YALOVA UNIVERSITY

HUAWEI

Android Programming with Huawei Mobile Services

Yunus OZEN
Muhammed Salih KARAKASLI
Cenk Faruk CAVGA
Sezer Yavuzer BOZKIR

JAVA

Introduction

- What is JAVA?
- Java Development Kit(JDK)
- Java Runtime Environment(JRE)
- Java Syntax

Java Basics

- Variables
- Privacy
- Data Types
- Operators

Control Statements

- If-else
- Switch-Case
- For loop
- While loop
- Do-while loop
- Continue statement
- Break statement

Object Oriented Programming

- Inheritance.
- Overriding
- Abstraction
- Interfaces
- Polymorphism
- Encapsulation

Introduction

JAVA

cross-platform
object-oriented
programming language

It was originally developed by James Gosling at Sun Microsystems and released in 1995.

Java with Numbers

5 million
students study Java

12 million
developers run Java

#1
programming language

#1 developer
choice for cloud

Check the offical web page: go.java

JAVA DEVELOPMENT KİT(JDK)

In order to create, compile and run Java program you would need **JDK** installed on your computer.

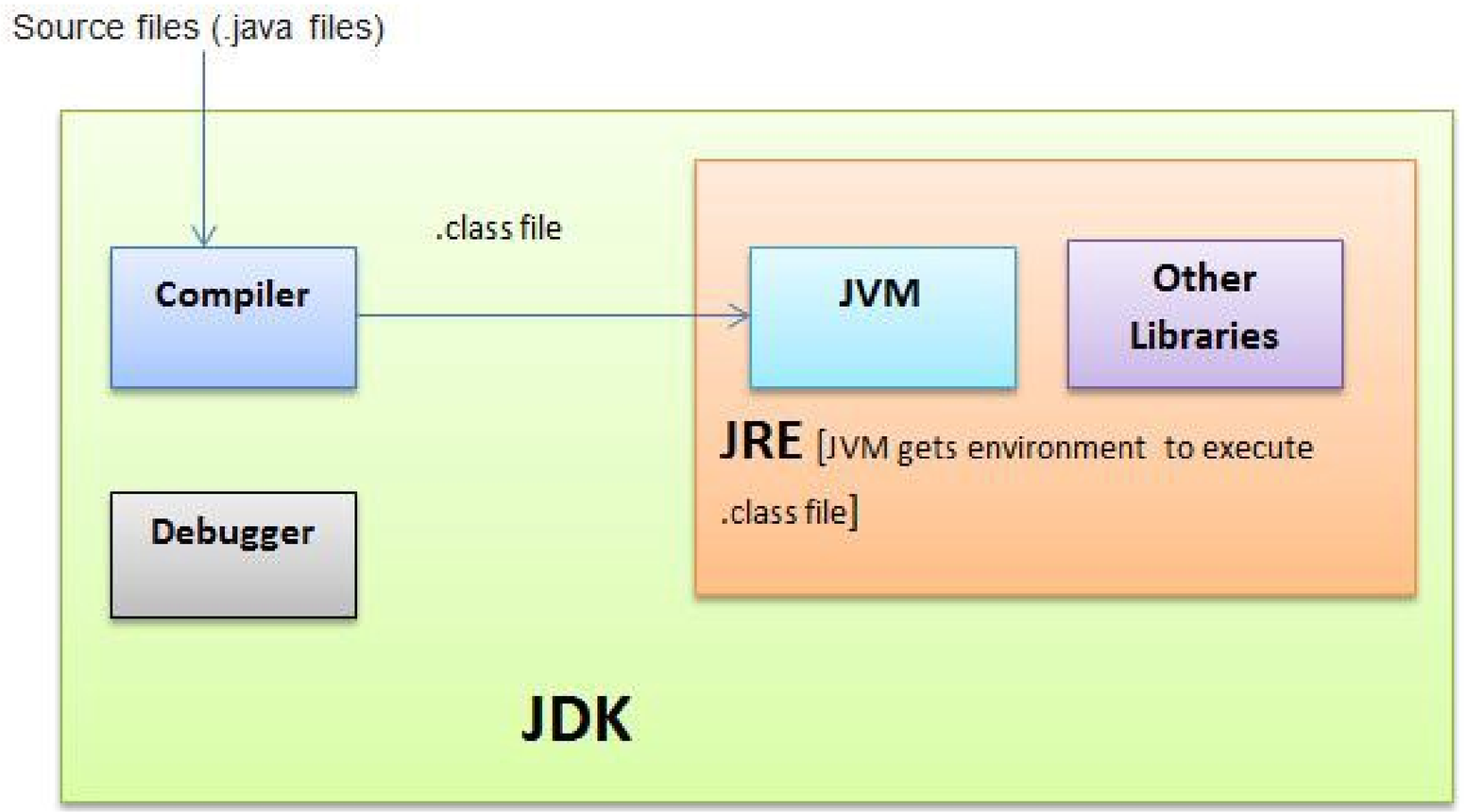
JAVA RUNTİME ENVİRONMENT(JRE)

JRE is a part of JDK which means that JDK includes JRE. When you have **JRE** installed on your system, you can run a java program however you won't be able to compile it. **JRE** includes JVM, browser plugins and applets support. When you only need to run a java program on your computer, you would only need **JRE**.

JAVA VİRTUAL MACHİNE (JVM)

JVM executes the bytecode produced by compiler. Each operating system has different JVM, however the output they produce after execution of bytecode is same across all operating systems. That is why we call java as platform independent language.

SIMPLIFIED ARCHITECTURE OF JDK



JAVA SYNTAX

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

Try First Code

Java Basics

VARİABLES İN JAVA

How to Declare a variable in Java?

data_type variable_name = value;

StudentObject studentObject = value;

VARİABLES İN JAVA

How to Declare a variable in Java?

```
char ch = 'A';
int number = 100;
```

or we can do it like this:

```
char ch;
int number;
...
ch = 'A';
number  = 100;
```

VARİABLES NAMİNG

- Variables naming cannot contain white spaces,
- Variable name can begin with special characters such as $ and _
- As per the java coding standards the variable name should begin with a lower case letter
- Variable names are case sensitive in Java.

TYPES OF VARİABLES İN JAVA

- Local variable
- Static (or class) variable
- Instance variable

LOCAL VARİABLE

```
public void myMethod(){
      // local variable
      String myVar = "Inside Method";
      System.out.println(myVar);
}
```

STATİC (OR CLASS) VARİABLE

```
public class StaticVarExample {
   public static String myClassVar = "class or static variable";
}
```

INSTANCE VARİABLE

```
public class InstanceVarExample {
   int instanceVar= 15;
}
```

PRIVACY

- public
- protected
- private

PRİMİTİVE DATA TYPES

In Java, we have eight primitive data types: **boolean**, **char**, **byte**, **short**, **int**, **long**, **float** and **double**.
Java developers included these data types to maintain the portability of java as the size of these primitive data types do not change from one operating system to another.

Type	Description	Default	Size	Example Literals
boolean	true or false	false	1 bit	`true, false`
byte	twos complement integer	0	8 bits	(none)
char	Unicode character	`\u0000`	16 bits	`'a', '\u0041', '\101', '\\', '\'', '\n', 'ß'`
short	twos complement integer	0	16 bits	(none)
int	twos complement integer	0	32 bits	`-2, -1, 0, 1, 2`
long	twos complement integer	0	64 bits	`-2L, -1L, 0L, 1L, 2L`
float	IEEE 754 floating point	0.0	32 bits	`1.23e100f, -1.23e-100f, .3f, 3.14F`
double	IEEE 754 floating point	0.0	64 bits	`1.23456e300d, -1.23456e-300d, 1e1d`

OPERATORS İN JAVA

- Basic Arithmetic -> { **+ - * / %** }
- Assignment -> { **= += -= *= /= %=** }
- Auto-increment & decrement -> { **++ --** }
- Logical Operators -> { **&&, ||, !** }
- Comparison (relational) -> { **==, !=, >, <, >=, <=** }
- Bitwise -> { **&, |, ^, ~, <<, >>** }
- Ternary ->

(expression) ? value if true : value if false

Control Statements

IF, IF..ELSE STATEMENT

```
int num=70;
if( num < 100 ){
    /* This println statement will only execute,
     * if the above condition is true
     */
    System.out.println("number is less than 100");
}
```

```
int num=120;
if( num < 50 ){
   System.out.println("num is less than 50");
}
else {
   System.out.println("num is greater than or equal 50");
}
```

SWİTCH CASE

```
int num=2;
switch(num+2)
{
   case 1:
     System.out.println("Case1: Value is: "+num);
   case 2:
     System.out.println("Case2: Value is: "+num);
   case 3:
     System.out.println("Case3: Value is: "+num);
   default:
     System.out.println("Default: Value is: "+num);
 }
```

FOR LOOP

What is the output?

```
for(int i=10; i>1; i--){
      System.out.println("The value of i is: "+i);
}
```

```
String arr[]={"hi","hello","bye"};
for (String str : arr) {
        System.out.println(str);
}
```

INFINITE LOOP

```
for(int i=1; i>=1; i++){
        System.out.println("The value of i is: "+i);
}
```

WHİLE LOOP

```
int i=10;
while(i>1){
        System.out.println(i);
        i--;
}
```

DO-WHİLE LOOP

```
        int i=10;
        do{
              System.out.println(i);
              i--;
        }while(i>1);
}
```

CONTİNUE STATEMENT

What is the output?

```
for (int j=0; j<=6; j++)
{
    if (j==4)
    {
        continue;
    }

    System.out.print(j+" ");
}
```

BREAK STATEMENT

What is the output?

```
    int num =0;
    while(num<=100)
    {
        System.out.println("Value of variable is: "+num);
        if (num==2)
        {
           break;
        }
        num++;
    }
    System.out.println("Out of while-loop");
}
```

Object Oriented Programming (OOP)

INHERITANCE

What is inheritance?

INHERITANCE

The aim of **inheritance** is to provide the reusability of code so that a class has to write only the unique features and rest of the common properties and functionalities can be extended from the another class.

Child Class:

The class that extends the features of another class is known as child class, sub class or derived class.

Parent Class:

The class whose properties and functionalities are used(inherited) by another class is known as parent class, super class or Base class.

INHERITANCE

Let's try an example

OVERRİDİNG

Let's talk about it

OVERRİDİNG

Let's code it

ABSTRACTION

```
abstract class Human {
    abstract void speak();
}
class British extends Human {
    @Override
    void speak() {
        System.out.println("Hi");
    }
}
class Turkish extends Human {
    @Override
    void speak() {
        System.out.println("Merhaba");
    }
}
```

INTERFACE

Interface looks like a class but it is not a class.
An interface can have methods and variables just like the class
but the methods declared in interface are by default abstract.
Also the variables declared in an interface are public, static & final by default. .

POLYMORPHİSM

```
abstract class Human {
    abstract void speak();
}
class British extends Human {
    @Override
    void speak() {
        System.out.println("Hi");
    }
}
class Turkish extends Human {
    @Override
    void speak() {
        System.out.println("Merhaba");
    }
```

ENCAPSULATİON

The whole idea behind **encapsulation** is to hide the implementation details from users . If a data member is private it means it can only be accessed within the same class. No outside class can access private data member (variable) of other class.

LET'S TALK
CONTACT INFORMATİON

Muhammed Salih KARAKASLI
muhammed.salih.karakasli@huawei.com

Cenk Faruk CAVGA
cenk.faruk.cavga@huawei.com

Sezer Yavuzer BOZKIR
sezer.bozkir@huawei.com

Telegram Channel
will be created

Official Website
https://developer.huawei.com