

Java:

Java is a high-level, class-based, object-oriented programming language that is designed to have as few implementation dependencies as possible.

**How it works**

**Github: hieunguyeu810**

|  |  |
| --- | --- |
| **Area** | **Contain** |
| **Heap** | Objects |
| **Stack** | Methods  Local variable  Reference Available |
| **Code** | Byte Code |
| **Static** | Static Methods and Data |

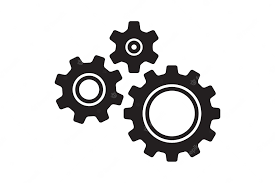
|  |  |  |
| --- | --- | --- |
| **Single GC** | Single thread environment--> Simple CMD line program | -XX: +UseSerialGC |
| **Parallel GC** | Multiple thread --> Ifapplication can handle such pauses and try to optimize overhead | -XX:+UseParallelGC |
| **Concurrent Mark Sweep (CMS) GC** | There are multiple threads that are used to mark instances for eviction and then sweep the marked instances from the heap memory --> CMS collector uses more CPU than the parallel garbage collector. Because it gives better performance to the application | -XX:+USeParNewGC |
| **G1 GC** | Used for large heap memory areas.The newest garbage collector is intended as a replacement for CMS. G1 also compacts the free heap space on the go just after reclaiming the memory | -XX:+UseG1GC |

|  |  |
| --- | --- |
| **Integer value** | id=10 |
| **String reference** | name |
| **Person reference** | this |

|  |  |
| --- | --- |
| **Integer value** | id=10 |
| **String reference** | name |
| **Person reference** | Perion |

|  |
| --- |
| **Person(int, String)** |
| **addPerson(int, String)** |
| **main(String[])** |

|  |  |
| --- | --- |
| **Integer value** | id=10 |
| **String reference** | name |
| **Person reference** | Person |



OS memory

**Host Operarting System**

**JVM**

Heap Memory

Initial Heap size

JVM

memory

JVM Max Heap Size

-Xmx

-Xms

Non-Heap Memory

**Executor Service**

**Task Queue**

Task

Task

Task

**Thread Pool**

Thread

Thread

Thread

**JVM Language Classes**

**Class loader**

**JVM Memory**

**Method Area**

**Heap**

**JMV Language Stacks**

**PC Registers**

**Native Method Stack**

**Execution Engine**

**Native Method Interface**

**Native Method Libraries**

**Heap Space**

“Tom”

10 | “name” reference

**Person Object**

**String Pool**

**Call Stack**

**Stack Memory**

**Task Submitters**

Submitter

Submitter

Submitter

**Thread Pool**

Blocked

Runnable

Running

New

Death

run()

sleep()

stop()

stop()

stop()

End of execution

**Thread**

JVM

Bytecode

Machine code

JTI Compiler

JAVAC

**Heap and Stack**

start()

Permanent Generation

New Object

**Garbage Collector**

Survivor Space 1

No longer in use

Major GC cycle

Tenured Space

GC cycle

GC cycle

**Old Generation**

Survivor Space 0

Eden Space

**Young Generation**

**Heap Memory**

Young Generation

(-XX: MaxNewSize)

Old Generation

Reversed

Tenured Pool

Eden Pool

Survivor Pool

Reversed

**JRE**

**Java class libraries**

**Class loader and Bytecode verification**

**JVM**

**Java Interpreter**

**Garbage collector**

**Threading**

**JDK: Java development kit**

**Development Tools**

**javac**

**java**

**javap**

**javaw**

**javadoc**

**...**

**JRE:**

**Java Runtime Environment**

**JVM:**

**Java Virtual Machine**

**Java library**

**Other files**

Java source code

Interpret

Compile

Hello.class

Hello.java

public class Hello {

...

}

**Non-Heap Memory**

Perm Gen

Code cache

Reversed

-XX:PermSize

-XX:MaxPermSize