Introduction of JAVA

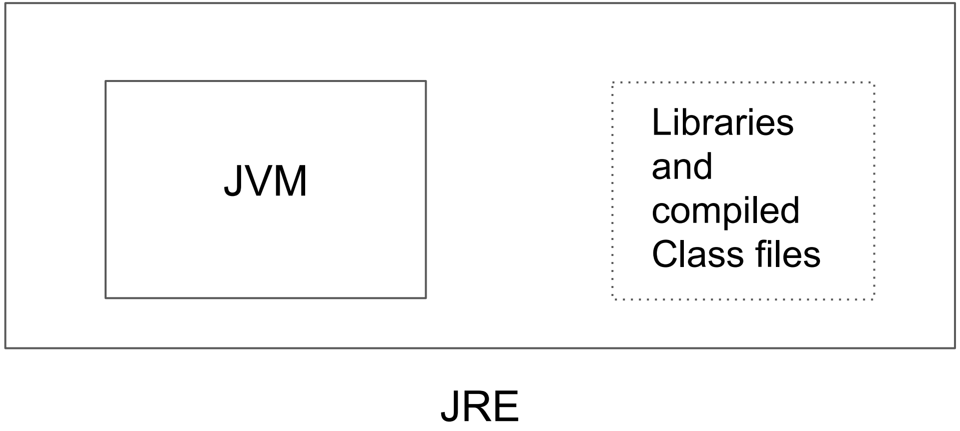
Dhvani Undhad

Main features of JAVA

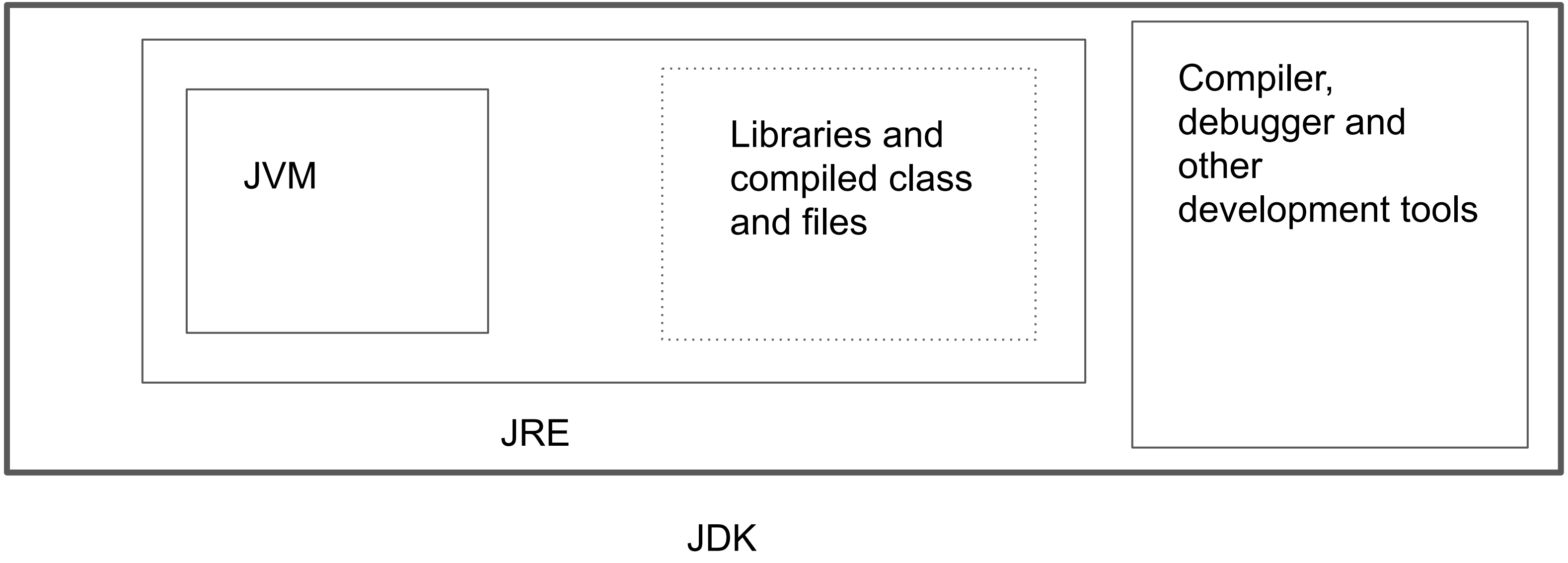
- **Platform independence**
- Java is s popular language used for Mobile applications, Desktop applications, Web application, Web servers and application servers, Games, Database connection and many more.
- **Java is open-source and free.**
- J**ava is object oriented programming language** which gives the clear structure to programs and allows code to be reused. Which includes below 4 main concepts of OOPs: **Abstraction , Encapsulation,Inheritance and Polymorphism.**

JAVA Terminologies

1. Java Virtual Machine (JVM): JVM runs the program by using class, libraries and files which JRE provided. JVM executes the byte code generated by compiler and produce output. JVM is the one that makes a java platform independence.



2. Java Runtime Error (JRE): JRE is the environment within which JVM runs. JRE contains Java Virtual Machine, class , libraries and other files. Which means you can run the code in JRE but can not develop or compile code in JRE without JVM.



JVM

Libraries and compiled class and files

Compiler, debugger and other development tools

JRE

JDK

3. Java Development Kit (JDK): It’s a software development environment used for developing Java applications and applets. It includes JRE, a compiler, jar files, and other tools needed in java program development.

Variables in JAVA

A variable is a name which contains specific values that can be changed in JAVA.

There are three types of variables in JAVA:

Local Variable

This variables are declared inside the method of the class. Their scope is limited to the method, which means we can not change it’s value.

Static Variable

Also know as Class Variables because they are associated with all class and common for all instance of class.

Instance Variable

Unlike Static variables, Instance variables have their own separate copy of instance variable. So in this case we can change value of instance variable.

Data types in JAVA

Data types defines the values that a variable can take.

In java there are two categories in Data Types:

Primitive Data-Types

byte
short
int
long
float
double
char
boolean

Non Primitive Data-types

Array
Strings
Class
Interface

Operators in JAVA

- An operator is an character that represents an action
- Types of Operator:

Arithmetic Operator (+, -, *, /, %)

Relational Operator (>, <, >=, <=, ==, !=, !)

Conditional operator (? :)

Logical Operator (&, |, &&, ||, ^)

Unary Operator : Increment and Decrement

Control statements in JAVA

Four types of control statements are available in JAVA:

1. **If statement**

```
if(condition){
   statements;
}
```

Ex:

```
if (20 > 18) {
  System.out.println("20 is greater than
18");

 }
```

2. **Nested if statement**

```
if(condition){
   statement1;
       if(condition){
          Statements;
       }
}
```

```
Ex:

if( num < 100 ){
           System.out.println("number is
less than 100");
           if(num > 50){
            System.out.println("number is
greater than 50");
        }
    }
```

3. If-else statement

```
if(condition){
   statements;
}
else{
statements;
}
```

```java
int time = 20;
if (time < 18) {
  System.out.println("Good day.");
} else {
  System.out.println("Good evening.");
  }
```

4. If- else-if statement

```
if(condition){
   statements;
}
else if {
   statements;
}
else if {
   statements;
}
else {
  Statements;
}
```

```
int time = 22;
if (time < 10) {
  System.out.println("Good
morning.");
} else if (time < 20) {
  System.out.println("Good
day.");
} else {
  System.out.println("Good
evening.");

 }
```

Switch case statement in JAVA

- **Switch case statement** is used when we have number of options (or choices) and we may need to perform a different task for each choice.

```
switch (expression){
    case x:
        break;
    case y:
        break;
    Default:
    }
```

Loops in Java

- Loops are used to execute a set of statements repeatedly until a particular condition is satisfied.
- In Java we have **three types of basic loops:**

1. **For loop**

```
for(initialization; condition ; increment){
        statements;
}
```

Ex:

```java
for (int i = 0; i < 5; i++) {
  System.out.println(i);
}
```

2. while loop

```
while (condition){

}
```

Ex:

```
int i = 0;
while (i < 5) {
  System.out.println(i);
  i++;

 }
```

3. Do while loop

```
do{
}
while(condition);
```

Ex:

```
int i = 0;
do {
  System.out.println(i);
  i++;
}
while (i < 5);
```

Continue and break Statements in JAVA

1. **Continue statement** is mostly used inside loops. Whenever it is encountered inside a loop, control directly jumps to the beginning of the loop for next iteration, skipping the execution of statements inside loop’s body for the current iteration. This is particularly useful when you want to continue the loop but do not want the rest of the statements(after continue statement) in loop body to execute for that particular iteration.

- Syntax:

Continue;

Ex:
```
for (int i = 0; i < 10; i++) {
  if (i == 4) {
    continue;
  }
  System.out.println(i);
}
```

2. break statement: The break statement used to jump out of the loop instantly.

- Syntax:

Break;

Ex:

```
for (int i = 0; i < 10; i++) {
  if (i == 4) {
    break;
  }
  System.out.println(i);
}
```