Preface

The origins of this book lie in supplementary material for a basic programming course that is offered in the first two semesters of the commercial IT course of study at the NORDAKADEMIE Business School. Since the winter semester of 1999/2000, *object-oriented programming* has been taught using the *Smalltalk* programming language. It was always difficult to decide which text books to assign to students, because the many books that were available to teach object-oriented programming in general and Smalltalk in particular often presupposed at the very least programming knowledge in a non-object-oriented programming language. Even though the number of beginning students who have such knowledge continues to increase, it's not something that one can generally assume to be true. In fact, when the first edition of this book appeared, texts for beginning programming students generally did not teach object-oriented programming. In the meantime, even though the "Objects-First" approach has become increasingly common, it's still relatively rare to encounter Smalltalk as a first programming language. For that reason, this book continues to fill a gap. It is intended not just for IT students, but for anyone who seeks a fundamental entry point for programming, especially object-oriented programming. For example, it can also be used for advanced IT courses in the later grades of secondary schools.

Choosing Smalltalk is important above all for pedagogical reasons. Smalltalk is a simple, strictly object-oriented language that practically forces one to think in an object-oriented way. In addition, almost all students can be said to suffer the same disadvantage, that of not knowing this language, which somewhat reduces the problems caused by the different levels of initial knowledge among the students. And furthermore, a variety of free development environments are available, which students can easily install on their own computers.

With regard to its use in the real world of industry, Smalltalk is certainly not as important as, say, Java. On the other hand, so-called *dynamic* languages like Python and Ruby, which adapted many of their ideas from Smalltalk, are enjoying increasing popularity and wider distribution. Smalltalk is becoming increasingly important as an introductory programming language for children in elementary schools. In this context, one can mention the *Etoys* and *Scratch* projects that have both been implemented in the Smalltalk dialect Squeak. Squeak runs on nearly all available operating systems as well as on the OLPC[1] XO computers ("$100 laptops").

This book, however, is not a programming-language course in a narrow sense. Above all, it should not be treated as a complete presentation of the extensive Smalltalk class library. For that, one must consult the relevant documentation for the development environment being used. Nevertheless, it's also necessary to treat the basic concepts of a development environment, since Smalltalk programming always occurs within one. For beginners in particular, this can be an additional impediment. Stated somewhat simply, it used to be enough to be able to use a text editor and a compiler; now, besides learning the basics of programming, a student must also learn technical skills for dealing with a complex development environment.

An introduction to programming must also include fundamental knowledge of how to construct algorithms—a theme that is usually not dealt with in connection with object orientation. This is necessary because it is only through concerning oneself with the elementary problems of programming that one can develop an understanding for how computers work.

The first chapter teaches readers enough basic concepts of computer science so that they can begin to learn programming. Then, in the second chapter, a simple example provides an initial introduction to the basic construction of algorithms and how to construct them in a specific programming language (in this case, Smalltalk). Later chapters bring up the topic of constructing algorithms again and again.

While Chap. 2 ignores typical object-orientation concepts, they form the main topic of Chap. 3. At the same time, the chapter also introduces the basic elements of the Smalltalk programming language.

Chapter 4 deals with programming repetitions ("loops"), once again in the context of algorithms. In addition, important methods available in the Smalltalk language for using loops are described.

In order to make it easier for readers to perform practical exercises and to understand the examples provided in the text, Chap. 5 provides instructions for using the VisualWorks development environment, which is used for the programming examples shown in this book.

Chapters 6, 7 and 8 centre on classes, which are the central concept of object orientation. The components of a class definition in Smalltalk are given first, followed by a description of how to create new classes. This is followed by a description of important basic classes in a Smalltalk class library. This leads to the introduction of additional basic concepts of object orientation, such as inheritance and polymorphism.

An entire Chap. 10 is devoted to collection classes, because of their complexity and importance.

[1] OLPC means One Laptop per Child.

First though, Chap. 9 summarises information that was already presented in earlier chapters dealing with error messages in the compiler and the runtime system. If—as is likely the case—readers have been confronted with error messages as they reviewed examples in earlier chapters or attempted their own exercises, it might be helpful to read this chapter earlier than its position in this text.

Chapter 11 takes up in a systematic fashion important aspects of object-oriented programming with Smalltalk, some of which, such as blocks and inheritance, already appeared in earlier chapters.

Chapter 12 picks up the principle of recursion, an important aspect of algorithms that cannot be omitted from an introduction to programming.

Chapter 13 deals briefly with processing sequential internal and external streams. It also explains how to access files from Smalltalk programs.

Although space considerations prevent discussion in this book of the development of larger Smalltalk applications, Chap. 14 nevertheless provides a few elementary instructions for how to structure programs. Chapter 17 provides suggestions for sources for this and other topics. The author's website (https://brauer.nordakademie.de) also contains additional information to accompany this book.

Before that, though, Chap. 15 deals with writing component tests and their automatic execution.

The book does not attempt to contrast traditional, procedural development practice with object orientation. Readers who already have programming experience in procedural programming languages are advised to look at literature concerning the Oberon 2 programming language. Reiser and Wirth (1994), for example, provide an excellent description of the transition from imperative to object-oriented programming.

It is important to emphasize once again that no previous knowledge of programming is necessary for successfully working one's way through this book. It is nevertheless assumed that readers have basic skills in working with a windows-based operating system, such as Microsoft Windows or the Apple Mac OS.

A Comment on Notation

Programming text, to the extent that it is not shown as screen captures, appears as a `monospace font`. The same is true for various Smalltalk concepts, such as class or method names.

Names of menus and menu contents are shown in **boldface** type.

The Development Environment

The Smalltalk system VisualWorks was used to create the examples in this book. This is a commonly used, professional Smalltalk development environment distributed by Cincom Systems. A fully functional instructional version that runs on many platforms can be downloaded from Cincom's website, www.cincomsmalltalk.com. Although it is not essential, it will prove very helpful to readers to have access to this system. Screen captures in this book were taken from Version 7.6 of VisualWorks.

Acknowledgements

First of all, I want to thank the publishing company and especially Dr. Klockenbusch, whose engagement is responsible for the appearance of the first edition of this Smalltalk book at a time when the whole world was talking only of a programming language whose name bears a resemblance to coffee. I owe thanks to Sybille Thelen for her support in the publication of this third edition.

I want to thank the employees of the Georg Heeg Company for their critical review of the first draft of this book and for several valuable corrections, suggestions and improvements. I also want to thank my former colleague at the NORDAKADEMIE, Professor Kleuker. Katrin Schimmeyer and Helmut Guttenberg helped me greatly in my struggles toward an orthographically and syntactically correct text. I want to thank Jan Bartelsen for reviewing the third edition.

Elmshorn Johannes Brauer
August 2008

Addendum to the Preface for the Fourth Edition

For this fourth edition, various small changes and corrections were made throughout the text. In addition, the numerous screen captures for the VisualWorks development environment were updated. For this, I want to thank Cincom Systems, which made Version 7.10 available to me before its release. For her support in this, I want especially to thank Yvonne Schickel.

The most significant improvement, however, is Chap. 16, which deals with an introduction to the development of Web applications in Smalltalk using the framework system, Seaside. The motivation for this chapter rests not least in the fact that the bachelor's degree program for computer science at NORDAKADEMIE is now called "object-oriented development of Web applications."

For reviewing this material and for their valuable suggestions, I am especially grateful to the following people: Stefanie Jasser, Joachim Sauer, Heiko Rehder, Daniel Purrucker and Carsten Becke.

I also wish to thank Bernd Hansemann of Springer Vieweg for his support in the publication of this fourth edition.

Elmshorn
September 2013

Johannes Brauer

Addendum to the Preface for the First English Edition

George Schober from Cincom Systems translated the original text in just a few months. He did an excellent job, and I am very grateful to him.

The numerous screen captures for the VisualWorks development environment were updated again, and for this, I want to thank Cincom Systems who made Version 8.0 available to me prior to its release.

The whole translation project was managed by Yvonne Schickel from Cincom Systems in a proficient and efficient manner. Without her, there would be no English edition. I am deeply grateful to her. Additionally, it's thanks to her, that the book could be extended by Chap. 18, containing ten reports about Smalltalk applications in industry.

I also wish to thank Sybille Thelen and Bernd Hansemann of Springer Vieweg for their support in the publication of this English edition.

Elmshorn
October 2014

Johannes Brauer

http://www.springer.com/978-3-658-06822-6

Programming Smalltalk – Object-Orientation from the Beginning
An introduction to the principles of programming
Brauer, J.
2015, XXI, 429 p. 277 illus., 1 illus. in color., Softcover
ISBN: 978-3-658-06822-6