

Engenharia de Software

Testes

© 2001, 2004 **Engenharia de Software**
Departamento de Engenharia Informática
Instituto Superior Técnico

1

Motivação

- Garantir comportamento pretendido
 - Testes devem ser definidos *à priori*
 - Desenvolvimento e testes em paralelo
 - Permite medir o progresso do desenvolvimento
- Testes não garantem ausência de erros
- Construir uma bateria de testes permite garantir que erros anteriormente corrigidos não reaparecem

2

JUnit

- JUnit
 - Define a estrutura dos testes
 - Testes são implementados como subclasses de *TestCase*
 - A prática comum é replicar a hierarquia de pacotes

- Resultados de um teste
 - Teste executa-se como esperado
 - *Failure* - falha detectada pelo teste
 - *Error* - erro (excepção) de *runtime*

- Caso de teste.
 - Definido numa subclasse de *TestCase*
 - É um método sem parâmetros
 - Condições são verificadas com *assertXXXX()*

```
public void testStringConcat() {  
    String expected = "Hello World!";  
  
    // lógica do teste  
    String result = _helloString.concat(_worldString);  
  
    // verificação de resultados  
    assertEquals(expected, result);  
}
```

3

JUnit

- *Test Fixture*
 - Contexto de execução de testes
 - instâncias de objectos a utilizar num teste
 - Definir uma *Fixture* comum a vários casos de teste:
 - Criar uma subclasse de *TestCase*
 - Criar uma variável de instância para cada objecto
 - Redefinição de *setUp()* para inicializar as variáveis e obter recursos
 - Redefinição de *tearDown()* para libertar recursos

```
public class StringTest extends TestCase {  
    private String _helloString;  
    private String _worldString;  
  
    protected void setUp() {  
        _helloString = "Hello ";  
        _worldString = "World!";  
    }  
}
```

4

JUnit Ant Task

```
<junit haltonfailure="true" printsummary="yes"
  dir="${tests.data}" fork="yes">
  <formatter type="plain" usefile="false"/>
  <batchtest>
    <fileset dir="${tests.src.java}">
      <include name="**/*.java"/>
      <exclude name="**/ServiceTestCase.java"/>
    </fileset>
  </batchtest>
  <classpath>
    <pathelement path="${target.classes}"/>
    <pathelement path="${tests.target.classes}"/>
    (...)
  </classpath>
</junit>
```

5

JUnit Ant Task

- Parâmetros principais:
 - printsummary – impressão de relatório de testes
 - haltonfailure – pára a execução se um teste falhar
- Sub-elementos:
 - formatter – define o formato de apresentação dos resultados de testes
 - test – define uma classe de testes a executar
 - name – nome da classe de testes
 - batchtest – define um conjunto de classes de testes a executar
 - sub-elemento <fileset/> para definir ficheiros com testes

6

DbUnit

- Extensão ao JUnit para testar projectos que usam bases de dados
- Exportar/Importar a informação de/para bases de dados
- Permite utilizar *asserts* para verificar o conteúdo esperado da base de dados

7

DbUnit API

- **IDataSet**
 - Representa um conjunto de dados de várias tabelas
 - Utilizado nas operações de configuração das tabelas e verificação do seu conteúdo
- **FlatXmlDataSet**
 - Descrição das tabelas em XML

```
<!DOCTYPE dataset SYSTEM "my-dataset.dtd">
<dataset>
  <PESSOA CODIGO_INTERNO="1" UTILIZADOR="utilizador" PASSWORD="pass"/>
  <SECCAO CODIGO_INTERNO="1" NOME="Anúncios" ORDEM="0" CHAVE_SITIO="1"/>
  <SITIO CODIGO_INTERNO="1" NOME="EP" ANO_CURRICULAR="4" SEMESTRE="1" CURSO="LEI C"
    DEPARTAMENTO="DEI" CHAVE_SECCAO_INICIAL="1"/>
  <SITIO CODIGO_INTERNO="2" NOME="PO" ANO_CURRICULAR="2" SEMESTRE="1" CURSO="LEI C"
    DEPARTAMENTO="DEI"/>
  <OBJ_HL_SEQ/>
</dataset>
```

8

DbUnit API

- **IDatabaseConnection**
 - Interface DbUnit para aceder à base de dados
- **DataBaseOperation**
 - UPDATE
 - INSERT
 - DELETE
 - DELETE_ALL
 - CLEAN_INSERT
 - ...

