

# Java Virtual Machine

<http://inbravo.github.io/html/jvm.html>

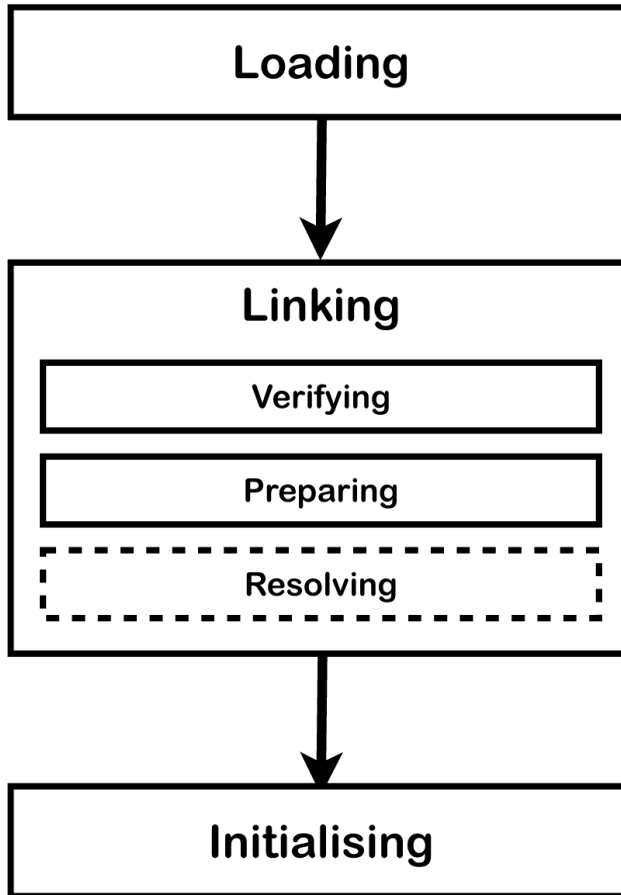
# Class file structure

Magic	Version
Constant Pool	
Access Flags	
this Class	
super Class	
Interfaces	
Fields	
Methods	
Attributes	

