

## Laborator 1

### 1. Ce este UML și pentru ce este utilizat?

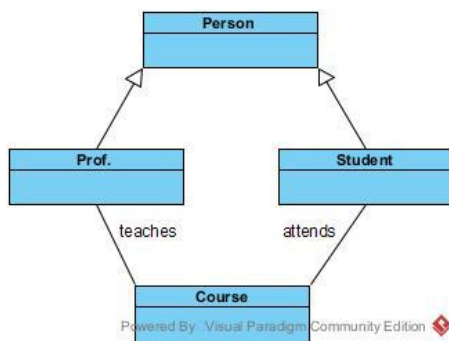
- Unified Modeling Language (UML) este un set standardizat de notări și simboluri utilizat în ingineria software și dezvoltarea sistemelor informatice. Scopul său este să ofere un cadru comun și o metodologie pentru proiectarea, documentarea și comunicarea aspectelor complexe ale sistemelor software, facilitând înțelegerea și colaborarea între membrii echipei de dezvoltare.

### 2. Ce sunt modelele și care este utilitatea lor?

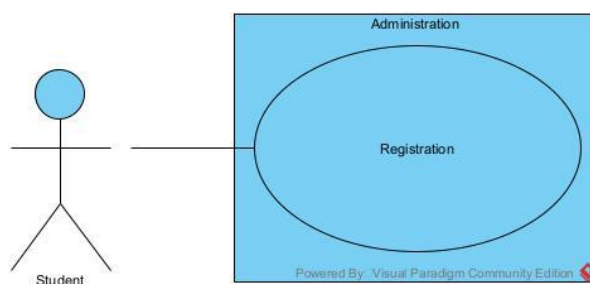
- Modelele sunt abstractizări ale sistemelor sau proceselor, concepute pentru a facilita înțelegerea, analiza, comunicarea și optimizarea acestora. În UML, un model este reprezentat grafic sub formă de diagrame. Modelele oferă o modalitate de a înțelege mai bine structura și comportamentul sistemului sau procesului, permițând identificarea relațiilor și dependențelor dintre diferitele componente. Acestea sunt folosite pentru a proiecta și dezvolta sisteme noi sau pentru a le îmbunătăți pe cele existente.

### 3. Familiarizarea cu mediul de lucru mai sus amintit prin crearea anumitor diagrame din secțiunile 2.3.1 și 2.3.2 din [1].

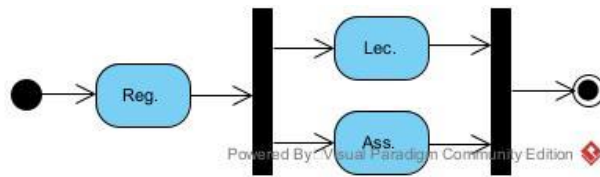
- Class diagram:



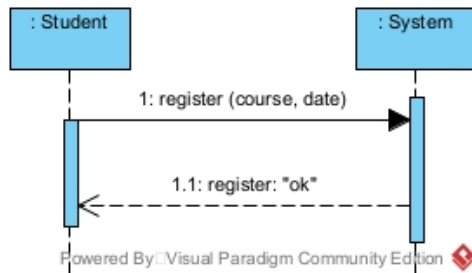
- Use case diagram:



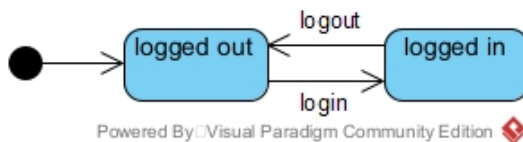
➤ Activity diagram:



➤ Sequence diagram:



➤ State machine diagram:



4. Ce tipuri de diagrame sunt prezentate în [1] – Capitolul 2? Care sunt asemănările și deosebirile între acestea?

- În capitolul 2 sunt prezentate diferite tipuri de diagrame UML, cele mai semnificative fiind cele 5 diagrame reprezentate la exercițiul anterior (class diagram, use case diagram, activity diagram, sequence diagram, state machine diagram). Toate aceste diagrame sunt folosite în ingineria software pentru a modela diverse aspecte ale unui sistem. Ele oferă o reprezentare vizuală a componentelor, interacțiunilor și comportamentelor sistemului, facilitând înțelegerea și comunicarea între echipele de dezvoltare.
- Cu toate acestea, există și diferențe semnificative între aceste diagrame, pe lângă reprezentarea vizuală a acestora:
  - use case diagram: se concentrează pe funcționalitățile sistemului și interacțiunile cu utilizatorii
  - class diagram: descrie structura statică a sistemului, inclusiv clasele, atributele și relațiile între acestea

