Object Oriented Analysis and Design

Duration: 5 days **Price: € 2650**

Open Schedule

Audience Course Object Oriented Analysis and Design

The course Object Oriented Analysis and Design is intended for developers and architects who want to learn object oriented analysis and design techniques and UML to design systems.

Prerequisites Course Object Oriented Analysis and Design

To join the course Object Oriented Analysis and Design knowledge of the basic principles of object orientation is required and experience in object oriented software development is desirable.

Realization Training Object Oriented Analysis and Design

The subject matter is treated on the basis of presentation slides. During the course two case studies are developed from requirements to design. A modern UML tool is used to draw UML diagrams in it. The course material is in English. The course times are from 9.30 up and to 16.30.

Certification Object Oriented Analysis and Design

Participants receive an official certificate Object Oriented Analysis and Design after successful completion of the course.

Content Course Object Oriented Analysis and Design

In the course Object Oriented Analysis and Design you will learn the object oriented ways of thinking and techniques to analyze, design and model a software system as a collection of cooperating objects. The UML language runs as a central thread through the course.

Iterative and Incremental Development

After an introduction and review of the key object oriented concepts and principles, the modern system development principle of iterative and incremental development is discussed.

Requirements Gathering and Uses Cases

Next attention is paid to how the requirements of a system can be analyzed and how the typical forms of system use can be described with uses cases.

Domain Modeling

After an overview of UML, it is discussed how a domain model can be established, how the various objects can be distinguished together with their attributes and relationships, and what information they exchange.

Interaction Modeling

Attention is paid to how responsibilities can be assigned to objects and how these can be translated and made visible with interaction modeling using sequence and collaboration diagrams and state charts. The various patterns that can be used in this process are also discussed.

Packages and Subsystems

Part of the subject matter is also how the translation of the analysis model to a design class model can take place, including the design of a logical architecture with packages and subsystems and the mapping to code.

Architectural Design

The course also considers aspects of architectural design that are dealt with using component and deployment diagrams.

Design Patterns

Finally the focus is on the importance of design patterns to implement standard solutions.

SpiralTrain BV
Standerdmolen 10, 2e verdieping
3995 AA Houten

info@spiraltrain.nl
www.spiraltrain.nl
Tel.: +31 (0) 30 – 737 0661

Locations
Houten, Amsterdam, Rotterdam, Eindhoven, Zwolle, Online

SpiralTrain
developer courses

Modules Course Object Oriented Analysis and Design

Module 1 : Software Process	Module 2 : Requirements Analysis	Module 3 : Use Case Modeling
Software Development Process Software Development Phases Good Software Characteristics Iterative and Incremental Development Requirements Capturing Requirements Analysis Process System Design Test Driven Development Waterfall Model Evolutionary Development Unified Process	Understanding Requirements Vision Documents Requirement Analysis Activities Requirement Types Functional Requirements Non-Functional Requirements Requirements Determination Requirements Classification Conflicting Requirements Requirements Risks The glossary	Use Cases and Actors Identifying Actors System Context Diagram Identifying Use Cases Use Case Diagram Use Case Modeling Steps High Level Use Cases Alternative Paths Scenarios Generalizations include and extends
Module 4 : UML Overview	**Module 5 : Domain Modeling**	**Module 6 : Use Case Realization**
What is UML? UML Diagrams Use Case View Logical View Component View Deployment View Notes and Adornments Stereotypes Tagged Values Constraints System Sequence Diagrams	Why Domain Modeling? Conceptual Classes Noun Identification Physical and Conceptual Objects Types of Classes Domain Analysis Classes Finding Associations Multiplicity and Associations Generalization and Specialization Aggregation and Composition Finding Attributes	Realizing Requirements System Behavior Input Events and Operations System Sequence Diagrams Derivation from Use Case Postconditions Class Responsibilities Class Collaborations Interaction Diagrams Sequence Diagrams Grasps Design Patterns
Module 7 : Statecharts	**Module 8 : Design Model**	**Module 9 : Architectural Design**
State Machine Concepts State Machine Diagram Event Driven Behavior State Machines and Objects Object Behavior Objects and Threads Passive and Active Objects Entry and Exit Actions Internal Transitions State Activities Guards History States	Transition to Design From Requirements to Design Class Design Diagrams The Design Model Design Aspects Design Model Characteristics Mapping to Code Packages Package Design Packaging Guidelines Data Access Class Packages Subsystems	System Partitioning Large Scale Element Collaboration Layers and Packages Simple Logical Architecture Consider Coordination Layer Web Application Architecture Consider MVC Architecture Package Dependencies Clustering Vertical Scaling Horizontal Scaling Physical Architecture
Module 10 : Applying Design Patterns		
What are Patterns? Creational Patterns Behavioral Patterns Structural Patterns Architectural Patterns Singleton Abstract Factory Factory Method Reducing Dependencies Observer Pattern Adapter Pattern FaÇade pattern Proxy Pattern		

SpiralTrain BV
Standerdmolen 10, 2e verdieping
3995 AA Houten

info@spiraltrain.nl
www.spiraltrain.nl
Tel.: +31 (0) 30 – 737 0661

Locations
Houten, Amsterdam, Rotterdam, Eindhoven, Zwolle, Online