

Tutorial 7

Unit Test and Web service deployment

junit, Axis

Last lecture...

On Software Reuse

- The concepts of software reuse:
to use the knowledge more than once
- Classical software reuse techniques
- Component-based software reuse
 - How web services help with component-based software reuse
- Quality-based software reuse
 - How aspect-orientation helps with quality-base reuse
- **Don't reinvent the wheel**

Today...

This tutorial will cover skills you can apply to your project immediately:

- Unit test
- Deployment of Web services

Note. We shall do the two tutorials on these subjects, but we have to save some time for you to express yourselves during the next lecture time. Thus we compress the two tutorials into one.

1. What is Unit Test?

- Unit Test: ...
- Integration Test: ...
- System Test: ...
- Acceptance Test: ...

The earlier a bug is found, the better!

Test can be done once a unit is ready!

Replace other units with stubs!

What can be tested in units?

- A functional requirement: Given input that satisfies the precondition, whether the output satisfies the post-condition
- A unit can be a member function, a class, a package, a component or a subsystem ...
- Automation is the key! Replace user interaction with the scripts, if possible; replace some with stubs ...
- A unit tested can still have bugs, but most trivial bugs should have been found

What can not be tested?

- Generally, test can not replace the verification or code review
- Specifically for unit test, interactions between this unit and other units after integration, system and user acceptance are not possible when the system is not ready yet

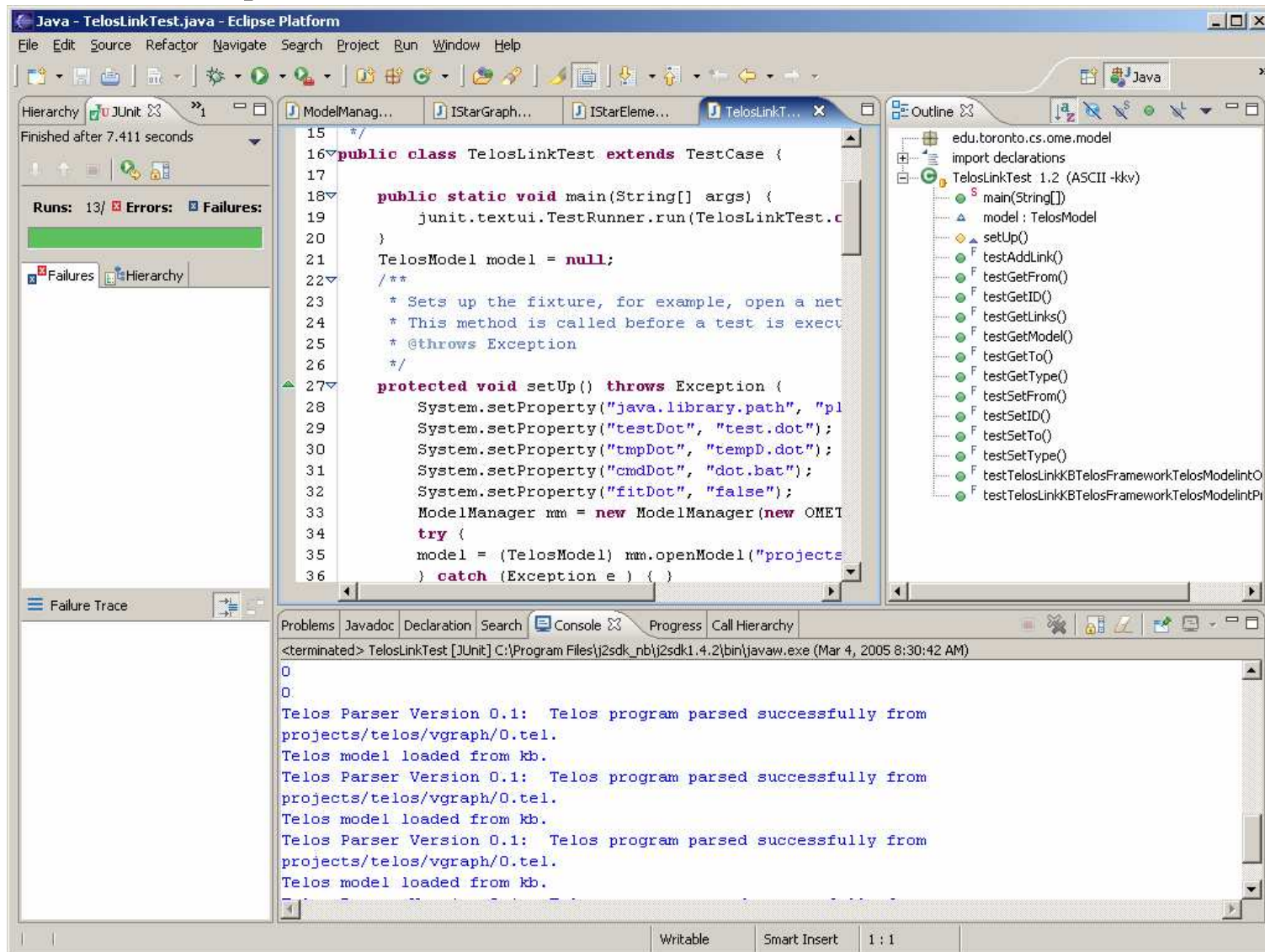
JUnit

- JUnit is an open source Java *testing framework* used to write and run repeatable tests.
- JUnit features include:
 - Assertions for testing expected results
 - Test fixtures for sharing common test data
 - Test suites for easily organizing and running tests
 - Graphical and textual test runners
- JUnit was originally written by Erich Gamma and Kent Beck (the authors of the Gang of Four (design patterns) book).

