
- ☐ MELPA (Emacs)
- ☐ Nimble (Nim)
- ☒ npm (node.js)
- ☐ nuget (.NET)
- ☐ Packagist (PHP)
- ☐ Pear (PHP)
- ☐ Perl 6 Ecosystem (perl 6)
- ☒ PyPI
- ☒ Rubygems.org
- ☐ Vim Scripts

time period ☒ all time ☐ last year ☐ last 90 days ☐ last 30 days ☐ last 7 days

Bazı sayılar

- 12+ milyon Java geliştirici
- 51+ milyar aktif JVM
- 30+ milyar JVM bulutta çalışıyor
- Üniversitelerde en yaygın öğretilen dil
- En çok iş ilanı verilen dillerden birisi (hem ülkemizde, hem dünyada)

Java'nın Avantajları

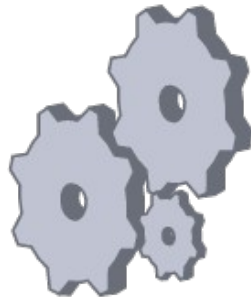
- **Platformdan bağımsız**
 - Write Once Run Anywhere (WORA)
- **Güvenli**
 - Pointer yok, otomatik çöp toplama (garbage collection)
- **Nesneye yönelik ve fonksiyonel**
- **Statik tipli**
- **Yüksek performans (Java Sanal Makinesi)**
- **Multithreaded**
- **Geniş kütüphane ve ekosistem**

Java Compilation 101

source code



HelloWorld.java



compiler

Java Byte Code



HelloWorld.class

Java Runtime Environment



WORA:

Write Once Run Anywhere

Java Byte Code



Linux



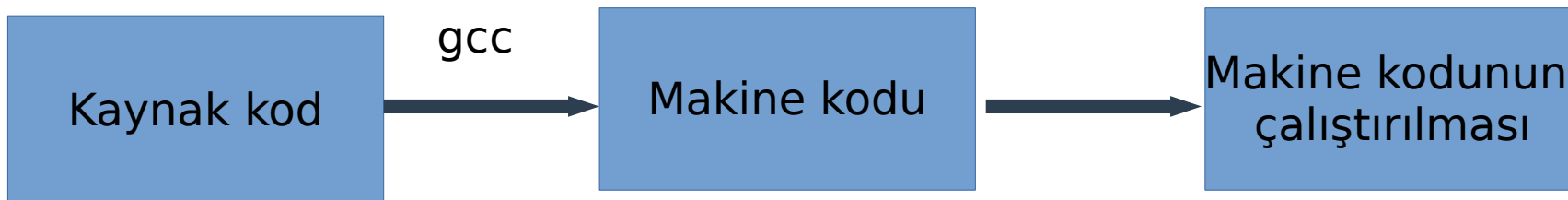
Windows



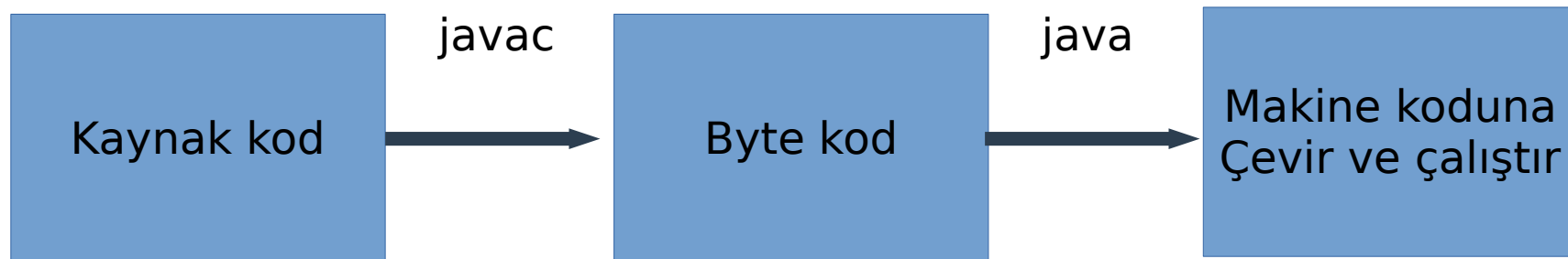
MacOS

Java Programlarının Derlenmesi

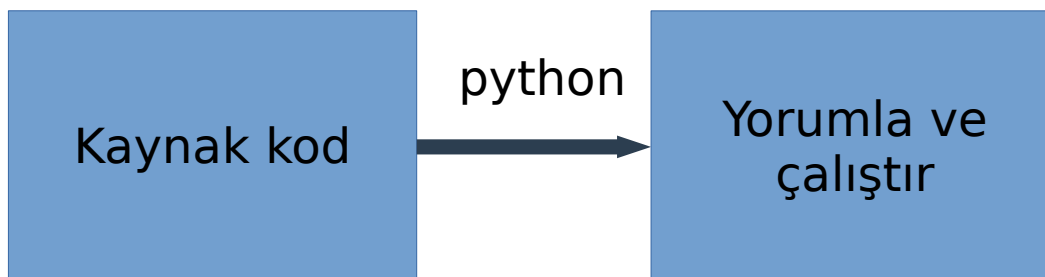
C++ kodunun derlenmesi (compiled/derlenen bir dil)



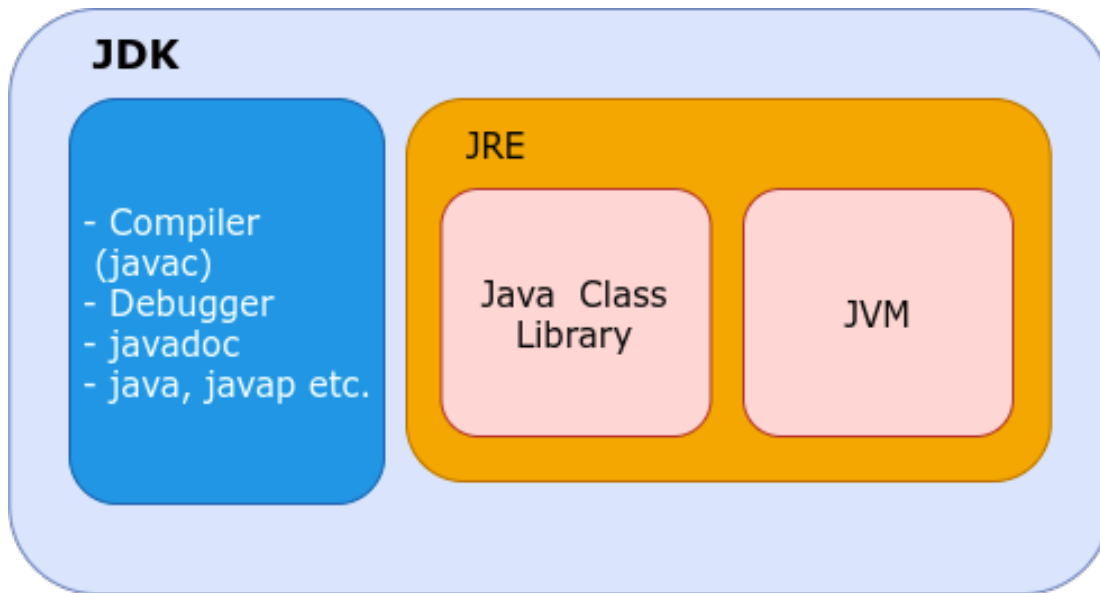
Java'nın Derlenmesi (compiled ve interpreted bir dil)



Python'un Kodunun Çalışması (Interpreted/yorumlanan bir dil)



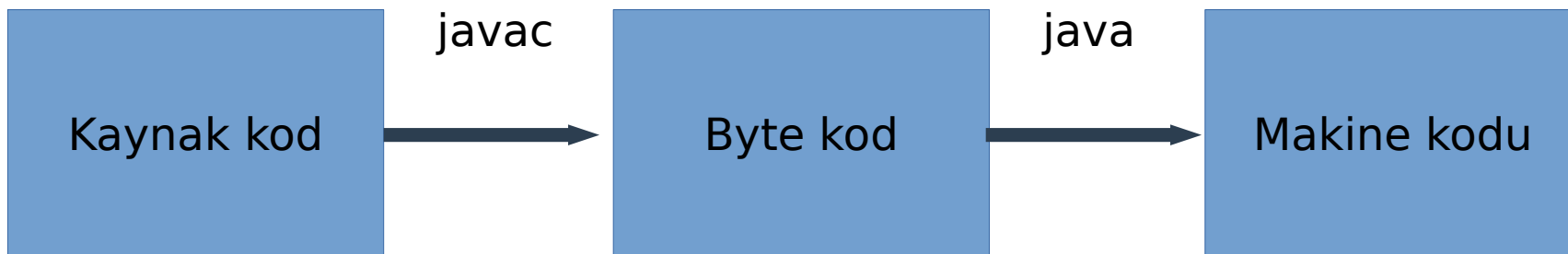
Java Ekosistemi



JVM: Java Virtual Machine
- Java Sanal Makinesi

JRE: Java Runtime Environment
- Java Çalışma Ortamı

JDK: Java Development Kit
- Java Geliştirme Kiti



Java Kurulumu

- <https://dev.java>
 - Download Java 17
- **IntelliJ IDEA Community Edition**
 - Add new JDK

