

java.lang.Object

- class java.lang.Object
- memory for java objects
- object header

class Object

class java.lang.Object is a superclass of all classes in Java

So, each object in Java is Object.

Methods of class Object

- `public Class<?> getClass()`
- `public int hashCode()`
- `public boolean equals(Object obj)`
- `protected Object clone()`
- `public String toString()`
- `public void notify()`
- `public void notifyAll()`
- `public void wait(long timeout)`
- `protected void finalize()`

Size of an object

Where is no `size_of()` in Java

Real size of the object depends on:

- The version of the jdk
- Vendor of the jdk (Oracle JDK, Open JDK...)
- Bit capacity (32 bit or 64 bit)
- JVM options

How to calculate the size of an object

Each object is a tree with primitive types as leafs and references as edges.

Total size of the object consists of:

- Object header (8 byte)
- Size of all primitive type class members
- Size of all references (32 or 64 bits)
- + Granularity – 0...7 bytes. Result must be a multiple of 8.

Header of an object

Oracle JVM (HotSpot), 32 bit architectures.

The object header consists of:

- mark word (4 bytes) – hash, age, lock;
- klass pointer (4 bytes) – pointer to the class of the object.

8 bytes for an Object

12 bytes for arrays. 4 bytes (int) for the length