

Tema Laborator 1-IS

- Ce este UML si pentru ce este utilizat?

UML, cunoscut și sub numele de Unified Modeling Language, este un limbaj standardizat pentru specificarea, documentarea, proiectarea și modelarea sistemelor software și a altor sisteme complexe. Creat inițial de Grady Booch, Ivar Jacobson și James Rumbaugh în anii 1990, UML a fost adoptat ca standard de către Object Management Group (OMG) și este larg utilizat în industria software.

UML este utilizat pentru o varietate de scopuri în dezvoltarea software-ului, inclusiv:

- Modelarea conceptuală: UML permite dezvoltatorilor să creeze modele conceptuale ale sistemului pe care îl dezvoltă. Acest lucru include identificarea entităților cheie, a relațiilor între aceste entități și a funcționalităților sistemului.
- Proiectarea arhitecturii: UML este folosit pentru a proiecta arhitectura software-ului, inclusiv structura sistemului, modulele componente și relațiile dintre acestea.
- Comunicarea între echipe: UML oferă un limbaj comun pentru dezvoltatori și alte părți interesate să comunice și să înțeleagă cerințele, proiectele și structura sistemului într-un mod clar și consistent.
- Documentarea software-ului: UML poate fi utilizat pentru a genera documentație detaliată despre sistem, inclusiv diagrame care descriu structura sistemului, comportamentul său și relațiile între componentele sale.
- Testarea și validarea: UML poate fi utilizat pentru a defini și planifica teste de validare și verificare a sistemului, ajutând la asigurarea calității și la identificarea potențialelor probleme înainte de implementare.

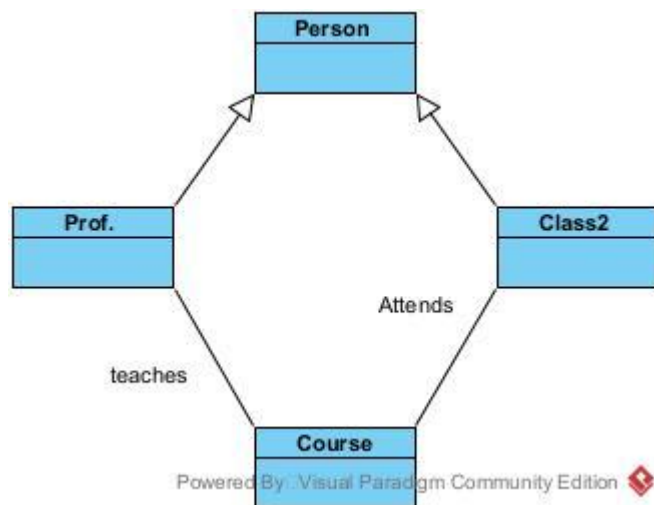
- Ce sunt modelele si care este utilitatea lor?

În ingineria software, modelele sunt reprezentări abstracte ale sistemelor software, proceselor și interacțiunilor acestora. UML oferă un set standardizat de diagrame și notații grafice pentru a descrie diferite aspecte ale unui sistem software.

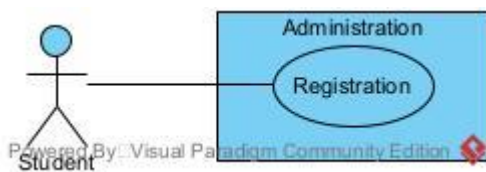
Utilitatea acestor modele constă în capacitatea lor de a simplifica și de a abstractiza complexitatea sistemelor software, permițând dezvoltatorilor să înțeleagă, să comunice și să gestioneze proiectele mai eficient. De asemenea, modelele UML facilitează colaborarea între membrii echipei și ajută la identificarea și rezolvarea problemelor înainte de implementarea efectivă a sistemului.

