Essential concepts in object development

Duration: 0,5 day
Audience: Professionnel TI

Who is this training for?

Analysts, designers, architects, programmers and managers involved in development

Training objectives

This training aims to expose all the terms, foundations, concepts and best practices on the object. She clearly explains with concrete examples how these concepts are effectively integrated into projects. By the end of this training, you will be able to better understand object-oriented jargon and promote better exchanges and discussions in your team.

Summary

This training is aimed at all those who want to learn or definitively clarify all object-oriented concepts. It also features detailed examples in Java and C# as well as exercises and quizzes to ensure that you apply these concepts correctly in your projects. Since agile processes promote the overlap of developing roles, this training is a great opportunity to expose all the terms, foundations, concepts, and best practices about objects. Demystifying and better understanding this jargon encourages better two-way exchanges between your team members and managers.

Course outline

- Evolution and continuous improvement in software development
- Full coverage of concepts: objects, classes, components, instances, methods, properties, encapsulation, inheritance, overload, polymorphism, override, and composition
- Different forms of inheritance
- Reuse mechanisms: inheritance vs. composition, interface, subclassification, compatibility, is-one and a-one rules
- Capturing business knowledge with UML2:Classes and objects Members Instance and Class Relationships Relationships: Dependence, Generalization, and Association Navigation and Special

Cases of AssociationsClass Constraints and Invariants Context Importance Visibility, Properties, and Stereotypes Abstract Classes and Methods Interfaces and Packages

- Create a Document with Business Context
- Present the Characteristics for a Clean Architecture
- Apply the Rules for a Clean Architecture
- Advantages and Benefits of the Oriented Approach Purpose

Approach and methodology

- Simple presentation of basic concepts Step-by-step learning
- Clear explanations with concrete examples
- Link to real development situations
- Goal: Understand and apply easily

Prerequisites

- Basic computer knowledge Have already used a programming language (beginner level)
- Understand simple concepts such as variables, conditions and loops

Recommendations

- Knowledge of an object-oriented language (asset) Have done some exercises or small programming projects
- Be interested in structuring and organizing code