

| | |
|-------------------------|---|
| 1. Record Nr. | UNINA9910456617703321 |
| Autore | Lavers Tim |
| Titolo | Swing extreme testing [[electronic resource]] : the extreme approach to complete Java application testing / / Tim Lavers, Lindsay Peters |
| Pubbl/distr/stampa | Birmingham, U.K., : Packt Publishing Ltd., 2008 |
| ISBN | 1-84719-483-4 |
| Edizione | [1st edition] |
| Descrizione fisica | 1 online resource (328 p.) |
| Collana | From technologies to solutions |
| Altri autori (Persone) | PetersLindsay |
| Disciplina | 005.1/4 |
| Soggetti | Computer software - Testing eXtreme programming Java (Computer program language) Electronic books. |
| Lingua di pubblicazione | Inglese |
| Formato | Materiale a stampa |
| Livello bibliografico | Monografia |
| Note generali | Includes index. |
| Nota di contenuto | Cover; Table of Contents; Preface; Chapter 1: What Needs Testing?; An Example; What Classes Do We Test?; Test First-Always!; What Classes Don't We Test?; What Methods Need Testing?; What Methods Don't We Test?; Invoking Hidden Methods and Constructors; Unit Test Coverage; Who Should Implement the Unit Tests?; What About Legacy Code?; Where Does Integration Testing Fit In?; Documentation of Unit Tests; Testing at the Application Level; Who Should Implement the Function Tests?; Automated Test Execution; A Hierarchy of Tests; What Language Should Our Tests Be In?; Is it Really Possible?; Summary Chapter 2: Basics of Unit TestingA Simple Example; The Default Implementation; Test Cases; Design by Contract and Non-Defensive Programming; Test Code Example; Bootstrapping Our Implementation; Load Testing; Summary; Chapter 3: Infrastructure for Testing; Where Should the Unit Tests Go?; Where Should the Function and Load Tests Go?; Management of Test Data; What Do We Require of a Test Data Management System?; Temporary Files; Summary; Chapter 4: Cyborg-a Better Robot; The Design of Cyborg; Using the Keyboard; Mousing Around; Checking the Screen; Summary Chapter 5: Managing and Testing User MessagesSome Problems with Resource Bundles; A Solution; The UserStrings Class; ResourcesTester; How ResourcesTester Works; Getting More from UserStrings; Summary; |

Chapter 6: Making Classes Testable with Interfaces; The LabWizard Comment Editor; The Wizard; A Test for Wizard; A Test for Step; Handlers in LabWizard; Summary; Chapter 7: Exercising UI Components in Tests; The LabWizard Login Screen; The Design of LoginScreen; UI Wrappers; The Correct Implementation of UILoginScreen; A Handler Implementation for Unit Testing; Setting Up our Tests Our First TestFurther Tests; Some Implicit Tests; Other User Interfaces; Summary; Chapter 8: Showing, Finding, and Reading Swing Components; Setting Up User Interface Components in a Thread-Safe Manner; Finding a Component; Testing Whether a Message is Showing; Searching for Components by Name; Reading the State of a Component; Case Study: Testing Whether an Action Can Be Cancelled; The Official Word on Swing Threading; Summary; Chapter 9: Case Study: Testing a 'Save as' Dialog; The Ikon Do It 'Save as' Dialog; Outline of the Unit Test; UI Helper Methods; Dialogs Getting the Text of a Text FieldFrame Disposal; Unit Test Infrastructure; The UISaveAsDialog Class; The ShowerThread Class; The init() Method; The cleanup() Method; The Unit Tests; The Constructor Test; The wasCancelled() Test; The name() Test; The show() Test; The Data Validation Test; The Usability Test; Summary; Chapter 10: More Techniques for Testing Swing Components; Testing with JColorChooser; Using JFileChooser; Checking that a JFileChooser has been Set Up Correctly; Testing the Appearance of a JComponent; Testing with Frames; Frame Location; Frame Size; Testing with Lists List Selection Methods

Sommario/riassunto

The Extreme approach to complete Java application testing
