JDK, JRE , JVM & Memory Management in JAVA

JAVA:

Java is Object Oriented. However, it is not considered as pure object-oriented as it provides support for primitive data types (like int, char, etc)

The Java codes are first compiled into byte code (machine-independent code). Then the byte code runs on Java Virtual Machine (JVM)

HELLO WORLD PROGRAM:

```
class HelloWorld
{
    // Your program begins with a call to main().
    // Prints "Hello, World" to the terminal window.
    public static void main(String args[])
    {
        System.out.println("Hello, World");
    }
}
```

public static void main(String[] args) definition:

public: So that JVM can execute the method from anywhere.

static: The main method is to be called without an object. The modifiers public and static can be written in either order.

void: The main method doesn't return anything.

main(): Name configured in the JVM. The main method must be inside the class definition. The compiler executes the codes starting always from the main function.

String[]: The main method accepts a single argument, i.e., an array of elements of type String.

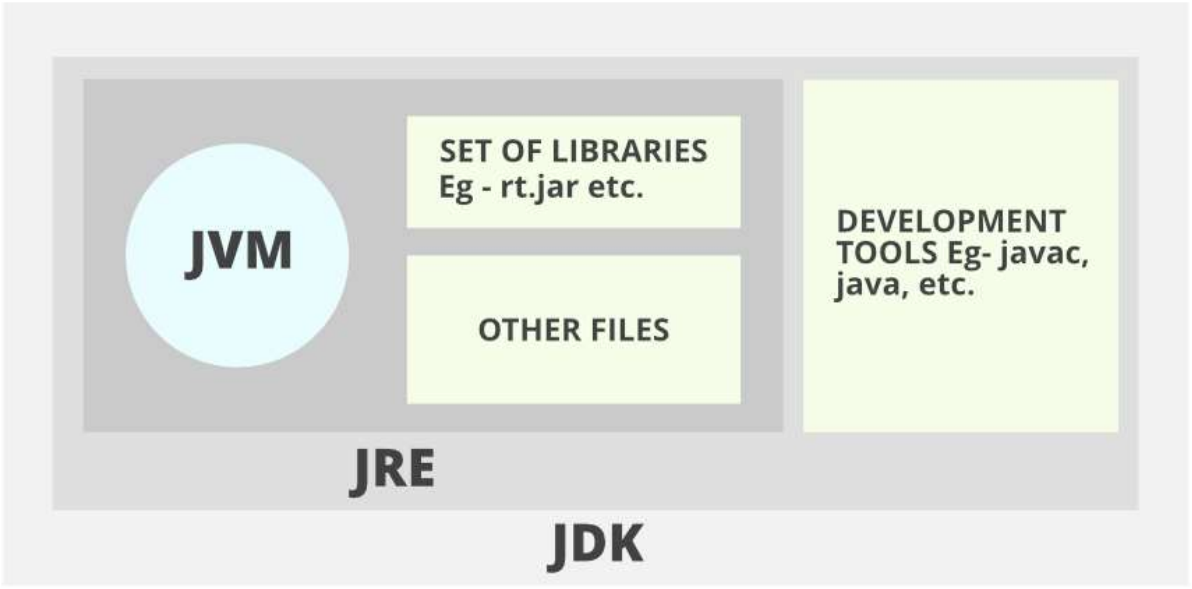
JDK, JRE AND JVM:

1. JDK (Java Development Kit) is a Kit that provides the environment to develop and execute(run) the Java program. JDK is a kit (or package) that includes two things

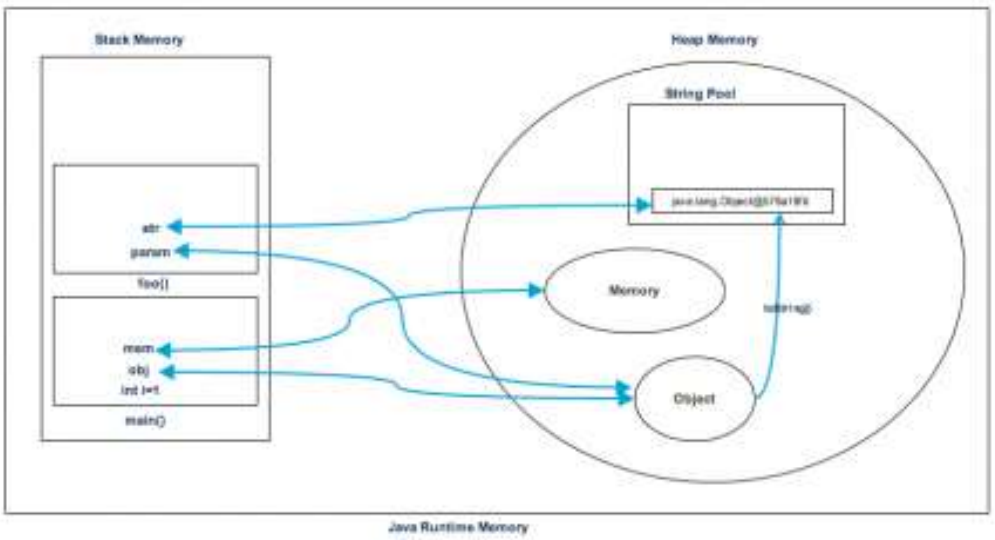
- Development Tools (to provide an environment to develop your java programs)
- JRE (to execute your java program).

2. JRE (Java Runtime Environment) is an installation package that provides an environment to only run (not develop) the java program (or application) onto your machine. JRE is only used by those who only want to run Java programs that are end-users of your system.

3. JVM (Java Virtual Machine) is a very important part of both JDK and JRE because it is contained or inbuilt in both. Whatever Java program you run using JRE or JDK goes into JVM and JVM is responsible for executing the java program line by line, hence it is also known as an interpreter.



JAVA MEMORY MANAGEMENT:



Java Stack Memory

Java Stack memory is used for the execution of a thread.

Stack memory is always referenced in LIFO (Last-In-First-Out) order.

Whenever a method is invoked, a new block is created in the stack memory for the method to hold local primitive values and reference to other objects in the method.

As soon as the method ends, the block becomes unused and becomes available for the next method. S

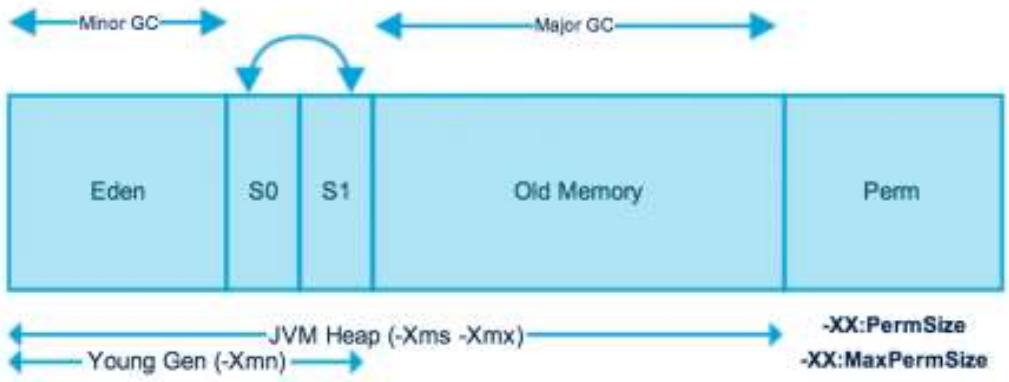
tack memory size is very less compared to Heap memory.

Java Heap Space

Java Heap space is used by java runtime to allocate memory to Objects and JRE classes.

Whenever we create an object, it's always created in the Heap space.

Garbage Collection runs on the heap memory to free the memory used by objects that don't have any reference



JVM Heap memory is physically divided into two parts - Young Generation and Old Generation.

Memory Management in Java - Young Generation

The young generation is the place where all the new objects are created. When the young generation is filled, garbage collection is performed. This garbage collection is called Minor GC. Young Generation is divided into three parts - Eden Memory and two Survivor Memory spaces.

Important Points about Young Generation Spaces:

- Most of the newly created objects are located in the Eden memory space.
- When Eden space is filled with objects, Minor GC is performed and all the survivor objects are moved to one of the survivor spaces.
- Minor GC also checks the survivor objects and move them to the other survivor space. So, at a time, one of the survivor spaces is always empty.
- Objects that are survived after many cycles of GC, are moved to the old generation memory space. Usually, it's done by setting a threshold for the age of the young generation objects before they become eligible to promote to old generation.

Memory Management in Java - Old Generation

Old Generation memory contains the objects that are long-lived and survived after many rounds of Minor GC. Usually, garbage collection is performed in Old Generation memory when it's full. Old Generation Garbage Collection is called Major GC and usually takes a longer time.

Memory Management in Java - Java Garbage Collection

Java Garbage Collection is the process to identify and remove the unused objects from the memory and free space to be allocated to objects created in future processing. One of the basic ways of garbage collection involves three steps:

- **Marking:** This is the first step where garbage collector identifies which objects are in use and which ones are not in use.
- **Normal Deletion:** Garbage Collector removes the unused objects and reclaim the free space to be allocated to other objects.
- **Deletion with Compacting:** For better performance, after deleting unused objects, all the survived objects can be moved to be together. This will increase the performance of allocation of memory to newer objects.