Java: features

* Java is the platform Independent.
* Java is the Object-Oriented language.
* Java is a multi-Threading.
* Java is secure language.
* Java is an interpreter language.

JDK, JRE, JVM

JDK: java development kit 🡺 Java software

JDK = JRE + JVM

JRE 🡺 Java runtime Environment

It is used to run the code.

JRE = JVM + libraries

JVM🡪java virtual machine: .java converts to .class file (byte code) and these byte code can be run different platform (windows, IOS, Linux)

**JVM is platform dependent**🡪 it requires one platform to convert into byte code.

**Java is platform independent**

Src: source folder🡪 .java

bin: binary 🡪 .class

ClassPath vs Path:

Classpath 🡺 all the .class files will be stored in a folder and that folder path is called classpath.

Path 🡺 it is a java path where java software path is set in environment variables.

Compile time: .java to .class file conversion time🡺 compile time errors/exceptions 🡺 compiler

Runtime🡺 .class execution 🡺 runtime exceptions

…..

…..

Hi();

….

……….

……..

HI();

………

Hi(){

\*\*\*\*\*

}

8Diagram

Description automatically generated

**Class**:

It is a template which consists methods, objects, constructors, variables.

**Method**:/ Functions

It is used to write logics inside the method.

With parameters and without parameters

**Object**:🡺 **new** keyword

It is used to store the data/ represent data/value

Call the methods

Its represent the behavior/ functionality

Object stores in heap memory

**Constructor**:

Class instance

With parameters and without parameters

* Class name starts with Capital letter. 🡺 Hi

**Public class Hi { // class**

**Hi(int id, String name){ // constructor**

**…**

**…**

**}**

**Hi() {**

**// logic**

**}**

**Hi h = new Hi(5, “test”); 🡪 Object**

**Hi h = new Hi();🡪 Object**

**Public int display(){ -🡪 method1**

**}**

**Public void display1(){ -🡪 method2**

**}**

**Public void display2(int num){ -🡪 method3**

**}**

**}**

**Variables**: it is used to hold data temporary.

3 types

Local: within the method

Instance: outside the methods and inside class

Static: outside the method and inside class uses **static** keyword

Example ex = new Example();

ex 🡪 variable

int a = 10;

a = variable

int = data type

10 = data

boolean b = true;

Public class Hello {

Static Int I = 10;

Hello hlw = new Hello();

hlw.print();

Public Hello(){

}

}