
Java™

1. Introduction to Programming

Types of Language

Procedural

- specifies a series of well structured steps and procedures to compose a program
- contains a systematic order of statements, function and commands to complete a task.

Functional

- Writing a program only in pure functions, i.e. never modify variables, but create new ones as an input.
- Used mainly when we have to perform different operations on the same set of data.

Object Oriented

- revolves around objects.
- code + data = object
- developed to make it easier to develop, debug, reuse and maintain software.

What is a first class function?

A programming language is said to have First-Class Functions, when functions in that language can be treated as a variable.

Static Language	Dynamic Language
1. Performs type checking at compile time.	1. Performs type checking at runtime.
2. Errors will show at compile time.	2. Error might not show, till the program is run.
3. Declare datatypes before you use it.	3. No need to declare datatype of variables.
4. More control.	4. Saves time in writing code, but might give error.

Memory Management

A programming language has two types of memory, (i) stack memory, (ii) heap memory.

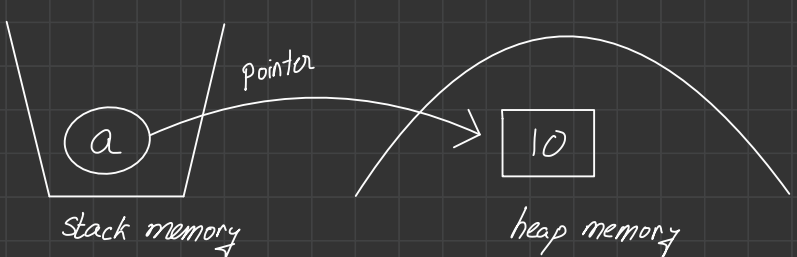
Let us consider a case, where we set a value to a variable.

```
a=10;
```

Reference variable → a

Object → 10

So in this case, the reference variable is stored in a stack, and the object is stored in a heap.



Now, if there is a scenario, where multiple reference variables point to the same object, and if there is a change in the original object, then the change will be seen by all the reference variables.

Ex:

```
int a[]= {1,2};
int b[] = a;
a[0] = 6;
```

→ In this scenario, the initial object of a is something. Then another another variable is also pointing to this same object. Now if the main object is modified, it will effect both the reference variables.

What is Garbage Collection in Java?

Java Application obtain objects in memory as needed. It is the task of the Garbage Collection (GC) in the Java Virtual Machine (JVM) to automatically determine what memory is no longer used by a Java Application and to recycle this memory for other users.

Ex:

```
int a = 10;
a = 20;
```

→ In this example, reference variable 'a' points to object later, this reference variable points to another object. And the previous object is left unused. So, when the Garbage Collection is called, this unreferenced object is basically destructed.

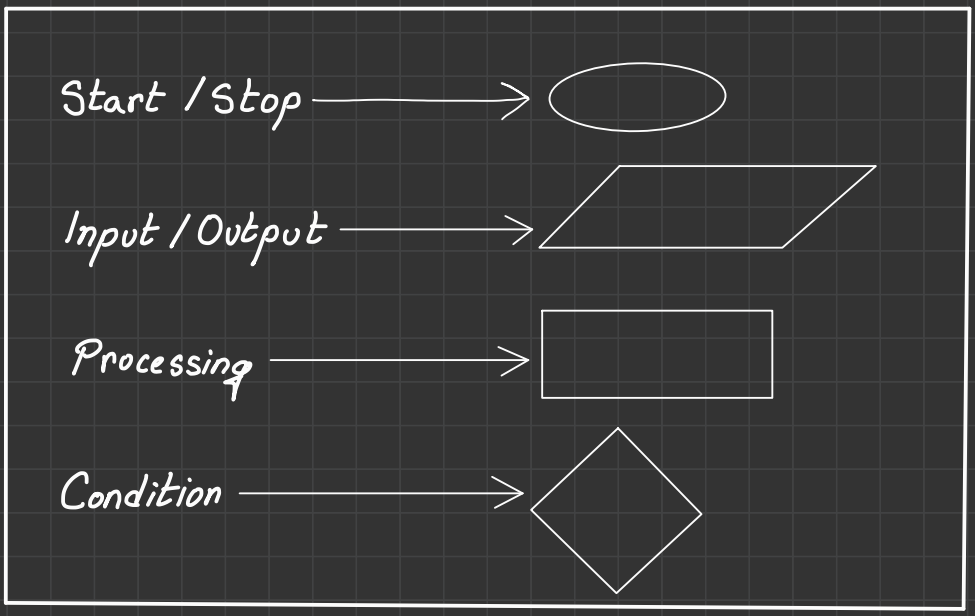
How to manually call the Garbage collection?

So, manually, the garbage collection can be invoked the following two ways:

- System.gc();
- Runtime.getRuntime().gc();

2. Flow of Program

Flowchart



A flowchart, is basically a diagramatic process, which details how a certain problem can be solved.

While, pseudocode is a rough logical written version.

3. Introduction to Java-Architecture

Java code execution

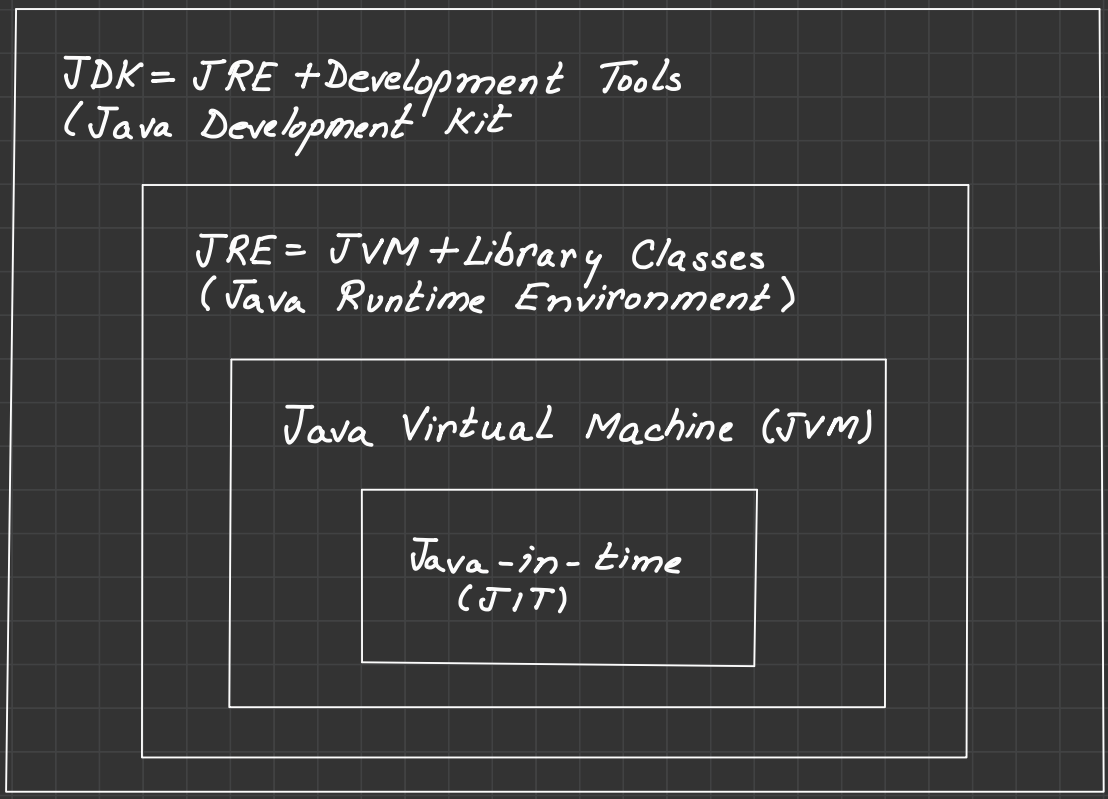
this is the source code

- this code will not directly run on the system
- JVM is needed to run this code
- This is reason, why Java is platform independent.

More about platform independence

- It means that byte code can run on all operating systems.
- We need to convert source code to machine code, so computer can understand
- Compiler helps in doing this by turning it into executable code
- This executable code is a set of instructions for the computer.
- After compiling C/C++ file, we get a .exe file, which is platform dependent.
- In Java we get bytecode, JVM converts this to machine code.
- Java is platform-independent, but JVM is platform dependent.

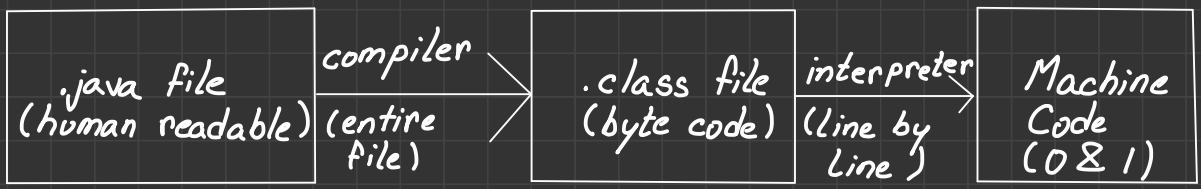
JDK vs JRE vs JVM vs JIT



JDK

- Provides environment to develop and run the Java program.
- It is a package that includes:
  1. Development Tools - To provide an environment to develop your program.
  2. JRE - To execute your program
  3. a compiler - javac
  4. archiver - jar
  5. docs generator - javadoc
  6. interpreter / loader

How Java code executes



This is the source code

- this code will not directly run on a system.
- we need JVM to run this
- Reason why Java is platform independent