- Familiarizarea cu mediul de lucru mai sus amintit prin crearea anumitor diagrame din secțiunile 2.3.1 și 2.3.2 din [1]. Mai exact:

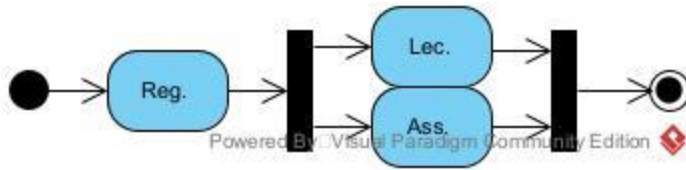
Class Diagram:



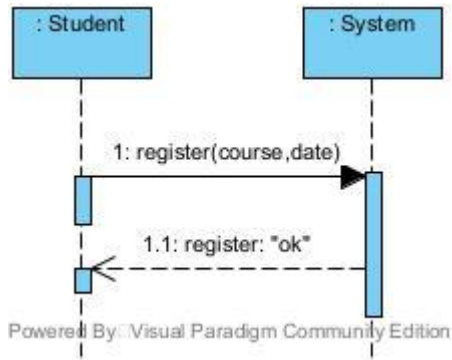
Use Case Diagram:



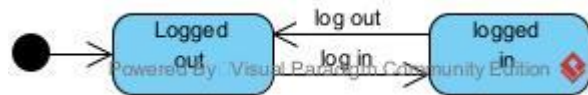
Activity Diagram:



Sequence Diagram:



State Machine Diagram:



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

- Class diagram
- Object diagram
- Package diagram
- Component diagram
- Composition structure diagram
- Deployment diagram
- Profile diagram
- Use case diagram
- State machine diagram
- Activity diagram
- Sequence diagram
- Communication diagram
- Timing diagram

- Interaction overview diagram

Caracterizati succint fiecare diagrama prezentata in [1] – Capitolul 2 (cele 5 mentionate mai sus)

1) Diagrama de clasă (Class diagram):

Folosită pentru a reprezenta structura statică a sistemului.

Arată clasele din sistem, atributele și relațiile între ele.

Include de obicei atributele clasei și metodele publice.

2) Diagrama de cazuri de utilizare (Use case diagram):

Folosită pentru a descrie comportamentul sistemului din perspectiva utilizatorului.

Definește cazurile de utilizare (scenariile) ale sistemului și actorii care interacționează cu aceste cazuri de utilizare.

3) Diagrama de activitate (Activity diagram):

Folosită pentru a descrie fluxurile de lucru sau procesele sistemului.

Arată activități și acțiuni în cadrul unui proces, precum și deciziile și bifurcațiile din acesta.

Utilă pentru a înțelege secvența de acțiuni necesară pentru a finaliza o anumită activitate.

4) Diagrama de secvențe (Sequence diagram):

Folosită pentru a modela interacțiunile între obiecte într-un sistem.

Arată mesajele trimise între obiecte într-o secvență specifică de timp.

Utilă pentru a înțelege cum obiectele cooperează într-un scenariu specific.

5) Diagrama mașinii de stare (State machine diagram):

Folosită pentru a descrie comportamentul unui obiect sau al unui sistem ca o secvență de stări.

Arată stările posibile ale unui obiect și tranzițiile între aceste stări în urma evenimentelor.

Utilă pentru a înțelege cum un obiect răspunde la diferite evenimente și cum își schimbă starea de-a lungul timpului.

Figura 1:

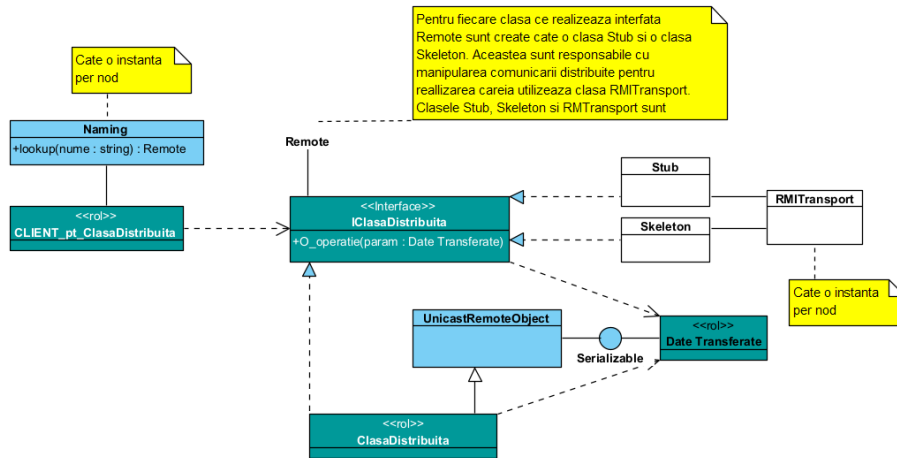


Figura 2:

