
Hafta-1

JAVA 101

OUTLINE

- ▶ Java Gereksinimleri
- ▶ İlk JAVA Kodu (Terminal vs IDE)
- ▶ IntelliJ IDE
- ▶ JAVA ile Değişkenler
- ▶ JAVA ile Akış kontrolü
- ▶ JAVA Array işlemleri
- ▶ JAVA Debug işlemi (IntelliJ üzerinden)



JAVA GEREKSİNİMLERİ

▶ Java Development Kit (JDK)

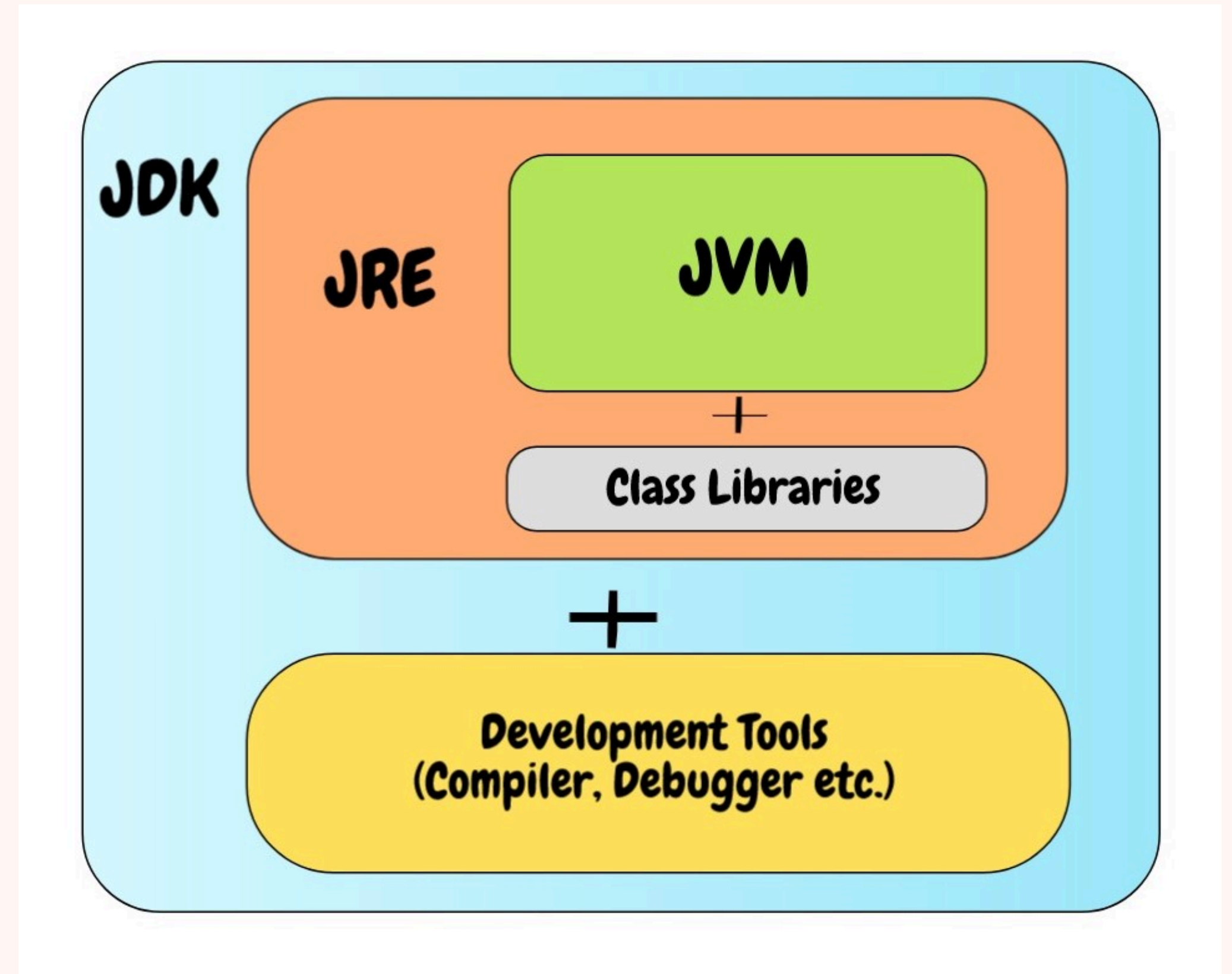
☑ Java Runtime Environment (JRE)

▶ Tüm değişken tipleri, built-in metodları (örn: println), JAVA sınıf kütüphaneleri.

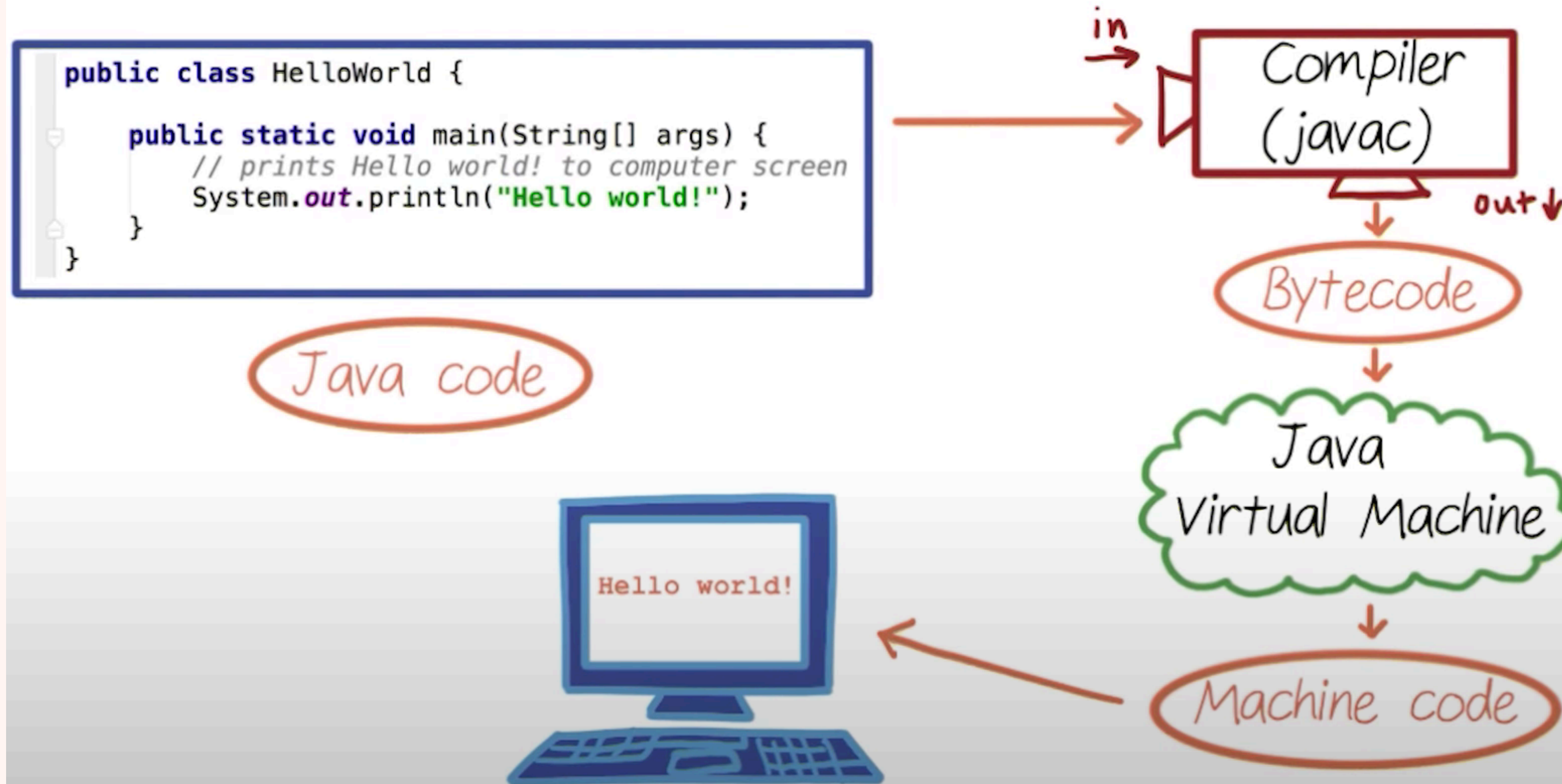
▶ Java Virtual Machine

☑ Java Compiler (javac)

