

SOLID principles in object development

Duration: 0,5 day

Audience: Professionnel TI

Who is this training for?

Designers, architects and developers.

Training objectives

- Quickly apply the best techniques for reducing dependencies between classes in order to improve structure, organization, and stability in your projects
- Improve the design of your applications to facilitate their maintenance and evolution

Summary

Eliminate the root causes of software coupling using five proven (SOLID) principles to produce more flexible and maintainable software architectures.

Course outline

This half-day course covers the following topics:

- Enumerating and describing the symptoms of poor design
- How dependencies hinder maintenance efforts, make extensibility difficult, prevent reuse, and restrict testing
- Key to the problem: Dependency management
- Introduction and practical application of advanced object-oriented design principles (SOLID) Single responsibility principle Open/closed principle Liskov substitution principle Interface segregation principle Dependency inversion principle

Approach and methodology

- Lectures
- Exercises
- Demonstrations

Prerequisites

- Essential concepts in object development, training
- Essential implementation patterns in object development or equivalent content

Recommendations

Experience in object-oriented programming