JAVA Simplica

1st Edition, Sudipta Kumar Das

2022, Publisher

Contents

List of Figures

List of Tables

Part I

Introduction

JAVA[1] is a Programming language which is used mostly in offial softwares because of it's strong security system. It is a high-level language which uses JVM to convert the high-level code to a machine code. It is one of the most popular programming languages out there. Released in 1995 and still widely used today. Java has many applications, including software development, mobile applications, and large systems development. Knowing Java opens a lot of possibilities for us as a developer.

Preface

JAVA knowledge is vast. People most often have to go through most of the documentations of the code then they could think of writing something. Moreover, sometimes people looses their interest in learning JAVA or write their codes in JAVA. So in that case they just give online posts and hire outsourcers to complete there school/college projects howmeworks and others. This processs is both insecure and costly. In this book I just tried to teach JAVA in a simple way and by which people can start doing their school/college projects howmeworks and others by their own having simple knowledge. Thus, they can learn the vast knowledge slowly and more interesting way.

Chapter 1

History of JAVA

Java was originally developed by James Gosling[2] at Sun Microsystems and released in May 1995 as a core component of Sun Microsystems' Java platform. The original and reference implementation Java compilers, virtual machines, and class libraries were originally released by Sun under proprietary licenses. As of May 2007, in compliance with the specifications of the Java Community Process, Sun had relicensed most of its Java technologies under the GPL-2.0-only license. Oracle offers its own HotSpot Java Virtual Machine, however the official reference implementation is the OpenJDK JVM which is free open-source software and used by most developers and is the default JVM for almost all Linux distributions.

Figure 1.1: James Gosling

Part II

Pre-Basic of JAVA

JAVA is a vast programming language, but it has some pre basic things, on whichs the whole language depends on. In this part we'll going to discuss It

Chapter 2

Package & Class Declaration

2.1 package

Package is kind of a folder, where all the class files are present. we can use them by using the keyword *import packageName.subPackageName.className* or *import packageName.**. Here * means all the things. we can use predefined packages of jdk or we can also import out own packages in any class from another folder.

2.1.1 Syntax

```
import packageName.subPackageName.className
```

2.1.2 Example

```
java.io.File;
```

2.2 Access modifiers

Access modifires basically used to control the access of the variables & methods form another class or package. It is mostly used in Encapsulation. There are basically 4 Access modifiers. Those are,

- Public
- Private
- Protected
- Default

2.2.1 Public

Public Keyword is used to make the variables and methods Public that means those thing can be access from anywhere, no matter where it is.

2.2.2 Private

Private Keyword is used to make the variables and methods inaccessible that means those thing can be access from nowhere, no matter where it is.

2.2.3 Protected

Protected Keyword is used to make the variables and methods only accessible from their children that means those thing can be access from nowhere except its child class, no matter where it is. IF a class is extended by another class then the class who extend in it, called child class of the class who got extended by the child class. And that class who got extended by the child class called parent Class.

2.2.4 Default

We don't need any access modifiers to make it default access. Default access is kind of private access modifier. Default access means that variable/methods can be accessible from anywhere inside the folder its in. And can not be accessible outside of the folder.

2.2.5 Syntax

Access_modifier dataType/returnType variableName/methodName()

2.2.6 Example

```
public boolean isAccessible = true;
private String name = "Sudipta Kumar Das";
protected String carModel = "Toyota CHR";
int age = 22; \\ This is Default Access Modifier
```

2.3 Class Declaration

JAVA is an Object Oriented Programming(OOP) Language. Here we have to use lots of classes. To use classes we have to declare it. Class declaration has its own syntax

2.3.1 Syntax

Access_modifier class className

2.3.2 Example

```
public class Mobile{

}
```

2.4 Main Method

JAVA is a high level language. It needs a compliler to convert the high level code into machione code. The compilers need to understand the starting point of the code conversion. Main method is the place from where the compilers start reading and start compiling. There should be only one main method for entire program or project. Classes can be many but main method must be one. main method is declared inside any one class.

2.4.1 Syntax

public static void main(String[] args){}

2.4.2 Example

```
public class Test{
    public static void main(String[] args){

    }
}
```

2.5 Show Output in JAVA

We use *System.out.println();* to print anything or show anything on console. Here println means print a newline also. That means the line will break and go to a new line after showing the output inside first bracket.

2.5.1 Syntax

System.out.println();

2.5.2 Example

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

```
}
```

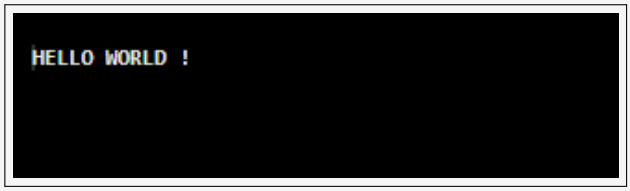

Figure 2.1: Show Output in JAVA[3]

Chapter 3

Escape Sequence & Format Specifier

3.1 Escape Sequence

Escape sequences[4] are some special characters who perfoms some special kinds of works on showing console output as like printing a backslash or a new line. Escape sequences are written after a backslash indicating it is a special character. And it is been written inside double quote marks("").

Escape Sequence	Meaning
\b	Backspace
\t	Tab (4 spaces at right)
\n	New Line/Break Line
\r	Carriage Return/ Break line & tart from the left most after this line
\"	Print Double quote mark on console
\'	Print Single quote mark on console
\f	Insert a form feed in the text at this point.
\\	Print Backslash on console

Table 3.1: Escape Sequences

3.1.1 Syntax

"\escapeCharacter"

3.1.2 Example

```
public class Test {
```

```
    public static void main(String args[]) {
        System.out.println("HELLO\b WORLD !");
        System.out.println("HELLO\t WORLD !");
        System.out.println("HELLO\n WORLD !");
        System.out.println("HELLO\r WORLD !");
        System.out.println("HELLO \"WORLD\" !");
        System.out.println("HELLO \‘W\’ORLD !");
        System.out.println("HELLO\f WORLD !");
        System.out.println("HELLO \\WORLD !");
    }
}
```

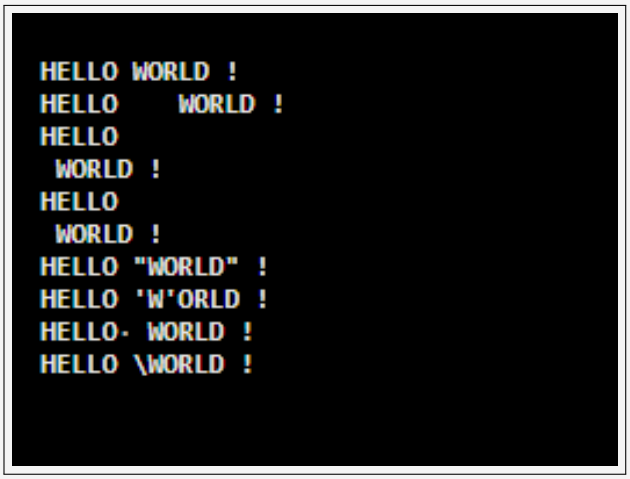

Figure 3.1: Escape Sequences[4][3]

3.2 Format Specifier

Format Specifier[5] is used to indicate the place where the value of a variable should appear in a string. That means sometimes we have to show the output inside a line, as like. Hii! I am {age} year old. here we want age = 22 or something just like that. So we'll write System.out.println("Hii! I am %d year old.",age); Here output will be if age = 22, Hii! I am 22 year old.

3.2.1 Syntax

"%formatSpecifier"

Format Specifier	Usual Variable Type	Display As
%f%f	float or double	Signed Decimal
%o	int	unsigned Octal value
%u	int	unsigned Integer
%x	int	unsigned Hex value
%H	int	unsigned Decimal Integer
%S	array of char	Sequence of Characters
%%	-	Inserts a % sign
%f	float	Decimal floating-point
%e%E	-	Scientific Notation / Exponential Format
%g	-	Causes formatter to use either %f or %e which one is shorter
%h%H	-	Hash code of the Argument
%d		Decimal Integer
%c		Character
%b%B	boolean	Boolean
%a%A	-	Floating Point hexadecimal

Table 3.2: Format Specifier

3.2.2 Example

```
public class Test {
    public static void main(String args[]) {
        int i = 1234567890;
        boolean b = true;
        char c = 'a';
        short s = 12345;
        float f = 10.2f;
        double d = 344.659;
        System.out.printf("boolean b = %b\n",b);
        System.out.printf("charater c = %c\n",c);
        System.out.printf("short s = %d\n",s);
        System.out.printf("integer i = %d\n",i);
        System.out.printf("float f = %1f\n",f);
        System.out.printf("double d = %3f\n",d);
    }
}
```

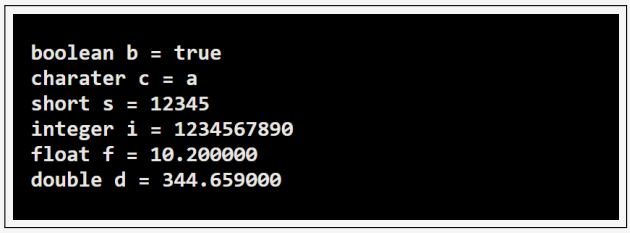

Figure 3.2: Format Specifier[5][3]

Index

Bibliography

[1] Java, https://en.wikipedia.org/wiki/Java_(programming_language)

[2] James Gosling, https://en.wikipedia.org/wiki/James_Gosling

[3] Java Compiler, https://www.jdoodle.com/online-java-compiler/

[4] Escape Sequence, https://docs.oracle.com/javase/tutorial/java/data/characters.html

[5] Specifier, https://www.geeksforgeeks.org/format-specifiers-in-java/