Laborator 1

2. Ce este UML și pentru ce este utilizat?

UML este un limbaj de modelare generalizat standardizat în domeniul ingineriei

software. Standardul este gestionat și a fost creat de OMG (Object Management Group).

Include un set de tehnici de notare grafică pentru a crea modele vizuale de sisteme

intensive în software. Este utilizat de programatori pentru a specifica, a vizualiza, a construi și a documenta sisteme de software.

3. Ce sunt modelele și care este utilitatea lor?

O diagramă de clasă modelează clasele unui sistem/aplicație care urmează să se dezvolte

O diagramă de secvență este folosită în primul rând pentru a arăta interacțiunile dintre obiectele care sunt reprezentate ca linii de viață într-o ordine secvențială.   
Diagrama componentelor arată aspectul fizic al unui sistem software orientat pe obiecte.  
Diagrama de implementare arată aspectele fizice ale unui sistem orientat pe obiecte.  
Diagrama pachetului arată aranjarea și organizarea elementelor modelului într-un proiect la scară medie până la mare.

4. Class Diagram (Diagrama de clasă) – Reprezintă structura statică a sistemului, evidențiind clasele, atributele, metodele și relațiile dintre ele (asociere, moștenire, compunere, agregare etc.).

Use Case Diagram (Diagrama de cazuri de utilizare) – Ilustrează interacțiunea dintre actorii externi și sistem, arătând ce funcționalități sunt oferite.

Activity Diagram (Diagrama de activitate) – Modelează fluxul de activități într-un proces, similar cu o diagramă de flux, fiind utilă pentru a descrie logica unui caz de utilizare.

Sequence Diagram (Diagrama de secvență) – Reprezintă interacțiunea dintre obiecte într-o secvență temporală, utilizată pentru a arăta ordinea apelurilor de metode între diferite entități.

State Machine Diagram (Diagrama mașinii de stare) – Modelează diferitele stări prin care poate trece un obiect în timpul execuției și tranzițiile dintre acestea.

5.Asemănări între diagrame:

Toate sunt utilizate în UML pentru modelarea și înțelegerea unui sistem software.

Reprezintă diferite aspecte ale sistemului, fie structural (Class Diagram), fie comportamental (Use Case, Activity, Sequence, State Machine).

Ajută la clarificarea cerințelor, la documentare și la comunicarea între echipele de dezvoltare.

Deosebiri între diagrame:

Diagrama de clasă este statică și arată structura sistemului, pe când celelalte diagrame sunt dinamice și descriu comportamente.

Diagrama de cazuri de utilizare este orientată spre interacțiunea dintre utilizatori și sistem, în timp ce diagrama de activitate se concentrează pe fluxul de procese.

Diagrama de secvență detaliază ordinea mesajelor între obiecte într-un scenariu specific, pe când diagrama mașinii de stare descrie tranzițiile unui obiect între diferite stări.

6.

Class Diagram (Diagrama de clasă)

Reprezintă structura statică a sistemului.

Conține clase, atribute, metode și relațiile dintre clase (asociere, moștenire, compunere, agregare).

Este esențială pentru proiectarea arhitecturii software.

Use Case Diagram (Diagrama de cazuri de utilizare)

Ilustrează interacțiunea dintre utilizatori (actori) și sistem.

Arată funcționalitățile oferite de sistem și relațiile dintre cazurile de utilizare.

Utilizată pentru analiza cerințelor și specificația funcțională.

Activity Diagram (Diagrama de activitate)

Modelează fluxul de activități într-un proces, similar cu o diagramă de flux.

Utilizată pentru a descrie logica unui caz de utilizare sau a unui algoritm.

Poate include decizii, paralelisme și puncte de început/sfârșit.

Sequence Diagram (Diagrama de secvență)

Reprezintă interacțiunea dintre obiecte într-o ordine temporală.

Arată mesajele schimbate între entități și ordinea lor.

Utilizată pentru analiza proceselor și înțelegerea fluxului de execuție.

State Machine Diagram (Diagrama mașinii de stare)

Modelează stările unui obiect și tranzițiile dintre acestea.

Evidențiază evenimentele care determină schimbările de stare.

Utilizată pentru sisteme care implică comportamente dinamice complexe.