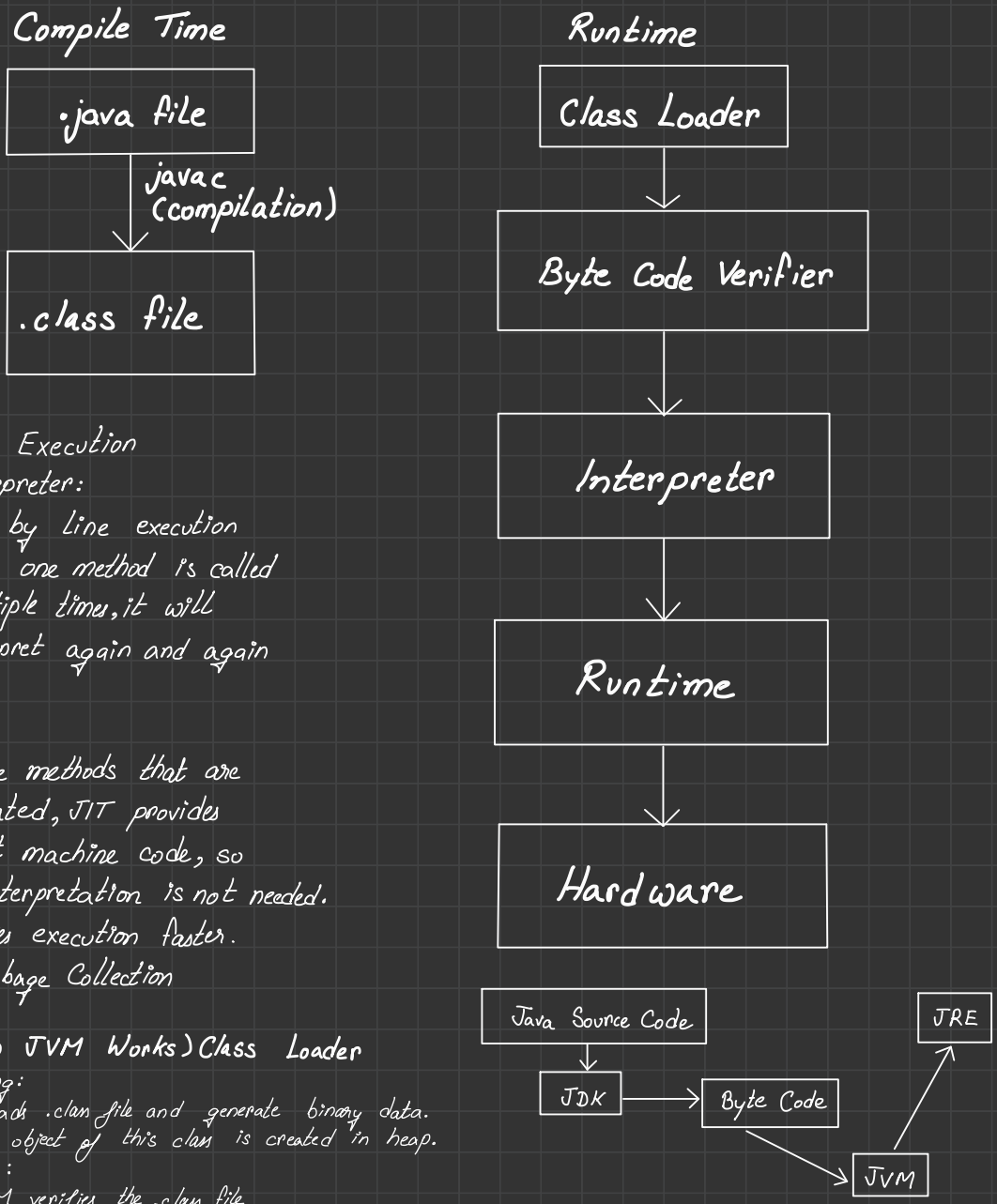
JDK

- Provides environment to develop and run the Java Program
- It is a package that includes:
  1. Development Tools - To provide an environment to develop your program
  2. JRE - to execute your program
  3. a compiler - javac
  4. archiver - jar
  5. docs generator - javadoc
  6. interpreter / loader

JRE

- It is an installation package that provides environment to only run the program
- It consists of:
  1. Deployment Technologies
  2. User interface toolkits
  3. Integration Libraries
  4. Base Libraries
  5. JVM

- After we get the .class file, the next things happen at runtime:
  1. Class loader loads all classes needed to execute the program.
  2. JVM sends code to Byte code verifier to check the format of code.



JVM Execution

Interpreter:

- Line by line execution
- When one method is called multiple times, it will interpret again and again

JIT:

- Those methods that are repeated, JIT provides direct machine code, so re-interpretation is not needed.
- Makes execution faster.
- Garbage Collection

(How JVM Works) Class Loader

- Loading:
  - Reads .class file and generate binary data.
  - An object of this class is created in heap.
- Linking:
  - JVM verifies the .class file
  - Allocates memory for class variables & default values.
  - Replace symbolic references from the type with direct references.
- Initilization
  - All static variables are assigned with their values defined in the code and static block.
- JVM contains the stack and heap memory allocations.