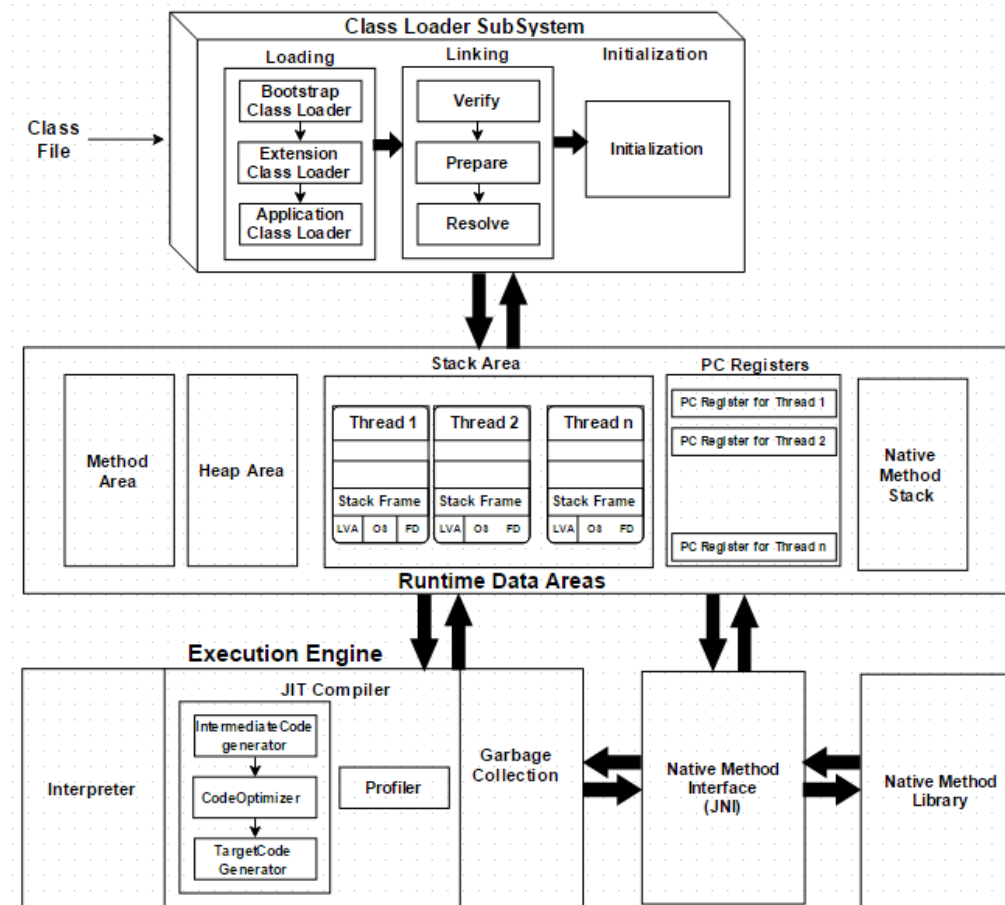
Lets run a java class with `-verbose`

Java `-verbose` Anagram

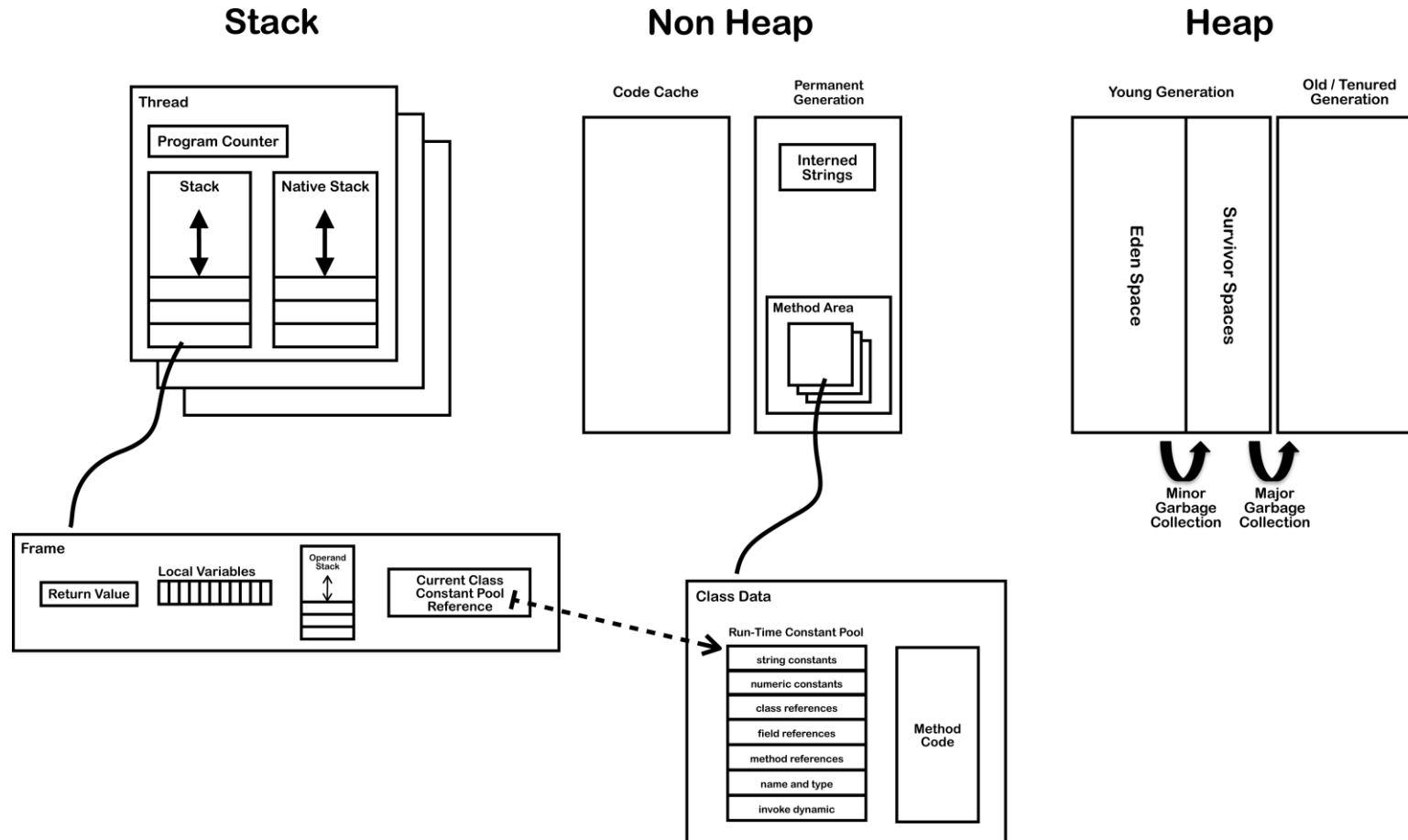
# Class loader sub system



# JVM architecture

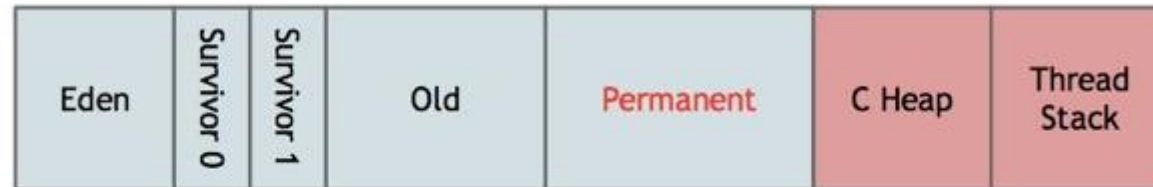


# Runtime data areas



# Heap

## JDK 7



## JDK 8



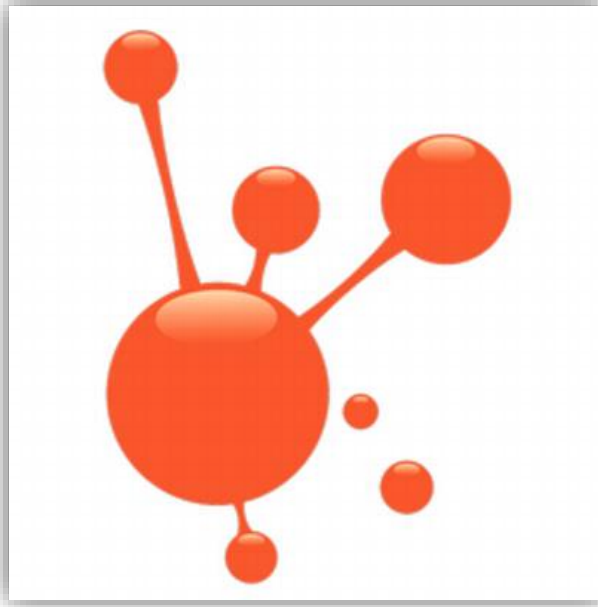
# JDK byte code disassembler

- JDK/bin/**javap** -c Anagram
- Javap looks boring 😞
- lets use another tool **byte code viewer** to do make it interesting 😊



# Jvisualvm

- Available since JDK 7
- Java process id based analysis and thread dump analysis



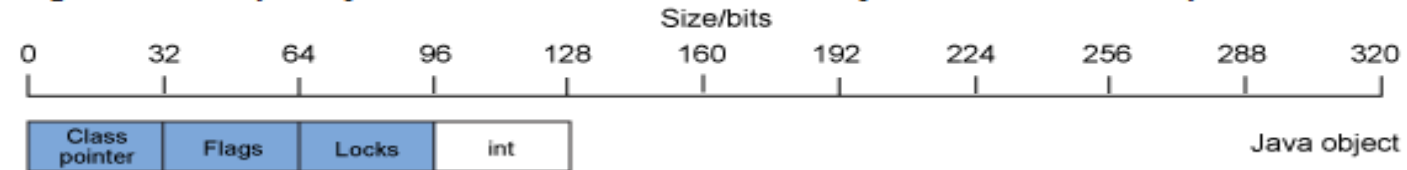
# Primitive variable sizing

- Int: 32-bit or 4 byte
- long: 64-bit or 8 byte
- float: 32-bit or 4 byte
- double: 64-bit or 8 byte
- boolean, char, byte, short: 32 bit or 4 byte

# Object sizing

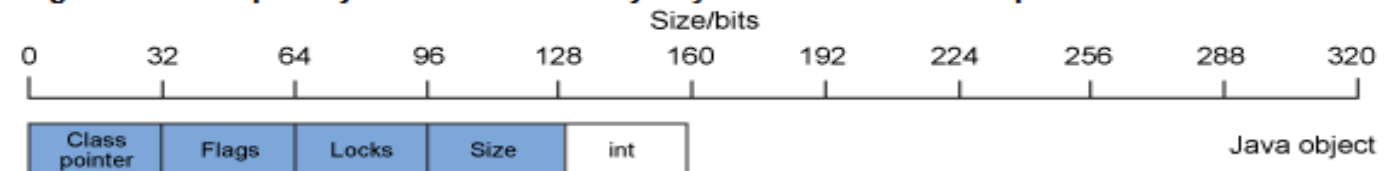
- Size of java.lang.Integer

Figure 2. Example layout of a **java.lang.Integer** object for a 32-bit Java process



