

13

*Test-driven
Programming
Strategies*

Self-review Questions

13.1 *What are the aims of a well-designed unit test?*

13.2 *What are the advantages and disadvantages of using real File objects when testing?*

13.3 *When does the design of application code have to be modified to facilitate testing?*

13.4 *What is a dummy object?*

A dummy object provides a fake implementation of a real object. A dummy object can be substituted for the real object when running a test, avoiding the need for the test to depend on creating and using a real object. This is particularly useful in situations where the real object is difficult to create, as it depends on creating other objects, and also where the real object makes use of resources such as files.

A dummy object is an instance of a dummy class that either implements an interface or subclasses the real class. A reference variable of the interface or superclass type can refer to the dummy object, allowing it to be injected into code being testing in place of an object of the real class. Typically any inherited behaviour is ignored and the dummy class overrides methods to check they are called or that certain parameter values have been passed.

A dummy class is typically written by a programmer as required, often as a nested class within a test class so that it forms part of the infrastructure of the test class. Commonly used dummy classes can be collected into a small class library and reused in different projects.

13.5 *What is a mock object?*

A mock object also provides a fake implementation but the mock object is generated automatically at runtime from an interface or class. In addition, the mock object has the ability to automatically check its usage in terms of method calls made and parameter values passed.

13.6 *What is 'injection'?*

Injection is used to insert a mock or dummy object into code being tested, in place of a real object. This can be done in several ways, the simplest of which is when the method being tested can be passed a mock object reference via a parameter normally used for a real object. If this cannot be done, the method being tested may need to be refactored to replace an expression with a method call for a method that can be overridden in a subclass created when testing. The overriding method can then create and inject a mock object.

13.7 Outline the advantages and disadvantages of mock compared to dummy objects.

13.8 Outline how a GUI object can be located for testing.

13.9 What features of a GUI can and cannot be unit tested?

Programming Exercises

13.1 Extend the `SwingAssert` class to include assert methods that check a component like a button is in the correct panel and a panel has the correct layout manager.

13.2 Extend the directory structure used when testing the `matchFiles` method to include another sub-directory and confirm that three files with the same name will be matched. Do this for both the live file test class and the dummy file test class.

13.3 Consider a test to confirm that a window is the correct size when it is displayed. Show how this test could be written and how needing to know the size affects the design of the GUI.

13.4 Use the test-driven approach to modify the searcher GUI to display a warning dialogue when the search button is clicked but no file name has been specified.

Challenges

13.1 Modify the `FileNameSearch` class to use regular expressions to match file names.

13.2 Use the test-driven approach to replace the `JTextArea` used to display the list of files found with a `JList`.

13.3 Extend `Searcher` to search the contents of files for a string specified using a regular expression.

13.4 Use test-driven development to write an application that searches websites for HTML files with a specific name.