

Course code: **J/BP8**

Course title: **Best practices of object-oriented programming in Java 8**

Days: 3

Description:

Course intended for:

The training is for programmers, designers and architects, who would like to acquire, develop or systematize their knowledge in the field of best practices, rules and patterns of object-oriented programming in Java and learn about the new features introduced in Java 8.

Course objective:

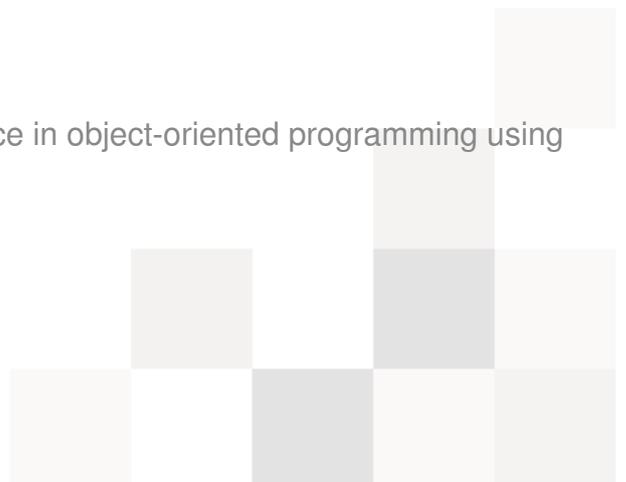
The training objectives include:

- Getting to know, understanding and exercising the best practices, principles and patterns of object-oriented programming in Java
- Ability to assess the code quality and alternative solutions
- The Clean Code rules
- The SOLID rules
- The GRASP rules
- The refactoring principles
- The selected patterns of GoF
- The selected features of Java 8

Requirements:

The training participant should have basic experience in object-oriented programming using Java and basic knowledge of UML.

Course parameters:



3*8 hours (3*7 net hours) of lectures and workshops (with a visible emphasis on workshops - 80%).

Group size: no more than 8-10 participants.

Course curriculum:

1. Introduction

- Why is the uniform mode of working with the code significant for the team?
- How do you measure the quality of the source code and the costs of low quality

2. A clean code

- Package, class, method naming
- Comments
- Classes
- Methods
- Data structures
- Management of exceptions and errors

3. GRASP patterns on the basis of problem-oriented workshops

- Low Coupling, High Cohesion
- Information Expert
- Creator
- Pure Fabrication
- Polymorphism
- Indirection
- Protected Variations



- Controller

4. SOLID rules on the basis of problem-oriented workshops with refactoring principles

- The Single Responsibility Principle
- The Open/Closed Principle
- The Liskov Substitution Principle
- The Interface Segregation Principle
- The Dependency Inversion Principle

5. GoF patterns on the basis of problem-oriented workshops with refactoring principles

- Building patterns: Builder, Prototype, Factory Method, Abstract Factory, Singleton
- Structural patterns: Facade, Proxy, Composite, Adapter, Decorator, Bridge
- Behavioral patterns: Command, Observer, State, Strategy, Chain of Responsibility, Mediator, Visitor, Template Method

6. Java 8

- Date/Time
- Collections
- Streams
- Lambda and functional programming fundamentals

