

Unified Modeling Language 2.0

Durata: 24 – 32 ore (funcție de ponderea părții practice)

Infrastructura folosită: retroproiector, tabla, calculator personal pe care este instalat un mediu de dezvoltare Java (Eclipse), C++ sau C# (Microsoft Visual C++/C# Express Edition, Microsoft Visual Studio 2009, etc).

Cine poate participa: programatori ce doresc să se familiarizeze cu folosirea UML în activitatea de programare; este necesară cunoașterea unui limbaj orientat pe obiecte (de preferință Java, C++ sau C#) precum și noțiunile de bază ale programării orientate pe obiecte.

Efectul cursului: participanții se vor familiariza cu UML, în principal vederi și tipuri de diagrame, pentru a exprima o structură sau o comportare în programare, pentru a documenta un sistem software.

Cursuri conexe: Șabloane de proiectare, Programare orientată pe obiecte, limbaje de programare orientate pe obiecte (C++, Java, C#)

Verificarea cunoștințelor: opțional, pe parcurs și/sau test final

Suport de curs: da

Descriere: Cursul oferă o abordare teoretică și practica a UML 2.0 cu accent pe aplicarea lui în activitatea de programare. Exemplele, temele și problemele practice oferă o înțelegere bună a modului de folosire a diagramelor UML în principal prin următoarele activități:

- Parcurgerea și înțelegerea principalelor elemente grafice, tipurilor de diagrame și contextul în care acestea se folosesc
- Înțelegerea, interpretarea diagramelor UML construite de alții
- Construirea de diagrame pentru a exprima o structură (aspecte statice) sau un comportament (aspecte dinamice)
- Traducerea diagramelor în cod folosind un limbaj orientat pe obiecte (C++, Java, C#) pentru a vedea particularități sau variante legate de un limbaj de programare particular

Chiar dacă UML nu este legat de nici un proces de dezvoltare software, este importantă înțelegerea particularităților de modelare pentru fiecare fază sau tip de activitate ce țin de dezvoltarea de software. Pentru acest lucru se prezintă pe scurt RUP – Rational Unified Process, se pun în evidență perspectivele de modelare, vederile asupra unui sistem software.

Cursul nu este legat de nici o unealtă CASE (cum ar fi ROSE) sau unealtă particulară de editare a diagramelor UML.

Conținut:

1. Introducere
2. Procese de dezvoltare software
3. Diagrame de clase
4. Diagrame de secvență
5. Diagrame de obiecte
6. Diagrame de pachete
7. Diagrame de instalare
8. Cazuri de folosire
9. Diagrame state machine
10. Diagrame de activitate
11. Diagrame de comunicare
12. Structuri compozite
13. Diagrame de componente
14. Colaborări
15. Diagrame sintetice de interacțiune
16. Diagrame de timp