Introduction to Java Programming

Danciu Gabriel Mihail

Editura Universității Transilvania din Brașov

2023

Contents

1 Introduction **1**

1.1 Purpose and Scope of the Book 1

1.2 Intended Audience 1

1.3 Book Organization 1

1.4 Why Learn Java? 2

2 Overview of Java **3**

2.1 What is Java? 3

2.2 Advantages of Java 5

2.3 What Can Java Offer? 6

2.4 A first example 7

2.5 Syntax errors, run-time errors 9

2.6 Role of Java in Software Development 10

2.7 Java Virtual Machine 10

2.8 Java Development Environment 11

2.8.1 Java Development Kit (JDK) 11

2.8.2 Integrated Development Environments (IDEs) 11

2.8.3 Other Java Development Tools 12

3 Basic Java Programming: Syntax and Structure **15**

3.1 Data Types and Variables 16

3.2 Operators 18

3.3 Control Structures 19

3.4 Arrays 23

3.4.1 Array Declaration and Initialization 23

3.4.2 Array Length 24

3.5 Understanding Strings in Java 24

3.5.1 Creating Strings 24

3.5.2 Common String Methods 24

3.5.3 Immutability of Strings 25

3.5.4 Common String Methods 25

3.6 Switch Statements 26

3.6.1 Basic Switch Statement 26

3.6.2 Switch with Strings 27
3.7 Methods 27
3.7.1 Method Declaration 27
3.7.2 Method Call 28
3.7.3 Method Overloading 28
3.7.4 Method Return 28
3.7.5 Recursive Methods 28

4 Object-Oriented Programming in Java 31
4.1 Classes and Objects 31
4.2 Functions 32
4.3 Encapsulation 33
4.3.1 Accessing members 34
4.3.2 Constructors 35
4.3.3 The keyword ”this” 37
4.3.4 Method Overloading 38
4.4 Inheritance 40
4.4.1 The role of constructors in inheritance 40
4.4.2 Method overriding 42
4.5 Object class 43
4.5.1 Overriding Methods 44
4.5.2 The `Cloneable` Interface 44
4.5.3 Concurrency Methods 45

5 Advanced Object-Oriented Programming in Java 47
5.1 Abstract classes 47
5.2 Interfaces 50
5.2.1 Interface Inheriting 51
5.2.2 Interface Implementation 52
5.2.3 Interface and asbtract classes 54
5.2.4 Interface and variables 54

6 Java Collections 57
6.1 Generics 59
6.2 ArrayList and LinkedList 60
6.2.1 The differences between ArrayList and LinkedList 61
6.2.2 Null Elements 62
6.2.3 Method Operations 62
6.3 HashMap and HashSet 62
6.3.1 Difference Between HashMap and HashSet 63

7 More Java Programming Concepts 69
7.1 Exceptions in Java . . . 69
7.1.1 Basic Exception Handling Structure . . . 69
7.1.2 Hierarchy of Exceptions . . . 70
7.1.3 Examples of Exceptions . . . 70
7.1.4 Strategies for Handling Exceptions . . . 73
7.1.5 Logging . . . 74
7.2 Concurrency . . . 75
7.2.1 Definition . . . 75
7.2.2 Difference from a Function . . . 75
7.2.3 Thread Characteristics . . . 75
7.2.4 Main Thread . . . 76
7.2.5 Creating a Thread . . . 76
7.2.6 Multiple Threads . . . 78
7.2.7 Using `isAlive` and `join` . . . 78
7.2.8 Synchronization . . . 80
7.2.9 Inter-Thread Communication . . . 82
7.3 Input and Output . . . 86
7.3.1 Input from Console . . . 89
7.3.2 Reading a String from the Console . . . 89
7.3.3 Writing to the Console . . . 90
7.3.4 Stream Classes . . . 90
7.3.5 Serialization . . . 92
7.4 Lambda Expressions . . . 93
7.4.1 Examples . . . 95

8 A closer look to Java Collections 99
8.1 TreeSet . . . 99
8.1.1 Red-Black Tree Properties . . . 99
8.1.2 TreeSet Methods . . . 100
8.1.3 TreeSet Examples . . . 100
8.2 Sorting Collections . . . 101
8.2.1 The Comparable Interface . . . 101
8.2.2 Comparator . . . 103
8.3 Dictionary, HashTable and Properties . . . 104
8.3.1 HashTable Class . . . 104
8.3.2 Properties Class . . . 106
8.4 Map . . . 107
8.4.1 Interface Map.Entry . . . 107
8.5 Collections Class . . . 109
8.5.1 Exercise 3: Using the `Collections` Class . . . 112
8.5.2 Exercise 4: Counting Word Occurrences . . . 112
8.5.3 Exercise 5: Using Properties . . . 112

