Programmazione Orientata agli Oggetti in Linguaggio Java

Programmazione Grafica: Conclusioni

versione 1.0

G. Mecca – Università della Basilicata – mecca@unibas.it

Sommario

- Test Funzionali
  - ⇨JFCUnit
  - ⇨Registrazione dei Test

Test Funzionali

- Test sull'interfaccia grafica
  - ⇨possono essere test di unità (es: test del singolo pannello o frame)
  - ⇨ma sono essenzialmente test funzionali
  - ⇨ovvero test dell'applicazione nel suo complesso, dal punto di vista dell'utente
  - ⇨molto importanti perchè esercitano tutti gli strati assieme (vista, controllo e modello)
  - ⇨vanno sviluppati assieme ai test di unità

Test Funzionali

- Il problema dei test funzionali
  - ⇨scrivere test sull'interfaccia grafica è complesso per varie ragioni
  - ⇨I ragione: temporizzazione; i metodi di test devono "attendere" che gli eventi siano gestiti sull'interfaccia prima di effettuare asserzioni
  - ⇨II ragione: manipolare i componenti Swing e "simulare" gesti dell'utente attraverso il codice dei test non è semplice

JFCUnit

- JFCUnit
  - una estensione di JUnit per le Java Foundation Classes
  - package junit.extensions.jfcunit
  - progetto open source disponibile sul sito jfcunit.sourceforge.net
  - appositamente sviluppato per verificare interfacce create con Java Swing

JFCUnit

- Installazione di jfcunit
  - scaricare lo .zip dal sito
  - estrarre lo zip
  - da quel momento sono necessari per la scrittura dei test due jar
  - jfcunit.jar
  - jakarta-regexp.jar (versione 1.2 o success.), dispon. su http://jakarta.apache.org/regexp

JFCUnit

- Le funzionalità di JFCUnit
  - ⇨acquisire riferimenti ai frame principali dell'applicazione
  - ⇨"trovare" i componenti all'interno dei frame
  - ⇨simulare gesti dell'utente su questi componenti scatenando eventi
  - ⇨effettuare asserzione sullo stato della vista a seguito degli eventi

JFCUnit

- Idea alla base del funzionamento
  - ⇨ogni metodo di test tipicamente "riesegue" l'applicazione e simula un caso d'uso o una porzione di caso d'uso
  - ⇨l'applicazione viene visualizzata ed il test avviene in modo visivo
  - ⇨tipicamente l'avvio dell'applicazione avviene nel metodo setUp()

JFCUnit

- La classe di test
  - ⇨deve estendere JFCTestCase
  - ⇨il metodo setUp() deve chiamare super.setUp() come prima operazione
  - ⇨nel setUp() è necessario creare un oggetto di tipo TestHelper, che guida l'esecuzione dei test
  - ⇨due possibili soluzioni: JFCTestHelper, e RobotTestHelper

JFCUnit

- JFCTestHelper
  - ⇨simula gesti dell'utente inserendo eventi nella coda degli eventi
- RobotTestHelper
  - ⇨genera eventi chiamando metodi opportuni sui componenti (es: JButton.doClick())
  - ⇨le due classi sono sostanzialmente intercambiabili

JFCUnit

- Per cercare i componenti
  - ⇨ varie classi “finder”
  - ⇨ FrameFinder: cerca il riferimento al frame principale
  - ⇨ NamedComponentFinder: cerca componenti che hanno un nome (datogli con setName())
  - ⇨ molte altre classi, ma tipicamente vengono utilizzate queste due
- Attenzione
  - ⇨ se il componente non viene trovato, l’esecuzione del test resta “appesa” e poi termina senza segnalare errori

JFCUnit

>> morracineseswing - TestGuiPartita

- Per scatenare eventi
  - ⇨ numerosi metodi di TestHelper
  - ⇨ in particolare: enterClickAndLeave()
  - ⇨ richiede di creare un MouseEventData in cui viene specificato il riferimento al componente su cui fare click
  - ⇨ dopo aver scatenato un evento: è necessario chiamare flushAWT() per scaricare gli eventi generati nella coda degli eventi
- Un esempio
  - ⇨ morra cinese swing, TestGUIPartita

Registrazione dei Test

- Registrazione dei test
  - ⇨una funzionalità utilissima di JFCUnit
- In sintesi
  - ⇨avviando l'applicazione in modalità particolare, il framework è in grado di registrare tutti i gesti effettuati dall'utente
  - ⇨salvarli in un file .xml
  - ⇨e poi riprodurli in un metodo di test, eventualmente arricchiti di asserzioni

Registrazione dei Test

- Il processo di registrazione
  - ⇨si trata di un processo meccanico
- I passo
  - ⇨predisporre un file .xml per la registrazione

```
<?xml version="1.0" ?>
<suite name="indovinaswing">
   <test name="Test partita" robot="true" debug="true">
            <record encoding="UTF-8" file="test/dati/saved.xml"/>
   </test>
</suite>
```

salva il file registrato nella cartella test/dati della cartella di progetto

Registrazione dei Test

- Il passo
  - ⇨avviare la registrazione
  - ⇨un comando complesso

```
java -Djfcunit.xmlroot.record=true \
      -Djfcunit.xmlroot.classname=demo.SwingSet\
      -Djfcunit.xmlroot.testsuite=testcases.xml\
      -classpath jfcunit.jar;SwingSet.jar;jarkarta-regexp-1.2.jar;junit.jar\
      junit.swingui.TestRunner junit.extensions.jfcunit.tools.XMLRoot
```


Registrazione dei Test

- Una buona alternativa
  - ⇨il task test-record di swing-template-build.xml

```
<target name="test-record" depends="compile, -prepare-test">
    <java classname="junit.textui.TestRunner">
      <classpath refid="test.classpath" />
      <sysproperty key="jfcunit.xmlroot.record" value="true" />
      <sysproperty key="jfcunit.xmlroot.classname"
                   value="${main.class}" />
      <sysproperty key="jfcunit.xmlroot.testsuite"
                   value="test/dati/recordtest.xml" />
      <arg value="junit.extensions.jfcunit.tools.XMLRoot" />
    </java>
 </target>
```

Registrazione dei Test

- III passo
  - ⇨salvare il file ottenuto con un nome significativo; es: testPartita.xml
  - ⇨in questa fase è possibile aggiungere asserzioni al file xml
- IV passo
  - ⇨predisporre una classe per riprodurre il test
  - ⇨la classe ha una struttura standard
  - ⇨e non contiene veri e propri metodi di test

```
public class TestGuiXML extends XMLTestSuite {

  public TestGuiXML() throws Exception {
    super("saved.xml", openFile("/dati/testPartita.xml"));
    XMLRecorder.setReplay(true);
    Controllo.main(new String[] {});
  }

  private static InputStream openFile(final String fileName) {
    InputStream stream =
            TestGuiXML.class.getResourceAsStream(fileName);
    return stream;
  }

  public void testGuiXML() {
  }
}
```

Registrazione dei Test

- Nota
  - ⇨il sistema di registrazione non è perfetto
  - ⇨alcuni gesti “scontati” (es: uso del tabulatore o cancellazione del testo selezionato) a volte non vengono correttamente registrati
  - ⇨bisogna acquisire dimestichezza
  - ⇨è opportuno registrare test brevi piuttosto che test molto lunghi che rischiano di fallire per inaccuratezza della reg.

Riassumendo

- Test Funzionali
  - ⇨JFCUnit
  - ⇨Registrazione dei Test

Termini della Licenza

- This work is licensed under the Creative Commons Attribution-ShareAlike License. To view a copy of this license, visit http://creativecommons.org/licenses/by-sa/1.0/ or send a letter to Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA.

- Questo lavoro viene concesso in uso secondo i termini della licenza “Attribution-ShareAlike” di Creative Commons. Per ottenere una copia della licenza, è possibile visitare http://creativecommons.org/licenses/by-sa/1.0/ oppure inviare una lettera all’indirizzo Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA.