Pragmatic Unit Testing in Java 8 with JUnit

Jeff Langr

with Andy Hunt
Dave Thomas

The Pragmatic Bookshelf
Dallas, Texas • Raleigh, North Carolina

Contents

Part II — Mastering Manic Mnemonics!

Part III — The Bigger Design Picture

Part IV — The Bigger Unit-Testing Picture

Foreword

Some time after Dave Thomas and I (Andy Hunt) wrote *The Pragmatic Programmer* and the first edition of *Programming Ruby*, we turned our attention to the most basic needs of modern software developers.

We came up with the idea of *The Pragmatic Starter Kit*, three books covering the most fundamental needs of a team: version control, unit testing, and automated build and test. These were the first three books we'd write and publish as the Pragmatic Bookshelf.

These topics are still fundamental and critical to any team's success, but a lot has changed over the last dozen years or so. Version-control technology has moved from centralized CVS and Subversion to a distributed model in Git. Automated build and related tools have become more scripted and more sophisticated, and testing has evolved from a hard-sell afterthought to a widely embraced approach via test-driven development.

Now Jeff Langr has taken on the task of updating and expanding our original unit-testing treatise for the modern world. The principles are the same, but the tools have gotten better, and I'd like to think the whole approach to software development has become more realistic, more professional, and—dare I say it?—more pragmatic. Jeff will show you the way.

Testing was always a poor name for this particular programming activity. The very name makes it sound like it's something separate from coding, separate from design, and separate from debugging.

It's not.

Your programming-language compiler/interpreter verifies that your source code is syntactically valid: that it makes at least some sort of sense according to the syntax of the language. But the compiler can't really tell what your code *does* and so can't help to determine if the code is correct or not.

Unit testing lets you specify what the code does and verifies that the code does it. Unit testing has become a marvelous intersection of design, coding, and debugging.

If you haven't gotten huge value from your testing yet, then this book will help you. Whether you're brand-new to the ideas here, or just trying to get the most benefit from unit testing, this book will help you.

Enjoy!

Andy Hunt
Publisher, The Pragmatic Bookshelf
Raleigh, NC