

## Preface

---

This book is written with two kinds of readers in mind: the complete novice and the person somewhat familiar with software development in Pascal, C++, or another programming language.

For the novice, *Java Methods* provides a solid introduction to object-oriented programming, an illustrated explanation of Java syntax, and some practice using library classes and packages for developing GUI (graphical user interface) applets and applications. While you may have to struggle a little with the details of syntax, you have the advantage on the conceptual level of a fresh, unprejudiced view of the object-oriented programming approach. It will be easy for you to embrace OOP without reservations and use Java in the most effective way.

For the reader who has done some programming in C++ or another language, the challenges are different. You are already familiar with the general elements of programming: program syntax, control structures, working with a compiler and other software development tools. If you are coming from C++, you will recognize declarations of variables, `if-else` statements, `for`, `while`, and `do-while` loops, classes, and many other elements of Java syntax. But when it comes to OOP, you'll need to combine your experience with the open mind of a beginner! For while it would be possible to emulate Pascal- or C-style programming in Java, that would completely miss the point.

Fortunately, OOP concepts are more straightforward than the terminology makes them appear. The notions of *objects* (entities that combine data elements and functions), *classes* (definitions of types of objects), *methods* (functions that handle certain tasks), *instantiation* (creating an object of a particular class), *inheritance* (one class extending the features of another class), *encapsulation* (hiding the implementation details of a class), *polymorphism* (calling correct methods automatically for specific objects disguised as more generic types), and *event-driven* applications (where the operating system, the user, or events in the program trigger certain functions) are actually quite intuitive and useful.

This book will teach Java and OOP through examples, exposing you to all three levels of the language — the syntax, the conceptual OOP level, and the intermediate layer of library classes and GUI tools — at once. Even for a novice, it would have been difficult and boring to learn the programming language syntax and control

statements without meaningful examples and case studies. (You can't learn to swim in a bathtub.) Instead we have provided more realistic examples, examples that apply OOP concepts and use GUI components. Our presentation strives to balance the technical details with clear explanations of fundamental concepts.

The examples may be daunting at first. Keep in mind that you are learning Java using an *immersion* method. Expect to *see* a lot from the very beginning, but don't try to understand all or even most of it. Only the highlighted little bits may make sense to you as you begin; the whole picture emerges gradually as you go along. The code in the early chapters, which you'll have to accept on faith at first, will make total sense by the end of the book. You will also pick up proper programming style, an element not mandated by formal Java language specifications but essential for writing readable professional programs.

We approach the library classes and packages in the same way. Of course you could learn them from Java's exhaustive online documentation, but a faster method, for the beginner and professional alike, is to look at examples. We provide more or less realistic examples from the outset, so that an inquisitive student can use them to start writing his or her own little programs with graphics and GUI right away. By the time we get to the more formal discussion of graphics (in Chapter 14) and GUI components (Chapter 15), you have already seen most of them in action in a case study or lab. Our examples use Java's Swing package, currently the standard one used by professionals.

Still, with all the examples and case studies, we leave a lot of work to you. This is not a *Java-in-n-days* book or an *n-hours-to-complete* book. It is a book for learning essential concepts and technical skills at a comfortable pace, for acquiring a repertoire of examples to work from, and for consulting once in a while when you start writing your own Java programs professionally or for fun.

Working through this book will not make you a Java expert right away — but it will bring you to the level of an entry-level Java programmer with a better than average understanding of the fundamental concepts. Object-oriented programming was invented to make software development more accessible to beginners, and *Java Methods* is written in that spirit.

Without further delay, let us begin learning object-oriented programming in Java!