CHAPTER

9 Well-Behaved Objects

Main concepts discussed in this chapter:

- testing
- debugging
- unit testing
- test automation

Java constructs introduced in this chapter:

(No new Java constructs are introduced in this chapter.)

9.1 Introduction

If you have followed the previous chapters in this book, and if you have implemented the exercises we have suggested, then you have written a good number of classes by now. One observation that you will likely have made is that a class you write is rarely perfect after the first attempt to write its source code. Usually, it does not work correctly at first, and more work is needed to complete it.

The problems you are dealing with will shift over time. Beginners typically struggle with Java *syntax errors*. Syntax errors are errors in the structure of the source code itself. They are easy to spot, because the compiler will highlight them and display some sort of error message.

More experienced programmers who tackle more complicated problems usually have less difficulty with the language syntax. They are more concerned with *logical errors* instead.

A logical error is a problem where the program compiles and executes without an obvious error, but delivers the wrong result. Logical problems are much more severe and harder to find than syntax errors. In fact, it is sometimes not easy to detect that there is an error in the first place.

Writing syntactically correct programs is relatively easy to learn, and good tools (such as compilers) exist to detect syntax errors and point them out. Writing logically correct programs, on the other hand, is very difficult for any nontrivial problem, and proof that a