

Unified Modeling Language 2.0

Length: 24 – 32 hours, depending on the practical part

Required facilities: VGA projector, white board, one workstation (per two programmers), development tools for writing programs in an object oriented language (Java, C++, C#, etc.)

Who can participate: programmers who are familiar with object oriented programming

Course objectives: learn UML, mainly the diagram types, in order to express a software construct or behavior, to document a software system


Related courses: Design Patterns, Object Oriented Programming, object oriented programming languages (C++, Java, C#)

Attendees' evaluation: optional, during the training and/or a final test

Course written support: yes

Description: the course offers a theoretical and practical approach of UML 2.0 towards its usage by a programmer. The examples, case studies, hands on assignments offer a good understanding of UML diagrams. There are performed the following activities:

- Presentation of the main graphical elements, diagram types, their semantics and how they are used
- Understanding, „reading” the UML diagrams which were built by others
- Build of diagrams in order to express structures (static aspects) or behaviors (dynamic aspects)
- UML diagrams implementation („translation”) in one object oriented language (C++, Java, C#) in order to emphasize variants and particularities related to that language



Even UML is not bound to a particular software development process, it is important the understanding of modeling particularities of every phase or activity type belonging to the software development. To illustrate this, the Rational Unified Process (RUP) is shortly presented; there are emphasized modeling perspectives, views on a software system.

The course is not based on any particular modeling tool (for example Rose) or any editing tool of the UML diagrams..

Contents:

1. Introduction
2. Software Development Processes
3. Class diagrams
4. Sequence diagrams
5. Object diagrams
6. Package diagrams
7. Deployment diagrams
8. Use cases diagrams
9. State machine diagrams
10. Activity diagrams
11. Communication diagrams
12. Composite structures
13. Component diagrams
14. Collaborations
15. Interaction overview diagrams
16. Timing diagrams