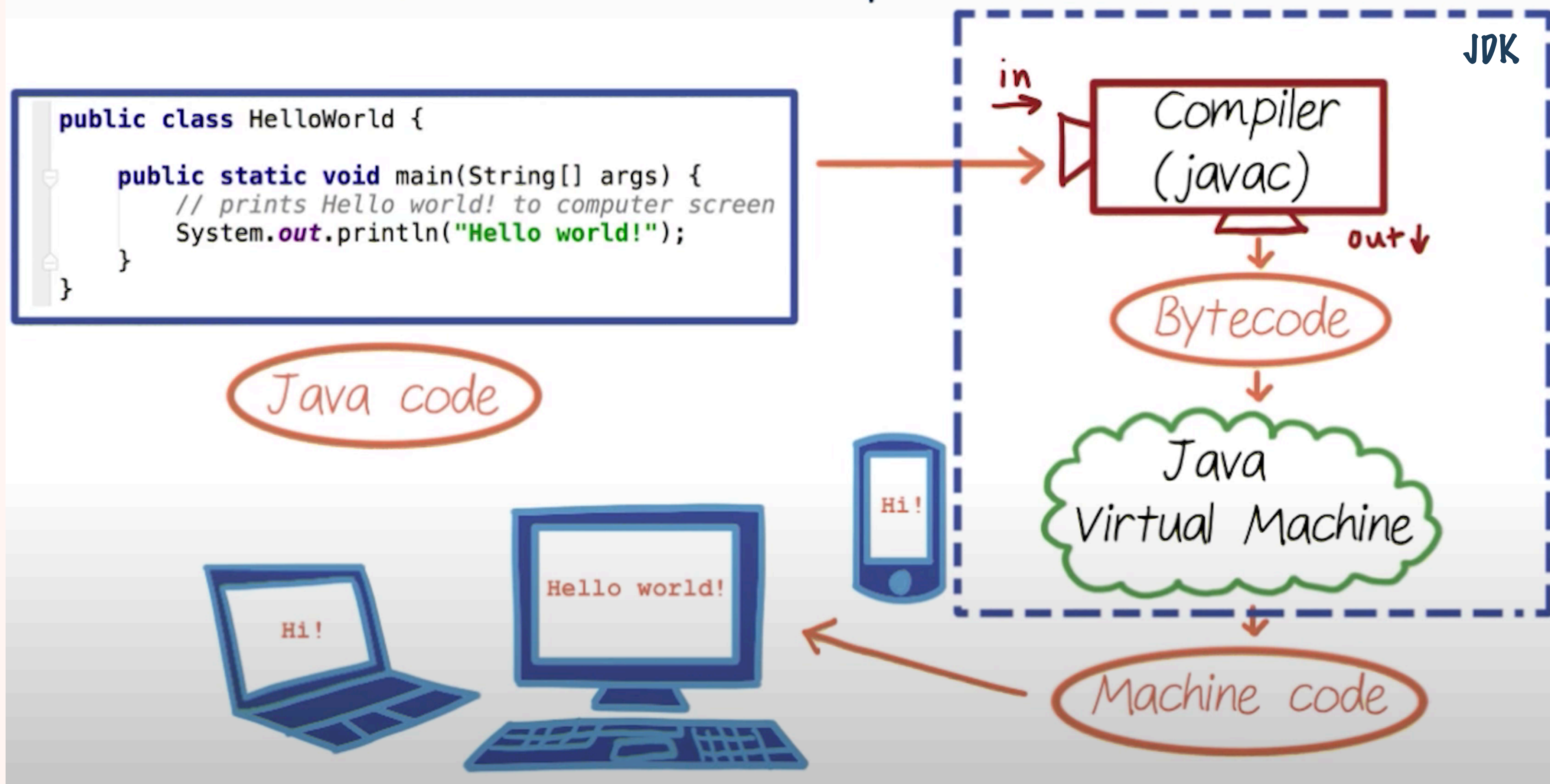
☑ debuggers, javadocs, vs.