Test-driven development

- Extreme Programming
 - Also called “Paired programming”
 - How test helps? Avoid misunderstandings of the requirements of the swapped units
- Refactoring
 - How test helps? Make sure each refactoring step preserves the semantics and achieves quality improvements

OpenOME and JUnit



2. Web Service Deployment in the course

- We have 11 teams, codenames are:
*mindz, photons, team2, xteam, lumiere
solutions, muggq, overnight enterprise, team7,
websilon, team9, canadiantired*
- The URL of the tomcat server is
<http://128.100.36.17:8080>
- The deployed web service should be placed
under c:\program files\apache software
foundation\tomcat 5.5\webapps\axis\WEB-
INF\classes\`<codename>`

2.1 Develop Web service in AXIS

`deploy.wsdd`, `undeploy.wsdd` can be generated from WSDL:

- `java -cp $AXISCLASSPATH org.apache.axis.wsdl.WSDL2Java -s -d Session -Nurn:AddressFetcher2=samples.addr samples/addr/AddressBook.wsdl`

Start a simple Axis server

- `java -cp .:$AXISCLASSPATH org.apache.axis.transport.http.SimpleAxisServer -p 9012 &`

Deploy the web service

- `java -cp $AXISCLASSPATH org.apache.axis.client.AdminClient -p 9012 samples/addr/deploy.wsdd`

Call the web service from the client program

- `java -cp .:$AXISCLASSPATH samples.addr.Main -p 9012 $*`

Feedback from the client

Using proxy without session maintenance.

(queries without session should say: "ADDRESS NOT FOUND!")

>> Storing address for 'Purdue Boilermaker'

>> Querying address for 'Purdue Boilermaker'

>> Response is:

[ADDRESS NOT FOUND!]

>> Querying address for 'Purdue Boilermaker' again

>> Response is:

[ADDRESS NOT FOUND!]

Using proxy with session maintenance.

>> Storing address for 'Purdue Boilermaker'

>> Querying address for 'Purdue Boilermaker'

>> Response is:

1 University Drive

West Lafayette, IN 47907

Phone: (765) 494-4900

>> Querying address for 'Purdue Boilermaker' again

>> Response is:

1 University Drive

West Lafayette, IN 47907

Phone: (765) 494-4900

2.2 Test Web Service using JUnit

Test Cases (e.g. `AddressBookTestCase.java`) can be generated by:

- `java -cp $AXISCLASSPATH org.apache.axis.wsdl.WSDL2Java -s -d Session -Nurn:AddressFetcher2=samples.addr --testCase samples/addr/AddressBook.wsdl`

Modify the generated `AddressBookTestCase.java` :

```
public void doTest () throws Exception {  
    String[] args = {"-p", "9012"};  
    Main.main(args);  
}
```

Run the following command:

- `java -cp .:$AXISCLASSPATH junit.textui.TestRunner -nolading samples.addr.AddressBookTestCase`

Feedback from the Unit Test

- **.- Testing address book sample.**
- Using proxy without session maintenance.
- (queries without session should say: "ADDRESS NOT FOUND!")
- >> Storing address for 'Purdue Boilermaker'
- >> Querying address for 'Purdue Boilermaker'
- >> Response is:
- [ADDRESS NOT FOUND!]
- >> Querying address for 'Purdue Boilermaker' again
- >> Response is:
- [ADDRESS NOT FOUND!]
- Using proxy with session maintenance.
- >> Storing address for 'Purdue Boilermaker'
- >> Querying address for 'Purdue Boilermaker'
- >> Response is:
- 1 University Drive
- West Lafayette, IN 47907
- Phone: (765) 494-4900
- >> Querying address for 'Purdue Boilermaker' again
- >> Response is:
- 1 University Drive
- West Lafayette, IN 47907
- Phone: (765) 494-4900
- **- Test complete.**

- **Time: 1.51**

- **OK (1 test)**

3. The Show time

- On March 9 (next Wednesday), Will be monitored and given an impression mark, 2% for presence, 8% for excellence in Phase B!
- Who will be the *OMEGraphEditor Idol* ?
- 11 teams in two rounds
 - *mindz, photons, team2, xteam, lumiere solutions*
 - *mugqq, overnight enterprise, team7, websilon, team9, canadiantired*
- We do Q&A session after the presentations of each round
- 5 minutes per team for presentation, 5 minutes for question/answers
- Which team will be the *star*? *A small prize will be endowed to the best teams.*

Last lecture...

On Software Reuse

- The concepts of software reuse:
to use the _____ more than once
- Classical software reuse techniques
- Component-based software reuse
 - How _____ help with component-based software reuse
- Quality-based software reuse
 - How _____ helps with quality-base reuse
- **Don't reinvent the wheel**

What can be tested in units?

- A _____ requirement: Given input that satisfies the precondition, whether the output satisfies the post-condition
- A unit can be a _____

- _____ is the key! Replace user interaction with the scripts, if possible; replace some with stubs ...
- A unit tested can still have bugs, but most trivial bugs should have been found

What can not be tested?

- Generally, test can not replace the _____ or _____
- Specifically for unit test, _____

_____ are not possible
when the system is not ready yet

Test-driven development

- Extreme Programming
 - Also called “Paired programming”
 - How test helps? _____
 - Refactoring
 - How test helps? _____
-