

## Object Oriented Design, Design Patterns

CODICE	DT0075
DURATA	5 gg
PREZZO	1.990,00 €
EXAM	

### DESCRIZIONE

---

Alla fine del corso, i partecipanti conosceranno e sapranno applicare la metodologia di analisi CRC e le tecniche ed i principi in base ai quali modellare il sistema e progettarne l'architettura. Le giornate di formazione illustreranno, anche attraverso esercitazioni, i principali diagrammi UML dal punto di vista sintattico e semantico. Saranno utilizzati in aula strumenti RAD per la realizzazione dei diagrammi UML, ed esempi di codice.

### TARGET

---

Sviluppatori

### PREREQUISITI

---

Nozioni di base relative al processo di sviluppo software  
Conoscenze generali di programmazione Object Oriented, preferibilmente in ambiente .NET

### CONTENUTI

---

## Introduction to UML

- A brief History of UML
- Overview of issues in the field of object-oriented modeling
- UML overview

## Requirements management

- Requirements Types
- Requirements Categories (FURPS)
- Methods for gathering requirements
- Modeling requirements using UML
- The relationship matrix for the requirements
- Creating a requirements specification

# Modeling business processes

- Activity Diagram
- Business process modeling in UML
- The definition of a business process
  - Concurrent flows and decisions
  - Exceptions and Exception Handling
  - Partition, fork, join and other elements

# Modeling non-functional requirements

- Components and Deployment diagrams
- The initial architecture of the system - logical and physical
- Modeling requirements for security, performance, reliability, ...

# Modeling functional requirements

- Modeling functionality with the Use Case diagram
- Determining the scope of the system
  - Actors and the relationships between them
  - Identifying use cases
  - Association "actor - use case" and its properties
  - The relationship between use cases: include, extend, generalization
- Creating a use case scenarios and generate diagrams from them (activity)

# Analytical model of the system

- Using sequence diagrams
  - The types of messages: asynchronous, synchronous, reply
  - Categories of objects: Boundary, Control and Entity
- Modeling the interaction

# Static Modeling

- Class Diagram
  - Class, abstract class, interface
  - Association relationship and its characteristics.
  - Other relationships: aggregation, composition, generalization, dependency, association class
- Forward/Reverse engineering (OPTIONAL)
  - Generating source code from the model
  - Generating diagram based on the source code
  - Synchronizing code and diagram

# Dynamic Modeling

- Verification of the static model
  - Clarification of method signatures
  - Verification of the class diagram
- The dynamic modeling at the level of method calls
- Sequence diagram on design level
- State Machine diagram (OPTIONAL)

## Overview of other diagrams (OPTIONAL)

- Object Diagram
- Composite Structure Diagram
- Package Diagram
- Timing Diagram
- Communication Diagram
- Interaction Overview Diagram