JAVA GEREKSİNİMLERİ



JAVA GEREKSİNİMLERİ



JAVA İLK PROGRAM

- ▶ Java kodu terminalden çalıştırmak
 - ▶ Println kodu text dosyasına kaydet. (HelloWorld.java)
 - ▶ İlgili dizine geç "cd desktop"
 - ▶ "javac HelloWorld.java" ile derle
 - ▶ "java HelloWorld" ile çalıştır.
 - ▶ Kodu güncelle, kaydet, tekrar çalıştır. (SONUÇ ?)
 - ▶ Bu zorlukla uğraşmak yerine IDE kullan :)

```
Last login: Tue Aug 10 12:00:15 on ttys000
(base) kadirseker@KadirMacPro ~ % java -version
java version "16.0.2" 2021-07-20
Java(TM) SE Runtime Environment (build 16.0.2+7-67)
Java HotSpot(TM) 64-Bit Server VM (build 16.0.2+7-67, mixed m
(base) kadirseker@KadirMacPro ~ % cd desktop
(base) kadirseker@KadirMacPro desktop % javac HelloWorld.java
(base) kadirseker@KadirMacPro desktop % java HelloWorld
Hello, World!
(base) kadirseker@KadirMacPro desktop % █
```

JAVA DEĞİŞKENLER

- ▶ **boolean** : true VS false (0 VS 1)
- ▶ **byte** : -128 to 127 (8-bit signed two's complement integer)
- ▶ **short** : -32768 to 32767 (16-bit signed two's complement integer)
- ▶ **integer** : (-2^{31}) to $(2^{31}-1)$ (32-bit signed two's complement integer)
- ▶ **long** : 0 to $(2^{64}-1)$ (64-bit signed two's complement integer) **(JAVA 8)**
- ▶ **double** : 64 bit kayan sayılar (double precision)
- ▶ **float** : 32 bit kayan sayılar (single precision)
- ▶ **char** : 16-bit Unicode character
- ▶ **string** : Strings in Java are not primitive types

JAVA İLE AKIŞ KONTROLÜ

- ▶ Temelde 2 şartlı durumları kontrol etmek için “if statement” kullanılır.
- ▶ Çoklu şartlı durumlar için “switch statement” kullanılır.
- ▶ Tekrar eden işlemler için döngüler kullanılır. “for and while loops”
- ▶ Belirli durumlara özgü kontrol işlemi için “break and continue statements” kullanılır.

JAVA İLE AKIŞ KONTROLÜ-2

► break and continue statements

```
while (testExpression) {  
    // codes  
    if (condition to break) {  
        break;  
    }  
    // codes  
}
```

```
do {  
    // codes  
    if (condition to break) {  
        break;  
    }  
    // codes  
} while (testExpression);
```

```
for (init; testExpression; update) {  
    // codes  
    if (condition to break) {  
        break;  
    }  
    // codes  
}
```

```
while (testExpression) {  
    // codes  
    if (testExpression) {  
        continue;  
    }  
    // codes  
}
```

```
do {  
    // codes  
    if (testExpression) {  
        continue;  
    }  
    // codes  
} while (testExpression);
```

```
for (init; testExpression; update) {  
    // codes  
    if (testExpression) {  
        continue;  
    }  
    // codes  
}
```

JAVA ARRAYS

- ▶ Array (dizi) ortak bir isimle gösterilen benzer türdeki değişken grubu
- ▶ Tek boyutlu dizi tanımlama : “ type_of_array variable_name”. (int month_days[]);
- ▶ “new” anahtar kelimesi ile diziye fiziksel olarak bir yer tahsis edilir*. (month_days = new int [12])
- ▶ Dizi elemanlarına değerler atanır. (month_days[0] = 31, month_days[1] = 28, ...)
- ▶ Bu iki işlem birleştirilebilir. (int month_days[] = new int[12];)
- ▶ Değer atama işlemi de bu aşamada yapılabilir. (int month_days[] = { 31, 28, 31, 30, 31, ... };)