## Size of int array

Figure 3. Example layout of an **int** array object for a 32-bit Java process



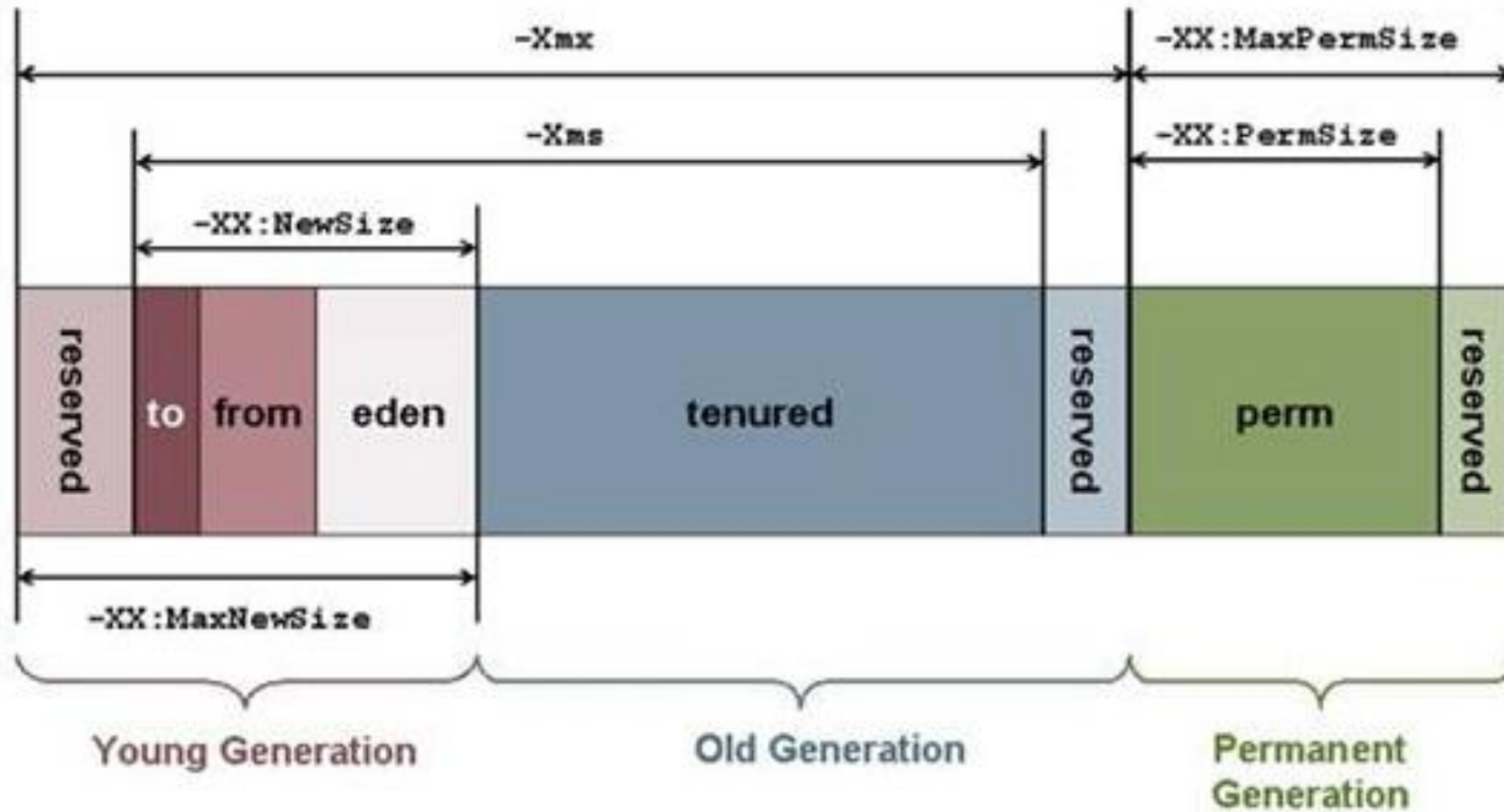
# JVM Source code

<http://hg.openjdk.java.net/jdk8/jdk8/hotspot>

# JVM options

- Option types: Standard(-), Non-standard(-**X**), Developer (-**XX**)
- Option: -**XX:+PrintFlagsFinal** lists all flags available
- Option types:
  - [Behavioural options](#)
  - [Garbage Collection options](#)
  - [Performance tuning options](#)
  - [Debugging options](#)

# JVM options



# JVM options

Option	Description
-Xms	Initial heap memory size
-Xmx	Maximum heap memory size
-Xmn	Size of Young Generation
-XX:PermSize	Initial Permanent Generation size
-XX:MaxPermSize	Maximum Permanent Generation size

# JVM options

Option	Description
-XX:+UseSerialGC	Serial Garbage Collector
-XX:+UseParallelGC	Parallel Garbage Collector
-XX:+UseConcMarkSweepGC	CMS Garbage Collector
-XX:ParallelCMSThreads=	CMS Collector – number of threads to use
-XX:+UseG1GC	G1 Garbage Collector



# References

- Topic details : <http://inbravo.github.io/html/jvm.html>
- Examples:
  - Anagram.java : <https://github.com/inbravo/java-src/blob/master/src/com/inbravo/string/Anagram.java>
  - PrimitiveVarsSizeTest.java: <https://github.com/inbravo/java-src/blob/master/src/com/inbravo/memory/PrimitiveVarsSizeTest.java>
- Byte code instructions: [https://en.wikipedia.org/wiki/Java\\_bytecode\\_instruction\\_listings](https://en.wikipedia.org/wiki/Java_bytecode_instruction_listings)
- Byte code viewer tool: <http://bytecodeviewer.com>
- Hex editor: <https://mh-nexus.de/en/hxd>
- Jvisualvm: <https://visualvm.java.net/download.html>

# Whacky questions 😊

- Why should I learn about byte code?
- Why should I know Jvisualvm?
- What the hell I am going to do with Hex Editor?
- What is the benefit of understanding object sizing?
- Why should I read java source code?
- Why understanding of JVM is so important?
- Why should I bother about JVM options?
- What is next?