- state machine diagram: modelează comportamentul obiectelor prin definirea stărilor și tranzițiilor între acestea
- sequence diagram: descrie interacțiunile între obiecte în ordinea cronologică a mesajelor schimbate
- activity diagram: reprezintă procesele sau fluxurile de lucru ale sistemului, evidențiind acțiunile necesare pentru completarea activităților

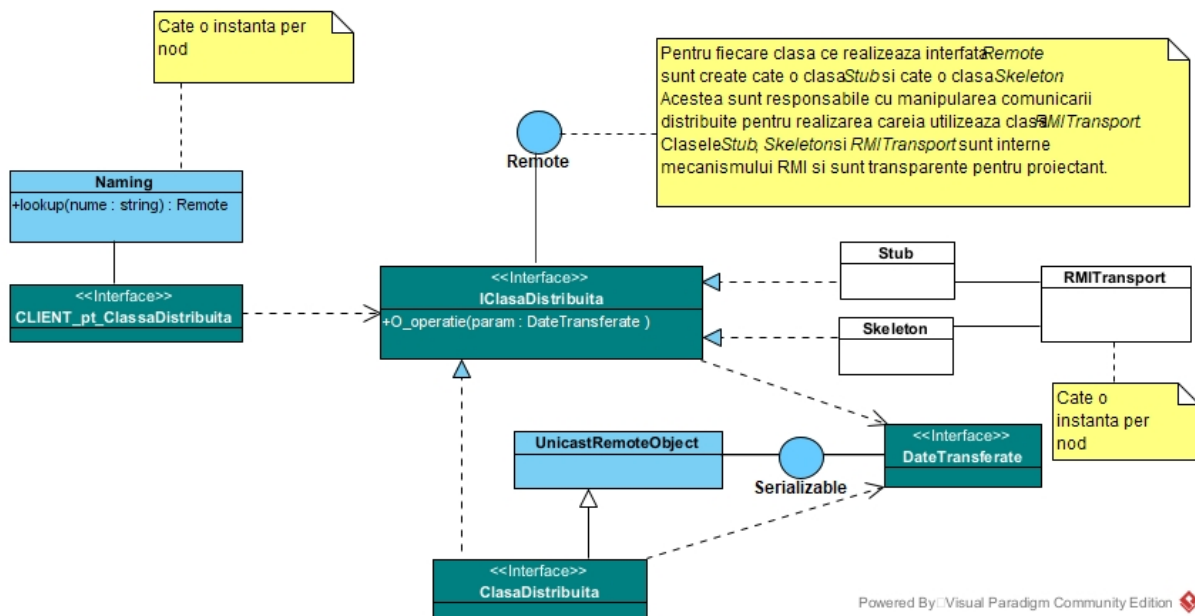
- Astfel, fiecare diagramă are un scop specific și oferă o perspectivă diferită asupra sistemului, contribuind la dezvoltarea și înțelegerea acestuia din diverse perspective.

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

- *Class diagram:*
- reprezintă structura statică a unui sistem software, incluzând clasele și relațiile dintre acestea
  - se bazează pe clase, generalizare și asociere
  - permite specificarea datelor și a structurii obiectelor
- *Use case diagram:*
- definește cerințele sistemului
  - descrie funcționalitățile sistemului utilizate de diferiți utilizatori, fără a aborda detalii specifice de implementare
- *Activity diagram:*
- folosită pentru a modela procese, fie de afaceri, fie software
  - evidențiază acțiunile necesare pentru a completa o anumită activitate sau process
  - include mecanisme pentru fluxul de control și fluxul de date pentru a coordona acțiunile din cadrul procesului
- *Sequence diagram:*
- descrie interacțiunile între obiecte pentru a îndeplini o sarcină specifică
  - se focusează pe ordinea cronologică a mesajelor schimbate între obiecte
- *State machine diagram:*
- reprezintă schimbările de stare ale obiectelor în timpul ciclului lor de viață
  - arată stările posibile ale obiectului și tranzițiile între acestea, declanșate de evenimente

6. Folosind Visual Paradigm, realizați diagramele de clasă din Figurile 1 și 2, prezentate în laborator.

➤ Figura 1:



➤ Figura 2:

