Restructuring and Incremental Code Improvement (Refactoring)

Duration: 2 days
Audience: Professionnel TI

Who is this training for?

Developers

Training objectives

Implement industry-leading class design techniques to improve the structure, readability, maintenance, and scalability of object software.

Summary

Improve the design and reuse of your classes (new and existing) in your object projects

Course outline

- Techniques for improving the design of existing codes without altering external behavior, principles and criteria for applying refactoring techniques
- Presentation of a catalog of implementation patterns for classes
- Reorganization of methods: how to extract, replace or add them
- Reorganization of attributes: how to create, change or replace accessors, values, references, Moving responsibilities using inline, delegated, or intermediate classes
- Simplifying conditional expressions: using subclasses, state/policy patterns, null object, and assertion
- Simplifying method calls to build interfaces and factories
- Moving methods in an inheritance hierarchy: how to extract and move attributes and methods

Approach and methodology

- Lectures
- Demonstrations
-

Copyright © 2026 AFI by Edgenda inc.

Guided and individual exercises

The distribution of the content is approximately: equipment 35% and laboratories 65%

Prerequisites

- Previous experience with an object-oriented programming language.
- Basic knowledge of software development and class design.
- Hands-on experience in development (desirable to get the most out of the exercises).

Recommendations

Review: the basics of clean code OOP concepts

Identify your challenges: code that is difficult to maintain duplication excessive complexity

Bring (if possible): an example of real code to improve

Copyright © 2026 AFI by Edgenda inc.