

CURSO UML

Object Oriented Analysis and Design with UML & RUP

Descripción: Nuestro curso Object Oriented Analysis and Design with UML & RUP está dividido en 2 secciones:

- Diagramas UML, donde se estudian los diagramas más importantes de UML 2.0 y técnicas de modelado.
- Metodología RUP, donde, con ayuda de UML se aprende a analizar y diseñar sistemas de software orientados a objetos. El alumno partirá desde el levantamiento de requerimientos y generación de primeros diagramas, hasta la entrega de diagramas de implementación, pasando por la evolución paulatina de los diferentes diagramas.

Objetivo: El alumno aprenderá las técnicas para analizar los requerimientos del mundo real y a diseñar soluciones que queden listas para codificar utilizando el lenguaje UML. Aprenderá a plasmar con casos de uso y diagramas de interacción el levantamiento de requerimientos del sistema a desarrollar; aprenderá a identificar y diseñar objetos, clases, y las relaciones entre ellos, incluyendo ligas, asociaciones y herencia. Asimismo, identificará la interrelación entre los diagramas UML y, utilizando la Metodología RUP, aprenderá a evolucionarlos para llegar a diagramas codificables en cualquier lenguaje orientado a objetos.

Audiencia: Analistas, diseñadores y programadores responsables de aplicar las técnicas Orientadas a Objetos en los proyectos de ingeniería de software.

Requisitos: Es recomendable que el alumno haya programado en algún lenguaje orientado a objetos.

Contenido

1.- INTRODUCTION TO ANALYSIS AND DESIGN

- Why is Programming Hard?
- The Tasks of Software Development
- Modules
- Models
- Modeling
- Perspective
- Objects
- Change
- New Paradigms

2.- OBJECTS

- Encapsulation
- Abstraction
- Objects
- Classes
- Responsibilities
- Attributes
- Composite Classes
- Operations and Methods
- Visibility
- Inheritance
- Inheritance Example
- Protected and Package Visibility
- Scope
- Class Scope

3.- ADVANCED OBJECTS

- Constructors & Destructors
- Instance Creation
- Abstract Classes
- Polymorphism
- Polymorphism Example
- Multiple Inheritance
- Solving Multiple Inheritance Problems
- Interfaces
- Interfaces with Ball and Socket Notation
- Templates

4.- CLASSES AND THEIR RELATIONSHIPS

- Class Models
- Associations
- Multiplicity
- Qualified Associations
- Roles
- Association Classes
- Composition and Aggregation
- Dependencies
- Using Class Models

10.- NEW MODELS IN UML 2.0

- New to UML 2.0
- Composite Structure Diagrams
- Timing Diagrams
- Interaction Overview Diagrams

11.- USE CASES

- Use Cases
- Use Case Diagram Components
- Use Case Diagram
- Actor Generalization
- Include and Extend
- Other Systems
- Narrative
- Template for Use Case Narrative
- Using Use Cases

12.- PROCESS

- Process
- Risk Management
- Test
- Reviews
- Refactoring
- History
- The Unified Process
- Agile Processes

13.- THE PROJECT

- Inception
- Elaboration
- Elaboration II
- Construction Iterations
- Construction Iterations - The Other Stuff

14.- DOMAIN ANALYSIS

- Top View – The Domain Perspective
- Data Dictionary
- Finding the Objects
- Responsibilities, Collaborators, and Attributes
- CRC Cards
- Class Models
- Use Case Models
- Other Models
- Judging the Domain Model

15.- REQUIREMENTS AND SPECIFICATION

5.- SEQUENCE DIAGRAMS

- Sequence Diagrams
- Interaction Frames
- Decisions
- Loops
- Creating and Destroying Objects
- Activation
- Synchronous & Asynchronous
- The Objects Drive the Interactions
- Evaluating Sequence Diagrams
- Using Sequence Diagrams

6.- COMMUNICATION DIAGRAMS

- Communication Diagrams
- Communication and Class Diagrams
- Evaluating Communication Diagrams
- Using Communication Diagrams

7.- STATE MACHINE DIAGRAMS

- What is State?
- State Notation
- Transitions and Guards
- Registers and Actions
- More Actions
- Internal Transitions
- Superstates and Substates
- Concurrent States
- Using State Machines
- Implementation

8.- ACTIVITY DIAGRAMS

- Activity Notation
- Decisions and Merges
- Forks and Joins
- Drilling Down
- Iteration
- Partitions
- Parameters and Pins
- Expansion Regions
- Using Activity Diagrams

9.- PACKAGE, COMPONENT, AND DEPLOYMENT DIAGRAMS

- Modeling Groups of Elements – Package Diagrams
- Visibility and Importing
- Structural Diagrams
- Components and Interfaces
- Deployment Diagram

- The Goals
- Understand the Problem
- Specify a Solution
- Prototyping
- The Complex User
- Other Models
- Judging the Requirements Model

16.- DESIGN OF OBJECTS

- Design
- Factoring
- Design of Software Objects
- Features
- Methods
- Cohesion of Objects
- Coupling between Objects
- Coupling and Visibility
- Inheritance

17.- SYSTEM DESIGN

- Design
- A Few Rules
- Object Creation
- Class Models
- Interaction Diagrams
- Printing the Catalog
- Printing the Catalog II
- Printing the Catalog III
- Object Links
- Associations

18.- REFACTORING

- Refactoring
- Clues and Cues
- How to Refactor
- A Few Refactoring Patterns

APPENDIX A – UML SYNTAX

APPENDIX B – DESIGN BY CONTRACT

- Contracts
- Enforcing Contracts
- Inheritance and Contracts

APPENDIX C– IMPLEMENTATIONS

- C++
- Java
- C#

Incluye:

- ❖ Una mochila de la empresa
- ❖ Un manual original e importado de UML
- ❖ Una licencia de la última versión Desktop de Enterprise Architect
- ❖ Un lápiz

Formas y condiciones de pago:

Para la inscripción, aplicación de los descuentos y aseguramiento de que el curso se impartirá en las fechas pactadas se requiere el pago anticipado antes con depósito o transferencia bancaria a la siguiente cuenta:

- ❖ Depósito en Banamex cuenta 4923239 Suc. 575 a nombre de Desarrollo y Capacitación en Internet, S. A. de C. V. (CLABE en caso de transferencia electrónica vía Internet 002180057549232394)
- ❖ O cheque a nombre de Desarrollo y Capacitación en Internet, S.A. de C.V.