Foundational Java

David Parsons

Foundational Java

Key Elements and Practical Programming

Springer

David Parsons
Inst. of Information and Math. Sciences
Massey University
17 State Highway
Albany Campus
Albany, Auckland
New Zealand

ISBN 978-1-4471-2478-8 e-ISBN 978-1-4471-2479-5
DOI 10.1007/978-1-4471-2479-5
Springer London Dordrecht Heidelberg New York

British Library Cataloguing in Publication Data
A catalogue record for this book is available from the British Library

Library of Congress Control Number: 2012931443

© Springer-Verlag London Limited 2012
Apart from any fair dealing for the purposes of research or private study, or criticism or review, as permitted under the Copyright, Designs and Patents Act 1988, this publication may only be reproduced, stored or transmitted, in any form or by any means, with the prior permission in writing of the publishers, or in the case of reprographic reproduction in accordance with the terms of licenses issued by the Copyright Licensing Agency. Enquiries concerning reproduction outside those terms should be sent to the publishers.
The use of registered names, trademarks, etc., in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant laws and regulations and therefore free for general use.
The publisher makes no representation, express or implied, with regard to the accuracy of the information contained in this book and cannot accept any legal responsibility or liability for any errors or omissions that may be made.

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

To my mother, Angela (1931–2011), always there for me

Preface

In the Spring of 1996, I attended the Object Technology conference at Christ Church College, Oxford. The excited buzz around the "new" Java language at this conference got me curious, and a few months spent getting up to speed with the syntax convinced me that it was worth trying out as a teaching language. After overcoming some hurdles, including getting a teaching lab upgraded from Windows 3.1 to Windows 95 so we could actually run Java, I taught a class of student volunteers at what is now Southampton Solent University the basics of Java in 1997. From this experience I wrote the first edition of *Introductory Java* which was published by Letts Educational in 1998. In 1999, I left academia to spend some years in industry, training and consulting in Java. This gave me a whole new perspective on the language as a professional tool. The second edition of *Introductory Java* was published in 2003, but another update was long overdue.

This book was originally intended to be the 3rd Edition of *Introductory Java*. However, such were the changes that had taken place over the years, both in Java and in my own experiences of it, that this became a whole new book, with a new title and focus. It has grown out of 15 years of teaching Java both to students and to professionals across the world, and reflects a wealth of experience and insight. I have had a great journey with Java, and I am grateful for the opportunity to share my Java story with readers of this book.

Overview and Goals

Foundational Java: Key Elements and Practical Programming guides the reader through all the core features of Java, and some more recent innovations, in a way that builds skills and confidence through tried and trusted stages, supported by exercises that reinforce the key learning points. Using this book, the reader is introduced to all the most useful and commonly applied Java syntax and libraries, and is provided with many example programs that can provide the basis for more substantial applications.

Integral to the book is the use of the Eclipse Integrated Development Environment (IDE) and the JUnit testing framework. This ensures maximum productivity and code quality when learning Java. However, the fundamentals of using the Java compiler and run time are also explained to ensure that skills are not confined to one environment. In addition, coverage of the Ant tool ensures that the reader is equipped

to automatically build, test, and deploy their applications, including simple web applications, independent of an IDE.

Organization and Features

The book is organized into 20 chapters that cover various levels of Java (see "Suggested Uses" for further information.) Each chapter covers a discrete topic and includes scaffolded exercises that build skills in a step by step fashion. Earlier versions of the book used an "objects first" approach; however experience has shown me that it is better to cover the language fundamentals before addressing object oriented concepts, so this book follows an "objects later" philosophy.

The key features of this book are that it

- Meets the needs of both students and professionals
- Provides both introductory and intermediate coverage
- Is completely up to date, including Java 7
- Makes unit testing one of its key themes, introducing the JUnit 4 testing framework to emphasize the importance of unit testing in modern software development
- Uses the Eclipse IDE, the most popular open source Java IDE, but also explains how Java can be run from the command line
- Includes coverage of the Ant build tool
- Comes with code examples and exercises throughout
- Is accompanied by a full set of PowerPoint presentation slides that have been road tested with classes
- Builds on two previous editions and a set of classroom training materials that have been refined and developed as Java has continued to evolve
- Includes some important illustrations in color

Target Audiences

This is primarily an undergraduate textbook. It can be used for basic introductory courses or for intermediate classes. From that perspective, it has been structured as a teaching text that breaks into weekly topics that build upon one another. It is also a book suitable for professional software developers who need to pick up Java from previous experience in other tools or languages. The materials have been tried and tested in commercial training courses for professional software developers over the last 7 years. The choice of intermediate topics has been driven by customer requirements. All of these topics have been requested by clients in various courses.

Suggested Uses

The book has been structured in such a way that it breaks easily into weekly topics. There is a core set of chapters that can be used as an introductory course, in a single semester, and a further set of chapters that can be used for intermediate study, for

follow-on, longer or double-weighted courses. It contains exercises throughout, designed to reinforce learning about the topics covered in each part of the chapter. The final exercises at the end of each chapter draw together the key aspects that have been covered, which are also reiterated in chapter summaries.

Foundational Java can be used for courses of different lengths and levels by using it in three different ways. The first 12 chapters, listed below, cover the core knowledge of Java, and provide a solid basis for an introductory course on object oriented programming with Java. These fit easily into a 12 week semester, providing 11 weeks of teaching material (Chap. 1 is just an introduction) and opportunities for revision and reflection.

1. The Java Story
2. Compiling and Running Java Programs
3. Data Types, Arithmetic, and Arrays
4. Control Structures
5. Creating Objects
6. Creating Domain Classes
7. Objects Working Together: Association, Aggregation, and Composition
8. Inheritance, Polymorphism, and Interfaces
9. Exception Handling
10. Unit Testing with JUnit
11. Exploring the Java Libraries
12. The Collections Framework and Generics

The following four chapters, listed below, are more intermediate and provide more specialized coverage of Java; interaction with external connections to files, databases and build tools, as well as providing an introduction to multithreading. These are useful in longer semesters, or courses that require intermediate level study.

13. Input and Output Streams
14. Automatic Building and Testing with Ant
15. Java and the Database (JDBC)
16. Multithreading

The final four chapters are specific to building applications with a graphical user interface, and launching applications from a web server, and provide optional coverage for courses that have requirements for this type of programming. They provide additional resources and flexibility for longer or broader courses.

17. Building GUIs with the JFC Swing Library
18. Event-Driven Programming
19. Dialogs and Menus, Models and Views
20. Java Web Start and Applets

Supplemental Resources

A number of supplemental resources are available from the book's website at http://www.introjava.com

Resources on the website for students include
- Downloadable source code for all the examples in the book
- Downloadable source code for solutions to selected exercises
- Self-test questions

Additional resources for instructors include
- A complete set of PowerPoint slides
- Downloadable source code for solutions to all exercises

A note about the code
Source code in the text appears in a Courier font to mark it out clearly from the surrounding text

```
Java source code appears in this font
```

Due to the page width, it has often proved necessary to break lines of code in places where the original source code (which can be downloaded from the website) would not have a line break.

In the majority of cases the line breaks have been inserted so that they do not affect compilation. For example, this code statement appears in Chap. 3.

```
double mean =
 ((double)intArray[0] + intArray[1] + intArray[2])
   / intArray.length;
```

This is, in fact, a single statement, even though it is broken across three lines of text. A single statement in Java is terminated by a semicolon, and line feeds do not, in most cases, affect the way the code works. However in a very small number of cases it has not been possible to break lines within the margin constraints of the book in such a way that their workings are unaffected. If in doubt, or if you are having problems with compiling or running code, please refer to the original source code files.

Acknowledgements

It is difficult to acknowledge all the individuals who have contributed to this book, because my experience of Java has been so long and broad. I am grateful to the many authors of courseware from my various past employers: The Object People, BEA Systems, Valtech, IBM and Software Education Associates. The experience of teaching from material authored by others, however good or bad it is, provides new perspectives and understanding that goes way beyond what is possible when only teaching from your own perspective.

I am grateful to my colleague Hokyoung Ryu, for giving me the impetus to (re) write this book and test the material on his students.

I am also grateful to Wayne Wheeler and Simon Rees at Springer-Verlag London Ltd., who gave me the opportunity to bring this book to publication.

Contents