SETC Training Corporate Logo

UML Training Training: Object Oriented Analysis & Design (BAOOAD, 4 jours)

Description

The course Object Oriented Analysis & Design (UML Training Training) is a complete exploration of software design & UML. The training discusses the differences between waterfall & iterative methodologies while focusing primarily on iterative methods. The training includes a thorough discussion of business use cases, system use cases, UML and database design. The focus is on the development of high quality functional requirements in support of design. The course also includes a workshop which illustrates how a design can be implemented using Java, C# or C++ including design patterns.

Tarifs

- Tarification: $3,750/person
- Rabais de 10% lorsque vous inscrivez 3 personnes.

Plan de cours

Software Engineering with Components.

- What is a good system?
- Do we have a good system?
- What are good systems like? Encapsulation, Abstraction, Coupling!
- How are good systems built?

Object Concepts

- Defining terms: Containment, Inheritance, Polymorphism, Interface and Implementation, classes, etc
- What is an object? Messages, Interfaces, Classes.
- Inheritance.
- Polymorphism and Dynamic Binding.
- How does this help us achieve the goals of Section I.

The Rational Unified Process

- The Happy Video Store
- The Rational Unified Process
- The Inception Phase
- The Elaboration Phase
- The Construction Phase
- The Transition Phase

UML: Essentials of Class Models

- Introduction to UML
- Identifying objects and classes.
- Associations.
- Attributes and Operations.
- Generalization.
- The class model during the development.
- CRC cards.

UML: More On Class Models.

- More about associations: Aggregation and Composition, Roles, Navigability, Qualified Associations, Derived Associations, Constraints.
- More about classes: Interfaces, Abstract Classes.
- Parametrized classes.
- Dependency

UML: Use Case Models

- Actors in detail.
- Use cases in detail
- System boundary
- Relationships between use cases.
- Relationships between actors.
- Actors and classes.

UML: Interaction Diagrams.

- Collaborations.
- Interactions on Collaboration Diagrams.
- Sequence Diagrams.
- More advanced features: Messages from an object to itself, Detailed Behavior, Returned Values, Creation and Deletion of Objects, Timing.
- Other Purposes: providing an operation, how design patterns work, how a component works.
- Conditional Behavior and Iteration.
- Concurrency.

UML: State And Activity Diagrams.

- State Diagrams
- Activity diagrams.
- Other kinds of events.
- Other kinds of actions.
- Looking inside states.
- Concurrency within states.

UML: Implementation Diagrams

- Component Model
- Deployment Model

UML: Reuse: Components and Patterns.

- What can be reused and how?
- Why reuse?
- Why is reuse hard?
- Which components are really reusable?
- What difference does object orientation make?

Creating Objects with Java/VB.NET/C#

- Proper Coding
- Using Classes
- Creating Classes
- Class Properties
- Object Methods
- Constructors
- Destructors
- Static Data
- A complete Object example

Inheritance and Polymorphism

- Inheritance
- Polymorphism
- Abstract Functions and Classes
- Interfaces
- Common Inheritance Examples

UML: Design Patterns (If Time Permits):

- Modeling Design Patterns

Singleton
Abstract Factory
Facade
Visitor
Observer