

IS – LAB1

1. Ce este UML si pentru ce este utilizat?

UML este un limbaj de modelare generalizat standardizat in domeniul ingineriei software. Standardul este gestionat si a fost creat de catre OMG (Object Management Group). Acesta include un set de tehnici de notare grafica pntru a crea modele vizuale de sisteme intensive in software.

2. Ce sunt modelele si care este utilitatea lor?

Modelele sunt reprezentări simplificate ale realității, create pentru a înțelege, comunica și prezice comportamentul sistemelor complexe. Ele permit abstractizarea, concentrându-se pe aspectele esențiale și ignorând detaliile nerelevante, ceea ce face sistemele mai ușor de gestionat și înțeles. Modelele sunt folosite în diverse scopuri, inclusiv ca schițe pentru comunicarea ideilor, ca planuri detaliate pentru construcția de sisteme sau ca programe executabile, unde modelele pot fi transformate direct în cod. Această abordare ajută la reducerea complexității și crește eficiența în dezvoltarea și întreținerea sistemelor software.

3. Ce tipuri de diagrame sunt prezentate in [1] – Capitolul 2? Care sunt asemanarile si deosebirile intre acestea?

Software Engineering Mentor

În capitolul 2 din UML@Classroom, sunt prezentate 14 tipuri de diagrame UML, împărțite în două categorii principale: diagrame de structură și diagrame de comportament:

Diagrame de structură:

1. Diagrama de clase
2. Diagrama de pachete

3. Diagrama de obiecte
4. Diagrama de componente
5. Diagrama de structură compozițională
6. Diagrama de desfășurare
7. Diagrama de profil

Diagrame de comportament:

1. Diagrama de cazuri de utilizare
2. Diagrama de stare (State Machine Diagram)
3. Diagrama de activități
4. Diagrame de interacțiune, care includ:
 - Diagrama de secvență
 - Diagrama de comunicare
 - Diagrama de timp (Timing Diagram)
 - Diagrama de prezentare a interacțiunii (Interaction Overview Diagram)

Asemănări:

- Toate sunt utilizate pentru a reprezenta diferite aspecte ale sistemelor software într-o manieră vizuală și structurată.
- Facilitează comunicarea și înțelegerea sistemului între diferiți stakeholderi, cum ar fi dezvoltatorii, analiștii de sistem și clienții.
- Sunt bazate pe aceleași principii și concepte fundamentale ale UML, oferind o coerență în reprezentarea sistemelor.

Deosebiri:

- Diagramele de structură sunt axate pe reprezentarea elementelor statice ale sistemului, cum ar fi tipurile de date, componentele și relațiile structurale dintre acestea.

- Diagramele de comportament descriu dinamica sistemului, adică modul în care elementele sistemului interacționează și se schimbă în timp, inclusiv fluxul de control și de date.
- Fiecare tip de diagramă are un scop specific și este utilizat pentru a evidenția aspecte diferite ale sistemului, de la structura internă și organizarea codului până la interacțiunile dintre utilizatori și sistem sau evoluția stărilor unui obiect în timp.

4. Caracterizați succint fiecare diagrama prezentată în [1] – Capitolul 2 (cele 5 menționate mai sus)

1. **Diagrama de clasă (Class Diagram):** Se bazează pe conceptele de clasă, generalizare și asociere pentru a specifica structurile de date și structurile obiectelor unui sistem. Profesorii predau cursuri, iar studenții participă la cursuri, având proprietăți comune, cum ar fi apartenența la clasa Persoană, exprimată printr-o relație de generalizare.
2. **Diagrama de cazuri de utilizare (Use Case Diagram):** Definește cerințele pe care un sistem trebuie să le îndeplinească, descriind ce funcționalități utilizează diferiți utilizatori ai sistemului, fără a adresa detalii specifice implementării.
3. **Diagrama de activitate (Activity Diagram):** Modellează procesele de orice fel, atât procesele de afaceri, cât și procesele software, oferind mecanisme de control al fluxului și mecanisme de flux de date care coordonează acțiunile ce compun o activitate sau un proces.
4. **Diagrama de secvență (Sequence Diagram):** Descrie interacțiunile dintre obiecte pentru a îndeplini o sarcină specifică, concentrându-se pe ordinea cronologică a mesajelor schimbate între partenerii de interacțiune. Aceasta permite modelarea interacțiunilor complexe prin diferite construcții pentru controlul ordinei cronologice a mesajelor și concepte pentru modularizare.
5. **Diagrama mașinii de stare (State Machine Diagram):** Reprezintă comportamentul permisibil al unui obiect sub forma stărilor posibile și a tranzițiilor de stare declanșate de diverse evenimente, ilustrând cum stările obiectelor se schimbă în timp.