Java SE 17: Programming Complete

Oracle

Live Training (também disponível em presencial)

- **Localidade:** Lisboa
- **Data:** 11 Dec 2023
- **Preço:** 2360 € (Os valores apresentados não incluem IVA. Oferta de IVA a particulares e estudantes.)
- **Horário:** Laboral das 09h30 - 16h30
- **Nível:**
- **Duração:** 30h

Sobre o curso

This comprehensive course is suitable for experienced programmers who want to learn the full Java programming language (Java SE), the language constructs, handle collections and concurrency, and utilize core language APIs and modularization.

It is designed for people with existing knowledge of computing and programming concepts, and who need to learn all aspects of Java SE in the shortest possible time. Students practice the skills learned in each lesson through hands-on practices.

Destinatários

- This course is intended for students with some programming experience and is a comprehensive training for the Java programming language.

Objetivos

After completing this course, you should be able to:

- Describe the object-oriented programming approach
- Explain Java syntax and coding conventions
- Use Java constructs and operators

- Use core Java APIs, such as Collections, Streams, IO, and Concurrency
- Deploy Java SE applications

Pré-requisitos

- Knowledge of basic computing concepts
- Knowledge and some experience with a programming language

Programa

- Introduction to Java
- Primitive Types, Operators, and Flow Control statements
- Text, Date, Time, and Numeric objects
- Classes and Objects
- Improved Class Design
- Implement Inheritance and Use Records
- Interfaces and Generics
- Arrays and Loops
- Collections
- Nested Classes and Lambda expressions
- Java Streams API
- Exception Handling, Logging, and Debugging
- Java IO API
- Java Concurrency and Multithreading
- Modules and Deployment
- Annotations
- Java Database Connectivity
- Java Security
- Advanced Generics
- Oracle Cloud Deployment
- Miscellaneous Java Topics

Introduction to Java

- Introduction to Java
- Object Oriented Principles
- Inheritance
- Use Access Modifiers

- Practices for Lesson 1: Overview

Primitive Types, Operators, and Flow Control statements

- Primitive Types, Operators, and Flow Control Statements
- Assignment and Arithmetic Operators
- Binary Number Representation
- Flow Control Using switch Construct
- Practices for Lesson 2: Primitive Types, Operators, and Flow Control Statements
- Practice 2-2: Use the if/else and switch Constructs and the Ternary Operator
- Practice 2-2: Use the if/else and switch Constructs and the Ternary Operator…ctnd

Text, Date, Time, and Numeric objects

- Text, Date, Time, and Numeric Objects
- Text Blocks
- Wrapper Classes for Primitives
- Local Date and Time
- Represent Languages and Countries
- Format and Parse Date and Time Values
- Practice 3-1: Explore String and StringBuilder Objects….Part 1
- Practice 3-1: Explore String and StringBuilder Objects….Part 2
- Practice 3-2: Use BigDecimal Class and Format Numeric Values
- Practice 3-3: Use and Format Date and Time Values
- Practice 3-4: Apply Localization and Format Messages

Classes and Objects

- Classes and Objects
- Local Variable Type Inference
- IntelliJ IDE: Introduction
- Practice 4-1: Create the Product Management Application
- Practice 4-2: Enhance the Product Class
- Practice 4-3: Document Classes

Improved Class Design

- Improved Class Design

- Access Modifiers Summary
- Enumerations
- Practice 5-1 and Practice 5-2
- Practice 5-3: Make Product Objects Immutable

Implement Inheritance and Use Records

- Implement Inheritance and Use Records
- Verify Object Type Before Casting the Reference
- Override Methods and Use Polymorphism
- Sealed Classes, Inheritance Under Control
- Factory Method Pattern
- Practice 6-1 and Practice 6-2
- Practice 6-2: Override Methods and Use Polymorphism….Cntd
- Practice 6-3: Create Factory Methods
- Practice 6-4 and Practice 6-5
- Practice 6-6: Explore Java Records
- Practice 6-7: Implement Sealed Classes with Records

Interfaces and Generics

- Interfaces and Generics
- Default, Private, and Static Methods in Interfaces
- Generics
- Practice 7-1 and Practice 7-2
- Practice 7-3: Test the Product Review Functionality

Arrays and Loops

- Arrays and Loops
- The Arrays Class
- Summary
- Practice 8-1: Allow Multiple Reviews for a Product

Collections

- Collections
- Create Set Object

- Create HashMap Object
- Access Collections Concurrently
- Practice 9-1: Organize Products and Reviews into a HashMap .
- Practice 9-2: Implement Review Sort and Product Search Features

Nested Classes and Lambda expressions

- Nested Classes and Lambda Expressions
- Define Lambda Expression Parameters and Body
- Practice 10-1: Refactor ProductManger to Use a Nested Class
- Practice 10-2: Produce Customized Product Reports

Java Streams API

- Java Streams API
- Bi-argument Variants of Functional Interfaces
- Aggregate Stream Data using reduce Operation
- Restrictions on Parallel Stream Processing
- Spliterator
- Practice 11-1: Modify ProductManager to Use Streams
- Practice 11-2: Add Discount Per Rating Calculation

Exception Handling, Logging, and Debugging

Exception Handling, Logging, and Debugging
Handling Exceptions
Normal Program Flow with No Exceptions
Practice 12-1: Use Exception Handling to Fix Logical Errors
Practice 12-2: Add Text Parsing Operations

Java IO API

- Java IO API
- Serializable Object Graph
- Working with Filesystems
- Delete Paths
- Practice 13-1: Print Product Report to a File
- Practice 13-2: Bulk-Load Data from Files
- Practice 13-3: Implement Memory Swap Mechanism

Java Concurrency and Multithreading

- Java Concurrency and Multithreading
- Manage Executor Service Life Cycle
- Writing Thread-Safe Code
- Summary
- Practice 14-1 and Practice 14-2
- Practice 14-3: Simulate Concurrent Callers .

Modules and Deployment

- Modules and Deployment
- JPMS Module Categories
- Open Module Content
- Multi-Release Module Archives
- Practice 15-1: Convert ProductManagement Application into a Module
- Practice 15-2: Separate Application into Several Modules

Annotations

- Annotations
- Annotations that Validate Design

Java Database Connectivity

- Java Database Connectivity
- Create and Execute Callable SQL Statements

Java Security

- Java Security
- Execute Privileged Code
- Erroneous Value Guards

Advanced Generics

- Advanced Generics
- Upper Bound Wildcard

Oracle Cloud Deployment

- Oracle Cloud Deployment
- Example of the Microservices Architecture for an Online Mobile Application
- Summary
- Practice E-1: Present Application Logic as a Service Using Helidon SE
- Practice E-1: Present Application Logic as a Service Using Helidon SE….Cntd

Miscellaneous Java Topics

- Miscellaneous Java Topics
- Bitwise Logical Operators
- Character Classes
- Using the replaceAll Method
- Stream Examples
- Java IO, File Watch Service
- Factory Design Pattern code demo
- Singleton Pattern code demo
- Bitwise Logical Operators code demo
- Regular Expressions code demo
- Threads deadlock, livelock, and starvation code demos
- Streams code demos
- Java IO, File Watch Service code demos