

# Agenda

---

- **Eclipse IDE**
- **JUnit**
- **ServletUnit, HttpUnit, StrutsUnit, Cactus**

# Eclipse IDE

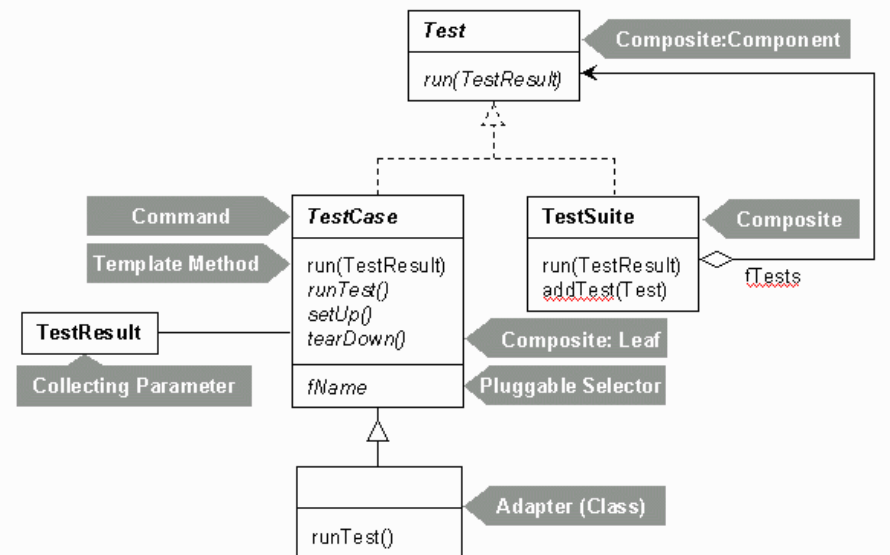
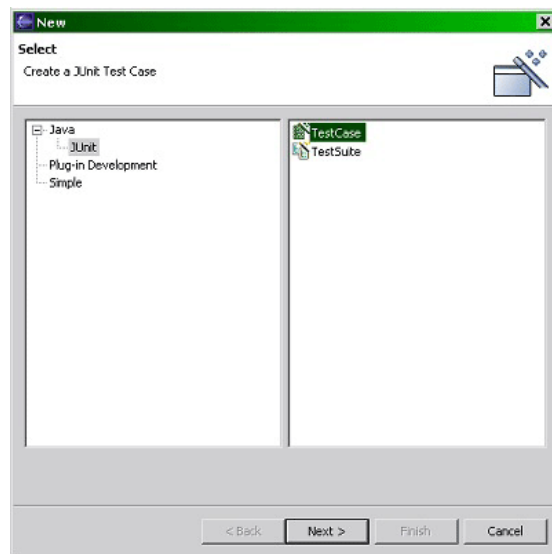
---

- <http://www.eclipse.org>
- Tutorials at  
[www.3plus4software.de/eclipse](http://www.3plus4software.de/eclipse)
- Online courses
  - Java Programming with Eclipse
  - Eclipse Plug-in Development<http://www.eclipse.org/ecesis/>



# JUnit

- [www.junit.org](http://www.junit.org)
- Using JUnit with Eclipse IDE  
<http://www.onjava.com/pub/a/onjava/2004/02/04/juie.html>
- New -> Other, Java, JUnit



# Example testing java.util.Vector (1 of 2)

---

```
00001: package junit.samples;
00002:
00003: import junit.framework.*;
00004: import java.util.Vector;
00005: import junit.extensions.*;
00006:
00007: /**
00008:  * A sample test case, testing <code>java.util.Vector</code>.
00009:  *
00010:  */
00011: public class VectorTest extends TestCase {
00012:     protected Vector fEmpty;
00013:     protected Vector fFull;
00014:
00015:     protected void setUp() {
00016:         fEmpty = new Vector();
00017:         fFull = new Vector();
00018:         fFull.addElement(new Integer(1));
00019:         fFull.addElement(new Integer(2));
00020:         fFull.addElement(new Integer(3));
00021:     }
00022:
```