JAVA ARRAYS-2

► iki boyutlu dizi tanımlama

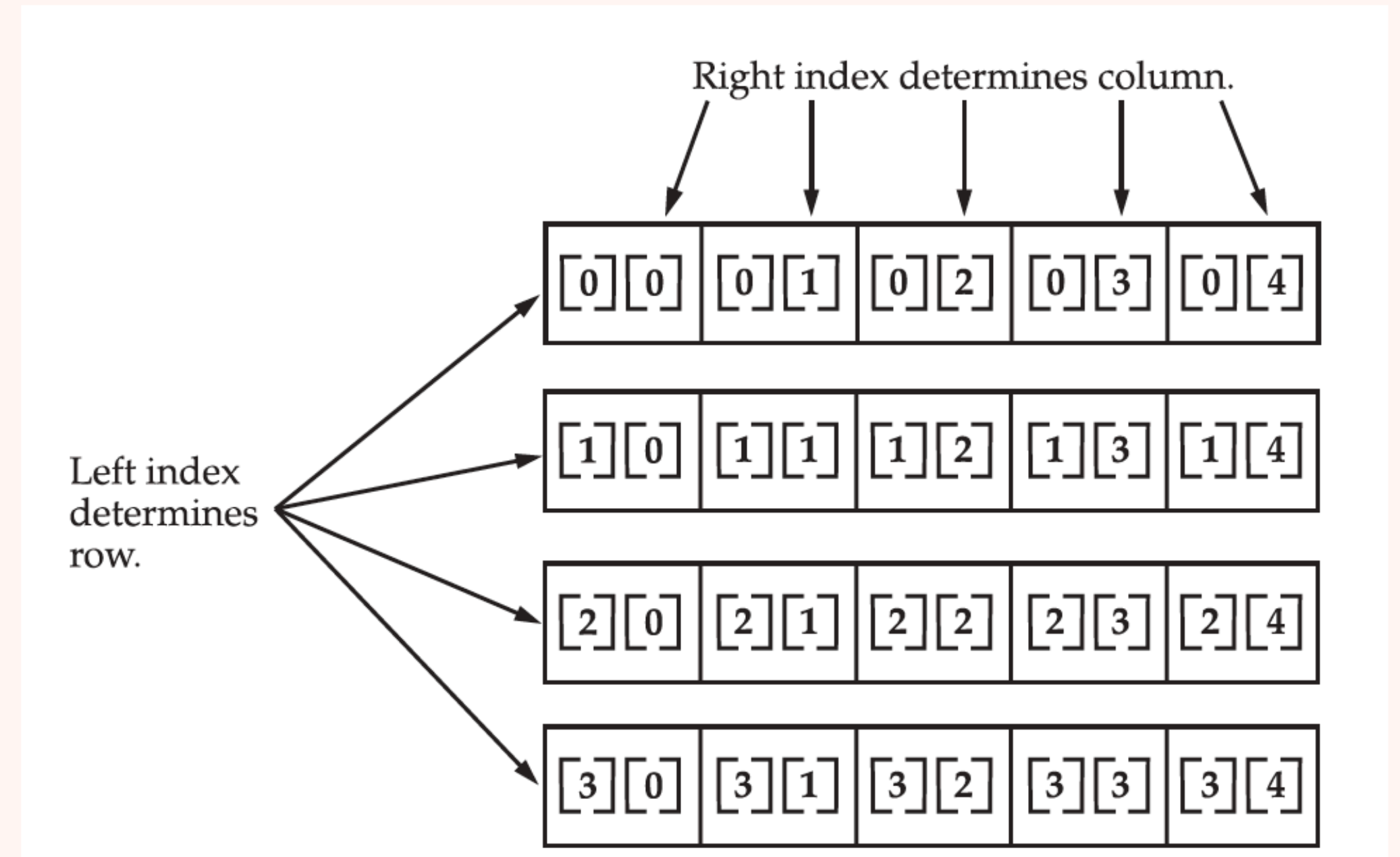
► 4 adet 5 uzunluklu integer dizi :

► `int twoD [][] = new int [4][5];`

► `int [][] twoD = new int [4][5];`

► sol index, satırları ; sağ index, sütunları gösterir.

► Diziye değer atamak için döngüler kullanılır.



JAVA DEBUG

- ▶ 3 tip hata ile karşılaşabilirsiniz;
 - ▶ Sözdizimi hatası (compile-time error)
 - ▶ Çalışma zamanı hatası
 - ▶ Mantıksal hata (Bug)
- ▶ IntelliJ debug mode;
 - ▶ breakpoint: incelenecek satır
 - ▶ step over; bir alt satıra atlar.
 - ▶ resume program; bir sonraki breakpoint e atlar.
 - ▶ step into: metodların içine girer.

