Design and Design patterns
advanced practices for object design

Hands-on course of 2 days - 14h
Ref.: PAT - Price 2024: €1 450 (excl. taxes)

The power of object design makes software modelling a specialist job where expertise can only gained through experience of and feedback from the field. This course will allow you to acquire operational skills in designing applications and to obtain improvements in productivity through the use of patterns. The many case studies will teach you to model and produce components and applications that can be upgraded and re-used, and to understand the main design patterns.

HANDS-ON WORK
The hands-on work modelling in UML and Java will allow the participants to handle the principles of design, identify the main patterns, and master the related tools.

THE PROGRAMME

last updated: 01/2018

1) Presentation of the design function
- Review of the basic notions of OO and UML programming. UML notation diagrams. Its advantages for design.
- The challenges for the design: increasing re-use without restraining enhancements.
- Re-use through inheritance: advantages and disadvantages.

2) The basic principles in object design
- The evolution strategy with the Open/Closed Principle (OCP).
- Effective re-use through inheritance and interfaces: the Liskov Substitution Principle (LSP)
- The concept of polymorphism.
- The impact of object design on project life cycles.
Hands-on work : Illustration of the division of responsibilities between classes.

3) The principles for organisation into packages
- The package as a design unit with the Reuse/Release Equivalency Principle (REP) and the Common Reuse Principle (CRP).
- Dividing up packages thanks to the Common Closure Principle (CCP).
- Organisation between packages: Acyclic Dependencies Principle (ADP) and Stable Dependencies Principle (SDP).
Hands-on work : The construction and hierarchical organisation of packages.

4) The principles for constructing classes
- Managing dependencies logically with the Dependency Inversion Principle (DIP).
- Reducing noticeable complexity with the Interface Segregation Principle (ISP).
- Allocating responsibilities with the GRASP principle.

5) The principles of design patterns
- The technical principles of designing an object application.
- How to re-use experience when designing and developing object applications: design patterns as software solutions.

- The origins and scope of the patterns.
- The advantages and limitations of design patterns.
- Design patterns as a response to technical problems.
- Resolving recurrent problems and ensuring a long life for developments.

6) Gamma and GoF: the founding principles
- The pattern catalogue of the "gang of four".
- The objectives and advantages.
- Isolating the creation of objects from their use with creational patterns for objects: factory, singleton and prototype.
- Refining the assignment of responsibilities through behavioural patterns: chain of responsibility, method pattern and ob
- Improving the structuring of classes with structure patterns: adapter, facade and composite.
Hands-on work : Example of designing and programming with the GoF patterns.

DATES

REMOTE CLASS
2025 : 17 Mar, 16 Jun, 18 Sep, 04 Dec