# Example testing java.util.Vector (2 of 2)

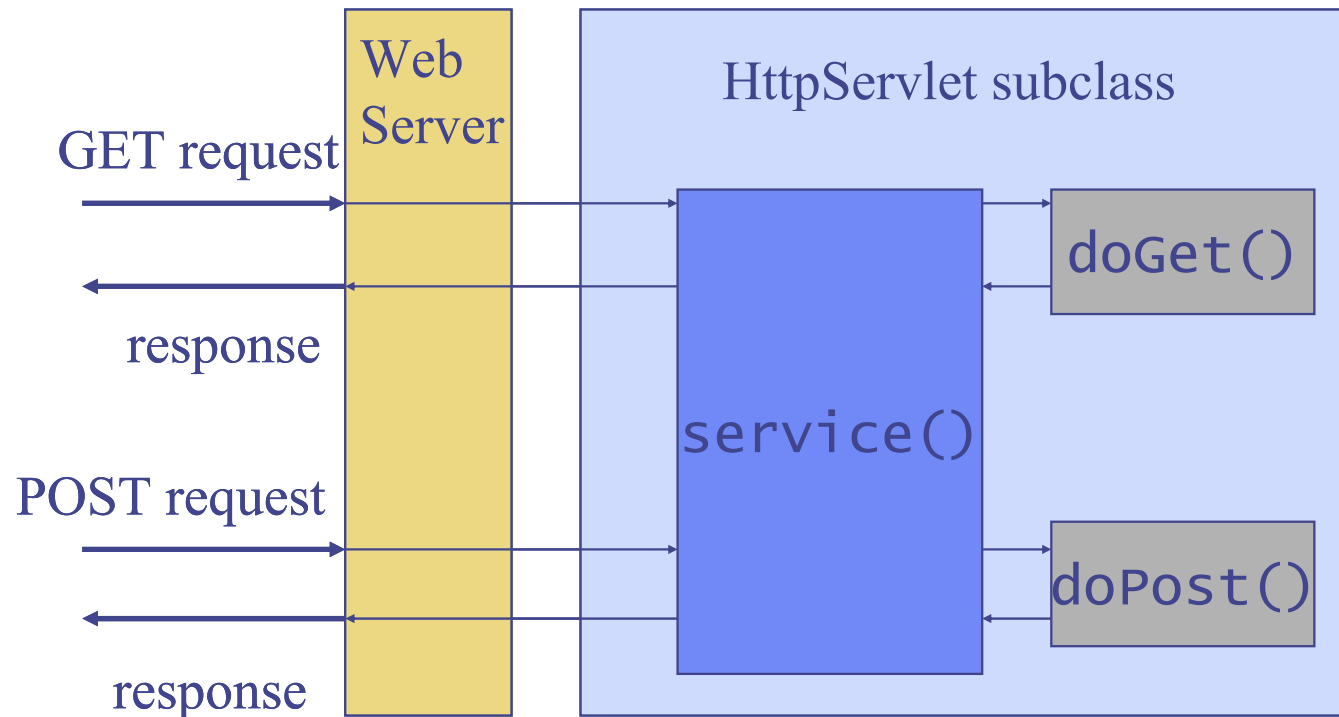
---

```
00023:         public void testCapacity() {
00024:             int size= fFull.size();
00025:             for (int i= 0; i < 100; i++)
00026:                 fFull.addElement(new Integer(i));
00027:             assertTrue(fFull.size() == 100+size);
00028:         }
00029:
00030:         public void testContains() {
00031:             assertTrue(fFull.contains(new Integer(1)));
00032:             assertTrue(!fEmpty.contains(new Integer(1)));
00033:         }
00034:         public void testElementAt() {
00035:             Integer i= (Integer)fFull.elementAt(0);
00036:             assertTrue(i.intValue() == 1);
00037:
00038:             try {
00039:                 Integer j= (Integer)fFull.elementAt(fFull.size());
00040:             } catch (ArrayIndexOutOfBoundsException e) {
00041:                 return;
00042:             }
00043:             fail("Should raise an ArrayIndexOutOfBoundsException");
00044:         }
00045:
00046:     }
```

# Test-driven scenarios

---

- Walkthrough: SimpleFractionTest
- Another scenario: web-based systems



From <http://sean.cpsc.ucalgary.ca/courses/SENG/513/F2002/slides/Servlets.ppt>

# Servlet counter example

---

```
public class Counter extends HttpServlet
{
    int count = 0;

    public void doGet(HttpServletRequest req, HttpServletResponse res)
        throws ServletException, IOException
    {
        res.setContentType("text/html");
        PrintWriter out = res.getWriter();
        synchronized (this) {
            count++;
            out.println("This servlet has been accessed " + count +
                " times since loading");
        }
    }
}
```

---

From <http://sern.cpsc.ucalgary.ca/courses/SENG/513/F2002/slides/Servlets.ppt>

# Think time (d.h. Übungsfragen)

---

- **Q1: Wie kann man das JUnit-Konzept erweitern, um eine webbasierte Anwendung zu testen?**
- **Q2: Wie sind die Ergebnisse einer Webanwendung zu testen? Welche Granularität?**
- **Q3: Webanwendungen haben normalerweise Zustand (eingeloggt, Warenkorb, To-Do-Liste) der in einer RDBMS gespeichert wird. Wie wird das Testen beeinflusst?**



# HttpUnit

---

- <http://www.httpunit.org>
- **Empfehlenswert:**
  - Cookbook
  - Tutorial
- **Weitere Beispiele:**
  - Build a Java Web Application Using HttpUnit and the Test-driven Methodology  
<http://www.devx.com/Java/Article/17908>
  - HttpUnit: A Civilized Way to Test Web Applications in WebSphere Studio  
[http://www-106.ibm.com/developerworks/websphere/library/techarticles/0303\\_bhogal/bhogal.html](http://www-106.ibm.com/developerworks/websphere/library/techarticles/0303_bhogal/bhogal.html)