Java Kurulumunun Testi

- **Kurulumu test etmek için**
 - javac -version
 - Java -version
- **javac komutu:**
 - java uzantılı kaynak kodumuzu derleyip
 - Kaynak kodu bytecode'a çevirir
- **Java komutu**
 - Derlediğimiz bytecode'u makine koduna çevirip jvm üzerinde çalıştırmak için

JShell (REPL)

- **REPL**

- Read-evaluate-print-loop
- Uzun programlar yazmadan komutları hızlıca denemek için kullanılabilir

- **jshell komutu:**

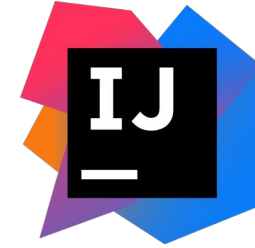
- Java'nın REPL aracı jshell'dir. Java9 sürümünde eklenmiştir.
- Çalıştırmak için komut satırına “jshell” yazın.
- Çıkış için CTRL+D

IDE Seçenekleri

- **IDE:** Integrated Development Environment
- Tümüleşik Geliştirme Ortamı
- **Faydaları:**
 - Hızlı kod geliştirme
 - Otomatik tamamlama, sentaks renklendirme
 - Test, build, kod analizi, deployment vb için entegre araçlar
- İlk öğrenme maliyeti yüksek olabilir
- Küçük programlar için hantal kalır
- Genelde fazla sistem kaynağı kullanırlar

IDE Seçenekleri

- **IntelliJ IDEA**
 - Community sürümü ücretsiz
 - Enterprise sürümü için lisans gerekiyor
- **Eclipse**
 - Ücretsiz, açık kaynak
 - Eclipse vakfı (aynı zamanda JakartaEE yi geliştiriyor)
 - Çok yetkin ancak karmaşık gelebilir
- **NetBeans**
 - Original Sun Microsystems / Oracle
 - Apache vakfına devredildi
 - Ücretsiz, açık kaynak
- **VSCode**
 - Microsoft ürünü
 - Ücretsiz, açık kaynak
 - Diğerleri kadar popüler ve yetkin değil ancak gelişmeye devam ediyor.
 - Daha hafif ve hızlı
- **Online alternatifler**
 - repl.it
 - gitpod
 - AWS Cloud9



Visual Studio Code



İlk Programın Anatomisi

```
1 // İlk Java programım
2
3 public class Merhaba {
4     public static void main(String[] args) {
5         System.out.println("Merhaba");
6     }
7 }
```

→ yorum

→ Sınıfın ismi

→ Main metodu

Java'da

- * Satırlar noktalı virgül ile bitmelidir.
- * Boşluklar sadece okunabilirlik için vardır, kodu karmakarışık da yazabiliriz. Ama yazmamalıyız.
- * Bloklar {} ile başlar ve biter. Süslü parantez, curly brace.

“Merhaba Hacettepe” Örnekleri

- 1. Console programı**
- 2. Window (desktop) programı**
- 3. Web (Spring Boot) programı**

Yorumlar (comments)

- **Tek satır yorumlar**
 - `//` bu bir yorum satırı
- **Çok satırlı yorum**
 - `/*` bu bir
çok satırlı
yorum `*/`

Değişkenler

değişkenİsmi = değer

- **Değişken ismi**
 - CamelCase konvansiyonu
 - Anahtar kelime olamaz
 - Sadece rakam ve _, \$ karakterleriyle başlayabilir
 - Özel işaretler içeremez
- **Değer**
 - Değişkene atanacak veri

Deklerasyon ve İklendirme

- **Java statik tipli bir dil olduğu için önce değişken tipini, sonra değişken adını yazmalıyız.**
 - `DeğişkenTipi DeğişkenAdı;`
- **Değişken tanımlama ve değer atama ayrı ayrı yapılabilir.**
 - `DeğişkenTipi = Değer;`
- **Değişkene değer atandıktan sonra, atana tip bir daha değiştirilemez. Ancak aynı tipte başka değer atanabilir.**

Java Veri Tipleri

- **Sayı tipleri (primitive)**
 - Tam sayılar
 - byte -> 8 bit (-128 / 127)
 - short -> 16 bit
 - int -> 32 bit (**varsayılan tip**)
 - long -> 64 bit
 - Ondalıklı sayılar
 - float -> 32 bit
 - double -> 64 bit (**varsayılan tip**)

Java Veri Tipleri

- **Mantıksal tipler (primitive)**
 - true ve false (boolean)
- **Karakter Tipi (primitive)**
 - char -> 16 bit
 - Tek tırnak içinde yazılır. Örnek: 'a'
 - Aslında bir tam sayı tipidir.
- **String**
 - Primitive değil, referans tipidir.
 - Çift tırnakla yazılır. Örnek: "java dili"

Aritmetik Operatörler

- **+, -, *, /, %**
 - `sayı1 (oparatör) sayı2`
- Birden fazla sayı varsa matematikteki öncelik sırası geçerlidir:
 - Önce çarpma, bölme ve modülüs
 - Sonra toplama, çıkarma
 - Hepsi aynı seviyedeyse soldan sağa doğru
 - İşlem önceliği parantezlerle değiştirilebilir

İlişkisel Operatörler

- **Eşitlik**

- ==

- **Eşitsizlik**

- !=

- **Büyüklik, Küçüklük**

- >, <, >=, <=

- **Bu operatörler boolean (true/false) sonuç döndürür**

Girdi-Çıktı (Input Output, I/O)

- **Ekrana yazmak için**

- `System.out.println(mesaj);`

- **Kullanıcıdan değer almak için**

- `Scanner scanner = new Scanner(System.in);`
 - `Scanner.nextLine();`

Örnek: Basit Toplama Programı

- **Planlama adımları:**

1. Hoş geldin mesajını ekrana yaz
2. Birinci sayıyı kullanıcıdan al
3. Kullanıcıdan alınan sayıyı ilkSayi isimli String değişkene kaydet
4. İkinci sayıyı kullanıcıdan al
5. Kullanıcıdan alınan sayıyı ikinciSayi isimli String değişkene kaydet
6. İlkSayi ve ikinciSayi değişkenlerini topla ve sonucu ekrana yaz.

Java API Dokümantasyonu

- Google “Java 17 API docs”
 - Sık kullanılanlara ekle
 - API dokümantasyonunu kullanmayı ve okumayı bilmek iyi bir programcının olmazsa olmaz özelliğidir.



Java® Platform, Standard Edition & Java Development Kit Version 17 API Specification

This document is divided into two sections:

Java SE
The Java Platform, Standard Edition (Java SE) APIs define the core Java platform for general-purpose computing. These APIs are in modules whose names start with `java`.

JDK
The Java Development Kit (JDK) APIs are specific to the JDK and will not necessarily be available in all implementations of the Java SE Platform. These APIs are in modules whose names start with `jdk`.

| All Modules | Java SE | JDK | Other Modules |
|----------------------------------|--|-----|---------------|
| Module | Description | | |
| <code>java.base</code> | Defines the foundational APIs of the Java SE Platform. | | |
| <code>java.compiler</code> | Defines the Language Model, Annotation Processing, and Java Compiler APIs. | | |
| <code>java.datatransfer</code> | Defines the API for transferring data between and within applications. | | |
| <code>java.desktop</code> | Defines the AWT and Swing user interface toolkits, plus APIs for accessibility, audio, imaging, printing, and JavaBeans. | | |
| <code>java.instrument</code> | Defines services that allow agents to instrument programs running on the JVM. | | |
| <code>java.logging</code> | Defines the Java Logging API. | | |
| <code>java.management</code> | Defines the Java Management Extensions (JMX) API. | | |
| <code>java.management.rmi</code> | Defines the RMI connector for the Java Management Extensions (JMX) Remote API. | | |
| <code>java.naming</code> | Defines the Java Naming and Directory Interface (JNDI) API. | | |
| <code>java.net.http</code> | Defines the HTTP Client and WebSocket APIs. | | |
| <code>java.prefs</code> | Defines the Preferences API. | | |
| <code>java.rmi</code> | Defines the Remote Method Invocation (RMI) API. | | |

Örnek: Basit Toplama Programı

- **Tekrar Düzenlenen Planlama adımları:**

1. Hoş geldin mesajını ekrana yaz
2. Birinci sayıyı kullanıcıdan al
3. Kullanıcıdan alınan sayıyı ilkSayi isimli String değişkene kaydet
- 4. İlk sayıyı integer tipine çevir.**
5. İkinci sayıyı kullanıcıdan al
6. Kullanıcıdan alınan sayıyı ikinciSayi isimli String değişkene kaydet
- 7. İkinci sayıyı integer tipine çevir.**
8. İlkSayi ve ikinciSayi değişkenlerini topla ve sonucu ekrana yaz.