9

Testes no GesDis

- Testes herdam de `ServiceTestCase`
- Cada classe só contém um caso de teste
- `localSetUp()` e `localTearDown()` podem ser redefinidos em cada classe de teste
- `testMe()` template de execução do teste
- `doTest()` implementação do caso de teste
- É possível redefinir *DataSet* de *setup* e de resultados esperados

10

gesdis.xml

```
<?xml version='1.0' encoding='ISO-8859-1'?>
<!DOCTYPE dataset SYSTEM "gesdis.dtd">
<dataset>
  <QJB_DLIST/>
  <QJB_DLIST_ENTRIES/>
  <QJB_DMAP/>
  <QJB_DMAP_ENTRIES/>
  <QJB_DSET/>
  <QJB_DSET_ENTRIES/>
  <QJB_HL_SEQ/>
  <QJB_LOCKENTRY/>
  <QJB_NRM/>
  <QJB_SEQ/>
  <ITEM CODIGO_INTERNO="1" NOME="1º Anúncio" ORDEM="1" INFORMACAO="Bem vindos a EPI"
    URGENTE="1" CHAVE_SECCAO="1"/>
  <ITEM CODIGO_INTERNO="2" NOME="2º Anúncio" ORDEM="2" INFORMACAO="+1 anúncio"
    URGENTE="0" CHAVE_SECCAO="1"/>
  <PESSOA CODIGO_INTERNO="1" UTILIZADOR="utilizador" PASSWORD="pass"/>
  <SECCAO CODIGO_INTERNO="1" NOME="Anúncios" ORDEM="0" CHAVE_SITIO="1"/>
  <SECCAO CODIGO_INTERNO="2" NOME="Material" ORDEM="1" CHAVE_SITIO="1"/>
  <SECCAO CODIGO_INTERNO="3" NOME="Teóricas" ORDEM="0" CHAVE_SECCAO_SUPERIOR="2"/>
  <SECCAO CODIGO_INTERNO="4" NOME="Práticas" ORDEM="1" CHAVE_SECCAO_SUPERIOR="2"/>
  <SECCAO CODIGO_INTERNO="5" NOME="Anúncios" ORDEM="0" CHAVE_SITIO="2"/>
  <SECCAO CODIGO_INTERNO="6" NOME="Planeamento" ORDEM="1" CHAVE_SITIO="2"/>
  <SITIO CODIGO_INTERNO="1" NOME="EP" ANO_CURRICULAR="4" SEMESTRE="1" CURSO="LEI C"
    DEPARTAMENTO="DEI" CHAVE_SECCAO_INICIAL="1"/>
  <SITIO CODIGO_INTERNO="2" NOME="PO" ANO_CURRICULAR="2" SEMESTRE="1" CURSO="LEI C"
    DEPARTAMENTO="DEI"/>
</dataset>
```

11

ServiceTestCase.java

```
protected final IDatabaseConnection getConnection() throws Exception {
    Class driverClass = Class.forName("com.mysql.jdbc.Driver");
    Connection jdbcConnection = DriverManager.getConnection("jdbc:mysql://localhost:3306/epXX", "epXX", "");
    return new DatabaseConnection(jdbcConnection);
}

public final void setUp() throws Exception {
    super.setUp();
    IDatabaseConnection connection = getConnection();
    IDataset dataSet = getDataSet();

    try {
        DatabaseOperation.CLEAN_INSERT.execute(connection, dataSet);
    } finally {
        connection.close();
    }
    localSetUp();
}

public final void tearDown() throws Exception {
    super.tearDown();
    localTearDown();
}
```

12

ServiceTestCase.java

```
protected abstract void doTest() throws Exception;

public final void testMe() throws Exception {
    try {
        doTest();

        String resultDataSetName = getResultDataSetName();

        IDataSet expectedDataSet = new FlatXmlDataSet(getClass().getResourceAsStream(resultDataSetName));
        IDataSet databaseDataSet = getConnection().createDataSet();

        String tableNames[] = expectedDataSet.getTableNames();
        for (int i=0; i < tableNames.length; i++) {
            ITable expectedTable = expectedDataSet.getTable(tableNames[i]);
            ITable actualTable = databaseDataSet.getTable(tableNames[i]);
            Assertion.assertEquals(expectedTable, new CompositeTable(expectedTable.getTableMetaData(),
                actualTable));
        }
    } catch (Exception ex) {
        fail("ServiceTest Case: " + ex.getMessage());
    }
}
```

13

CriarSitiolnexistente.java

```
public class CriarSitoInexistente extends ServiceTestCase {
    UserView _userView;

    protected void localSetUp() throws Exception {
        _userView = new UserView("utilizador");
    }

    protected void doTest() throws Exception {
        SitioView sitioViewExpected = new SitioView(new Sitio("ES", 3, 2, "LEIC", "DEI");
        SitioView sitioView;

        try {
            Object argsCriarSito[] = {"ES", new Integer(3), new Integer(2), "LEIC", "DEI"};
            sitioView = (SitioView) _gestorServicos.executar(_userView, "CriarSito", argsCriarSito);

            assertEquals("CriarSitiolnexistente: ", sitioViewExpected, sitioView);
        } catch (Exception e) {
            fail ("CriarSitiolnexistente: " + e);
        }
    }
}
```

14

CriarSitiolnexistente.xml

```
<?xml version='1.0' encoding='ISO-8859-1'?>
<!-- This file should not contain a DOCTYPE declaration -->
<dataset>
<!-- Sem referências às tabelas do OJB -->
<ITEM CODIGO_INTERNO="1" NOME="1º Anúncio" ORDEM="1" INFORMACAO="Bem vindos a EPI"
  URGENTE="1" CHAVE_SECCAO="1"/>
<ITEM CODIGO_INTERNO="2" NOME="2º Anúncio" ORDEM="2" INFORMACAO="+ 1 anúncio"
  URGENTE="0" CHAVE_SECCAO="1"/>

<PESSOA CODIGO_INTERNO="1" UTILIZADOR="utilizador" PASSWORD="pass"/>

<SECCAO CODIGO_INTERNO="1" NOME="Anúncios" ORDEM="0" CHAVE_SITIO="1"/>
<SECCAO CODIGO_INTERNO="2" NOME="Material" ORDEM="1" CHAVE_SITIO="1"/>
<SECCAO CODIGO_INTERNO="3" NOME="Teóricas" ORDEM="0" CHAVE_SECCAO_SUPERIOR="2"/>
<SECCAO CODIGO_INTERNO="4" NOME="Práticas" ORDEM="1" CHAVE_SECCAO_SUPERIOR="2"/>
<SECCAO CODIGO_INTERNO="5" NOME="Anúncios" ORDEM="0" CHAVE_SITIO="2"/>
<SECCAO CODIGO_INTERNO="6" NOME="Planeamento" ORDEM="1" CHAVE_SITIO="2"/>

<SITIO CODIGO_INTERNO="1" NOME="EP" ANO_CURRICULAR="4" SEMESTRE="1" CURSO="LEI C"
  DEPARTAMENTO="DEI" CHAVE_SECCAO_INICIAL="1"/>
<SITIO CODIGO_INTERNO="2" NOME="PO" ANO_CURRICULAR="2" SEMESTRE="1" CURSO="LEI C"
  DEPARTAMENTO="DEI"/>
<SITIO CODIGO_INTERNO="3" NOME="ES" ANO_CURRICULAR="3" SEMESTRE="2"
  CURSO="LEI C" DEPARTAMENTO="DEI"/>
</dataset>
```

15

CriarSitoExistente.java

```
public class CriarSitoExistente extends ServiceTestCase {
    UserView _userView;

    protected void localSetUp() throws Exception {
        _userView = new UserView("utilizador");
    }

    protected String getResultDataSetName() {
        return "/gesdis-result.xml";
    }

    protected void doTest() throws Exception {
        SitioView sitioView = null;

        try {
            Object argsCriarSito[] = {"EP", new Integer(4), new Integer(1), "LEI C", "DEI"};
            sitioView = (SitioView) _gestorServicos.executar(_userView, "CriarSito", argsCriarSito);

            fail("CriarSitoExistente: Service should NOT have executed OK.");
        } catch (NotExecutedException e) {
            // empty
        } catch (Exception e) {
            fail ("CriarSitoExistente: " + e);
        }
    }
}
```

16

Exercício

- Escrever testes para o serviço AlterarSitio
 - AlterarSitioInexistente
 - AlterarSitioExistente
 - AlterarSitioIgual
 - AlterarSitioSemUtilizador
- Template de teste
 - Redefinir o `localSetUp()`
 - Definir o `doTest()`
 - Invocar o serviço
 - Verificar estado esperado após execução
 - (Se necessário) redefinir o `getResultDataSetName()`
- Não Esquecer!
 - Definir os *DataSets* em XML com o resultado esperado na Base de Dados

17

Dicas para o Eclipse

- Dentro de Window -> Preferences -> Ant -> Runtime:
 - Em Properties definir a *global property* `ant.home` (não é `ANT_HOME`!)
 - Em classpath adicionar `junit.jar` e `tools.jar`
 - (o primeiro vem com o projecto. O segundo está em `%JAVA_HOME%\lib`)
- Não colocar o projecto num directório que contenha espaços no caminho (ex. Desktop)

18