Blogs

Core java

PageNo: ▼ Go << >>

--Tag-- ▼ Search

Operator Overloading in JAVA

Q: How do you do Operator Overloading in JAVA?
A: Operator overloading is done in c++ but JAVA does not support it.

Class Loader

Q: What is class loader?
A: JVM uses class loaders to load classes in the memory.

Q: How many types of class loaders are there?
A: There three type of class loaders

1. Bootstrap class loader: Loads the core Java libraries from *<JAVA_HOME>/lib* directory.
2. Extensions class loader: loads the code in the extensions directories from *<JAVA_HOME>/lib/ ext directory.*
3. System class loader: loads code found in JAVA CLASSPATH

Q: Can you define or customize ClassLoader?
A: Yes, Classloader can be customized by extending java.lang.ClassLoader.

Q: How to unload a class?
A: Classes will be unloaded by Garbage Collector. Garbage Collector can be explicitly called by System.gc();

Make an Immutable Class

Q: How you can make a Java Class Immutable
Ans: There are 4 steps to make a Class immutable
(1) Define **private final** attributes to hold the Class state
(2) Define **parameterized Constructor** to initialize final attributes.
(3) Define **getter** methods the fetch the attributes.
(4) Define Class **final** so that Child cannot be created and Child cannot change the behavior of Class.

Example.

```
// The immutable class which is made final
final class MyImmutableClass
{
// instance var are made private & final to restrict the access
private final int count;
private final double value;

// Parameterized  Constructor where we can provide the constant value
public MyImmutableClass(int paramCount,double paramValue)
{
   count = paramCount;
   value = paramValue;
}
// provide only methods which return the instance var
// & not change the values
public int getCount()
{
   return count;
}
public double getValue()
{
  return value;
}
}
// class TestImmutable
public class TestImmutable
{
  public static void main(String[] args)
  {
   MyImmutableClass obj1 = new MyImmutableClass(3,5);
   System.out.println(obj1.getCount());
   System.out.println(obj1.getValue());
   // there is no way to change the values of count & value-
  // no method to call besides getXX, no subclassing, no public access to var -> Immutable
}
}
```

String Vs StringBuffer

Q: What are differences between String and StringBuffer classes?
A: String is immutable whereas StringBuffer is mutable.

Sting comparison : == operator Vs equals() in String

Q: What is the difference between == operator and equals() (http://download.oracle.com/javase/1.5.0/docs/api/java/lang/Object.html#equals%28java.lang.Object%29) **method in String comparison?**
A: Differences

- The equals( ) method compares the characters inside a String object The == operator compares two object references to see whether they refer to the same memory address.
- == operator Compares references, not values. Where as Equal Method Compares values for equality. Because equal() method is defined in the Object class, from which all other classes are derived, it's automatically defined for every class.
- Equal is a method of object Class where as == is an Operator.