Karar Yapılarına Giriş (if-else)

- Normalde program akışı yukarıdan aşağıya sırayla gider.
- Program akışını bazı şartlarda değiştirmek için if-else yapısını kullanabiliriz.

```
if (şart) {  
    // işlemler  
}
```

* tek bir işlem satırı varsa
blok açmaya gerek yok.

```
if (şart) {  
    // işlemler  
} else {  
    // işlemler  
}
```

* şart true ise üstteki blok

* false ise alttaki bloğu çalıştır

Örnek: Tek mi Çift mi

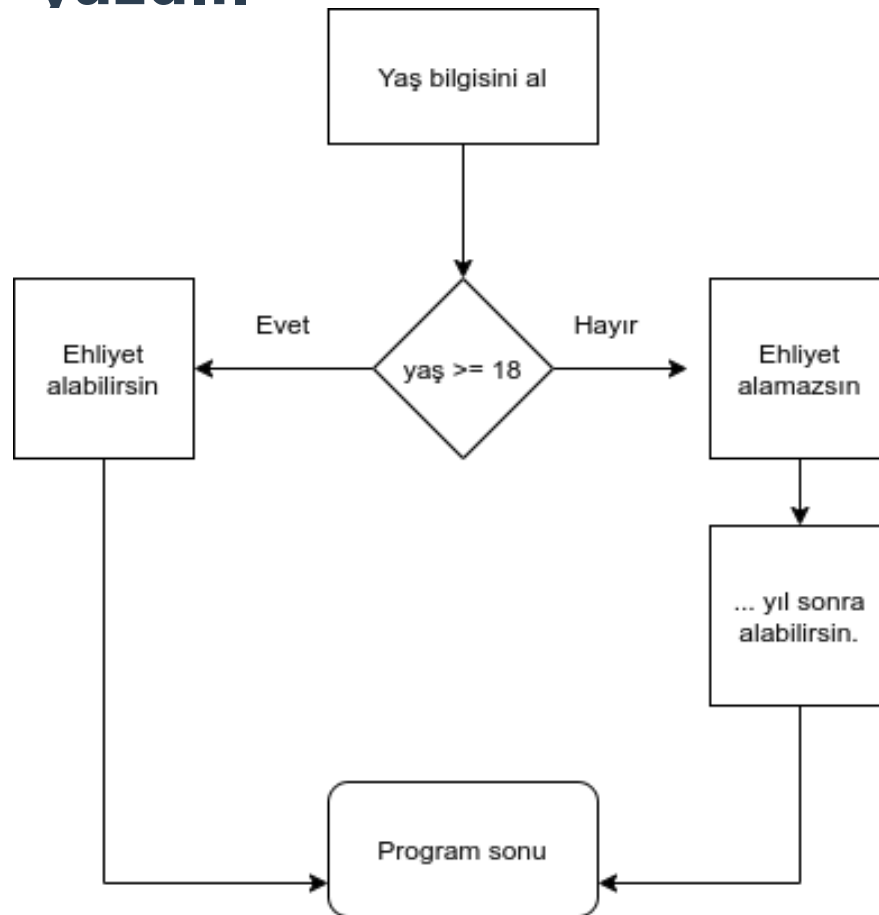
- **Kullanıcıdan bir sayı alarak, girilen sayının tek mi çift mi olduğunu ekrana yazan program.**
 - Sayıyı kullanıcıdan al
 - Sayı çift mi? (buradaki karar if-else yapısı ile verilir)
 - Evet -> Ekrana “Girdiğiniz sayı çift” yaz
 - Hayır -> Ekrana “Tekrar deneyiniz” yaz

İlave Kaynaklar

- University of Helsinki Java MOOC
<https://java-programming.mooc.fi/>
- Official Tutorials
<https://dev.java/learn/>
- Codecademy Learn Java Track
<https://www.codecademy.com/learn/learn-java>

Örnek: Yaş Kontrolü

- Kullanıcıya yaşını sor. Yaşı 18'den küçükse “Ehliyet alabilirsin” vazdır.



Ödevler

1. Kişinin boyu ve kilosunu alarak vücut kitle indeksini hesaplayan ve çıkan sonuca göre kişinin kategorisini ekrana yazan bir program yazın.

- < 18.5 : zayıf
- $\geq 18.5 - < 25$: normal
- $\geq 25 - < 30$: fazla kilolu
- ≥ 30 : obezite

2. İki nokta arasındaki mesafeyi hesaplayan programı yazın.

- Not: Pisagor teoremi. Karekök alma için `Math.sqrt()` metoduna bakın.