Random Testing
Test Case Generator

Class under test:

class A {
    public A(int x){...}
    public int m1(){...}
    public int m2(B b){...}
    private int m3(){...}
}

Thursday, January 17, 13
Generated Test

class ATest {
    @Test
    public generateTest() {
        A v1 = new A(5236);
        v1.m1();
        B v2 = new B();
        v1.m2(v2);
        A v3 = new A(-7829);
        v3.m2(v2);
        ...
    }
}
Test Case Generator

- Use **reflection** to get constructors and methods, and their parameters.

- Use constructor objects to create an instance.

- Use method objects to invoke methods.

- If the call succeeded produce equivalent Java code.
Reflection -- some code snippets

Loading a class

Class.forName(fullyQualifiedName)

Create an instance

Class[] types = new Class[]{String.class,String.class};
Constructor cons=TwoString.class.getConstructor(types);
Object[] args = new Object[] {"a","b"};
TwoString ts = (TwoString)cons.newInstance(args);

Invoke a method

int value = 1;
mname = "set";
types = new Class[] { int.class };
method = obj.getClass().getMethod(mname, types);
method.invoke(obj, new Object[] {new Integer(value)});
Object Pool

- Returns objects for given class/interface
- Allows reuse of objects
Class Loading in Java

- Class loading: Loading the binary representation into the JVM.
- Each class is uniquely defined by its ClassLoader and its name,
- Classes are loaded at the first active use or explicitly with a call to loadClass()
Bootstrap Class Loader

• Loads bootstrap classes, e.g. classes from rt.jar / classes.jar (containing java.lang classes) and from the given classpath.

• Order on classpath matters
When to stop

• Max number of invocations has been reached

• Runtime exception occurs and it may be considered as a bug.
  
  • Exceptions that are subclasses of **Error** are considered as wrong calls
  
  • **Checked** exceptions are considered as wrong calls
  
  • **Unchecked** exceptions are considered as bug

  • e.g. ArrayIndexOutOfBoundsException, ClassCastException, ArithmeticException, NullPointerException...