8.5.4 Exercise 6: Map Entry . . . 113

9 Java Application Development 115

9.1 Graphical User Interfaces . . . 115

9.1.1 JavaFX Basics . . . 115

9.1.2 JavaFX Layouts . . . 116

9.1.3 Event Handling . . . 117

9.1.4 FXML . . . 117

9.2 Web Applications . . . 118

9.2.1 Servlets and JSPs . . . 118

9.2.2 Introduction to Spring Framework . . . 119

9.3 Exercises . . . 123

9.3.1 JavaFX . . . 123

9.3.2 Web Applications . . . 124

10 Java Tools and Libraries 127

10.1 Tools . . . 127

10.1.1 Integrated Development Environments . . . 127

10.1.2 Build Tools . . . 127

10.1.3 Third-Party Libraries . . . 127

10.2 Network Communication in Java . . . 128

10.2.1 Sockets . . . 128

10.2.2 URL and HttpURLConnection . . . 130

10.3 Database Access in Java . . . 130

10.3.1 Java Database Connectivity (JDBC) . . . 130

10.3.2 Hibernate ORM . . . 131

10.4 Messaging Frameworks in Java . . . 133

10.4.1 Java Message Service (JMS) . . . 133

10.4.2 RabbitMQ . . . 133

10.4.3 Other Messaging Frameworks . . . 134

11 Laboratory Exercises 137

11.1 Laboratory 1: Introduction . . . 137

11.1.1 Development Environments . . . 137

11.1.2 A Simple Example . . . 138

11.1.3 Variables . . . 138

11.1.4 Arrays . . . 139

11.2 Laboratory 2: Object Oriented Programming . . . 139

11.2.1 Introduction . . . 139

11.2.2 Project Description . . . 139

11.2.3 Code Snippets . . . 140

11.2.4 Exercises . . . 143

11.2.5 Additional Exercises: Database . . . 144

11.3 Laboratory 3: Inheritance . . . 144
11.3.1 Introduction . . . 144
11.3.2 Person Class . . . 145
11.3.3 Requirements . . . 145
11.3.4 CourseOperations Interface . . . 145
11.3.5 Requirements for CourseOperations Interface . . . 145
11.3.6 ManagerCourseOperations Interface . . . 146
11.3.7 Requirements for ManagerCourseOperations Interface . . . 146
11.3.8 Additional Exercises . . . 146
11.4 Laboratory 4: Collections . . . 146
11.4.1 Introduction . . . 146
11.4.2 Modify the Course Class . . . 147
11.4.3 Modify the CourseManager Class . . . 147
11.4.4 Modify MockClassesManager to Make It Functional . . . 147
11.4.5 Additional Implementations . . . 148
11.4.6 Example Modifications . . . 148
11.4.7 Exercises . . . 148
11.5 Laboratory 5: Exceptions and Generics . . . 149
11.5.1 Introduction . . . 149
11.5.2 Exceptions . . . 150
11.5.3 Additional Exercises . . . 151
11.6 Laboratory 6: I/O operations . . . 151
11.6.1 Introduction . . . 151
11.6.2 Example of Reading/Writing to a File . . . 152
11.6.3 Exercises . . . 153
11.7 Laboratory 7: Threads . . . 154
11.7.1 Introduction to Threads . . . 154
11.7.2 Thread . . . 154
11.7.3 Exercises . . . 155
11.8 Laboratory 8: File I/O with JavaFX . . . 155
11.8.1 Introduction . . . 155
11.8.2 Functional Requirements . . . 155
11.8.3 Exercises . . . 156
11.9 Laboratory 9: H2 Database and Spring Framework . . . 156
11.9.1 Introduction . . . 156
11.9.2 Functional Requirements . . . 156
11.9.3 Technology Stack . . . 157
11.9.4 Exercises . . . 157
11.10Laboratory 10: Networking, Messaging, and Apache Frameworks . . . 158
11.10.1 Introduction . . . 158
11.10.2 Functional Requirements . . . 158
11.10.3 Technology Stack . . . 158
11.10.4 Exercises . . . 158

Chapter 1

Introduction

This chapter provides an introduction to the book and its contents. We will explain what the book is about, who it is intended for, and what the reader can expect to learn from it.

1.1 Purpose and Scope of the Book

The purpose of this book is to provide a comprehensive introduction to Java programming for beginners. It covers the basic concepts and principles of Java programming, as well as more advanced topics such as application development and third-party libraries.

The scope of this book is limited to the Java programming language and its related tools and libraries. We will not cover other programming languages or technologies in depth, although we will make occasional comparisons and references to related technologies Schildt (2021).

1.2 Intended Audience

This book is intended for anyone who wants to learn Java programming, regardless of their prior programming experience. It is suitable for students, hobbyists, and professionals who want to learn a new programming language or improve their existing skills.

No prior programming experience is required, although familiarity with basic computer concepts such as file systems, data types, and algorithms is helpful.

1.3 Book Organization

This book is organized into several chapters, each covering a different aspect of Java programming. The chapters are organized as follows:
- Chapter 1: Introduction
- Chapter 2: Overview of Java
- Chapter 3: Basic Java Programming Concepts
- Chapter 4: Object-Oriented Programming Java

- Chapter 5: Advanced Object-Oriented Programming Concepts
- Chapter 6: Java Collections
- Chapter 7: More Java Programming Concepts
- Chapter 8: A closer look to Java Collections
- Chapter 9: Java Application Development
- Chapter 10: Java Tools and Libraries
- Chapter 11: Laboratory exercises

Each chapter is further divided into sections and subsections, which provide a more detailed explanation of the concepts covered in the chapter.

1.4 Why Learn Java?

Java is one of the most popular programming languages in the world, and for good reason. It is used by millions of developers to create a wide variety of applications, from desktop software to mobile apps and web services.

Some of the benefits of learning Java include:

- Job opportunities: Java is one of the most in-demand programming languages in the job market, and is used by many large companies such as Google, Amazon, and Oracle.
- Versatility: Java can be used to create a wide range of applications, from simple command-line tools to complex web applications and mobile apps.
- Robustness: Java's strict type checking and runtime checking help ensure that programs are free from errors and crashes.
- Portability: Java programs can run on different platforms without modification, thanks to the Java Virtual Machine (JVM).
- Security: Java's security model helps protect against malicious code and other security threats.

Overall, learning Java is a valuable skill for anyone interested in software development, and can open up many career opportunities.