

Laborator 1

1. Ce este UML și pentru ce este utilizat?

Unified Modeling Language (UML) este o consolidare a celor mai bune Unified Modeling practici care au fost stabilite de-a lungul anilor în utilizarea modelului de limbaj (UML). UML ne permite să prezentăm aspectele foarte variate ale unui sistem software (de exemplu, cerințe, structuri de date, fluxuri de date, și fluxuri de informații) într-un singur cadru folosind orientarea pe obiecte.

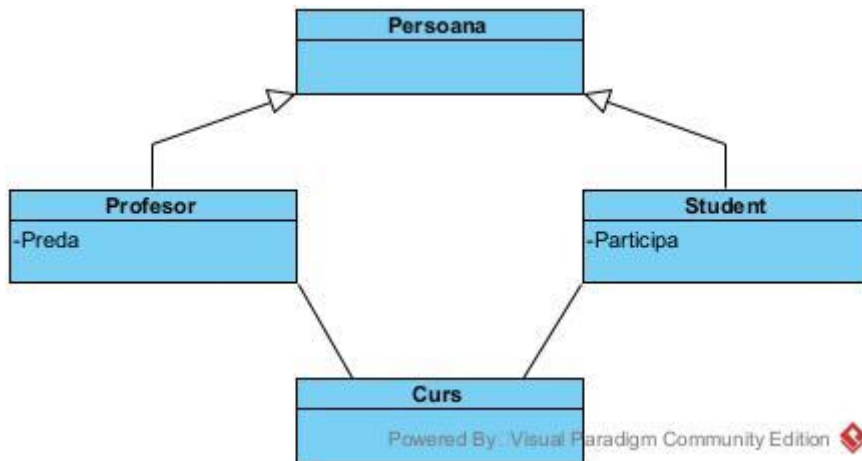
2. Ce sunt modelele și care este utilitatea lor?

Derivat din latinescul „modulus”, care desemnează o scară în arhitectură, cuvântul „modello” a fost folosit în Italia pentru un obiect ilustrativ destinat să prezinte unui client forma și designul unei clădiri planificate și pentru a clarifica problemele de proiectare și arhitectură. Pe parcursul secolului următor, noțiunea de „model” a fost folosită în diverse ramuri ale științei pentru o descriere simplificată a faptelor complexe din realitate.

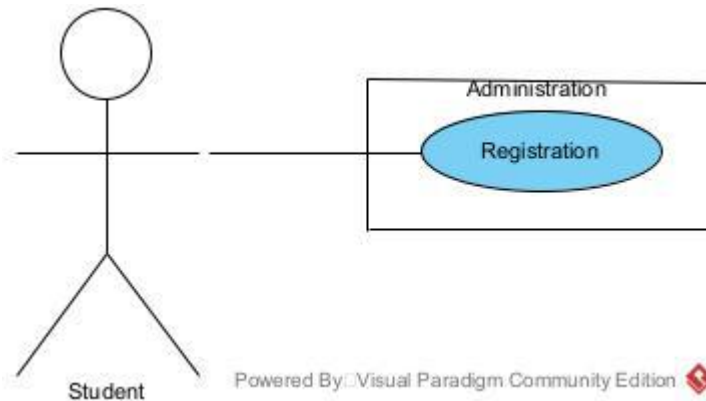
Modelele ne permit să descriem sistemele eficient și elegant. Modelele suportă o reprezentare a unui sistem care se reduce la esențiale pentru a minimiza complexitatea sistemului la aspecte gestionabile.

3. 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:

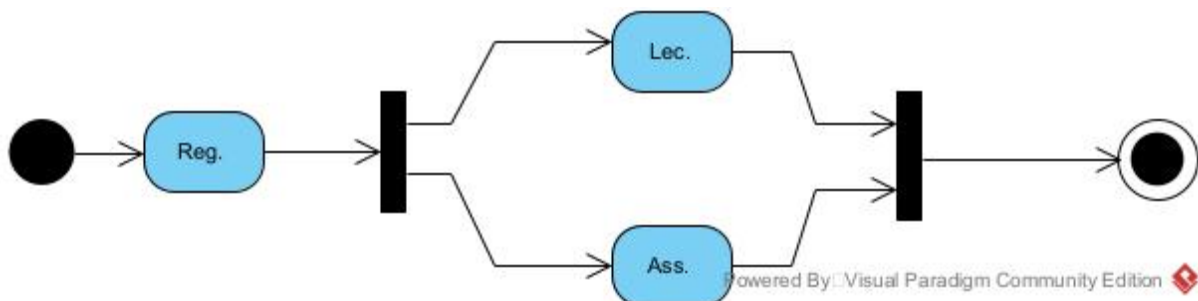
- Class diagram – diagrama de clasă



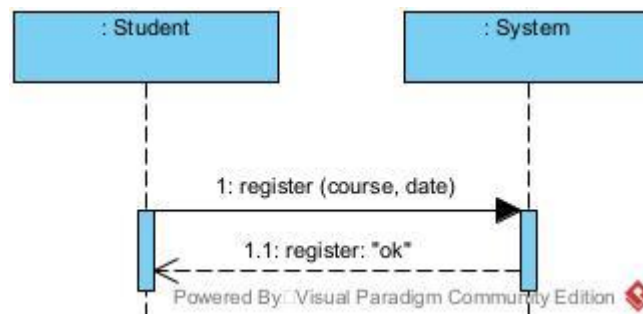
- Use case diagram – diagrama de cazuri de utilizare



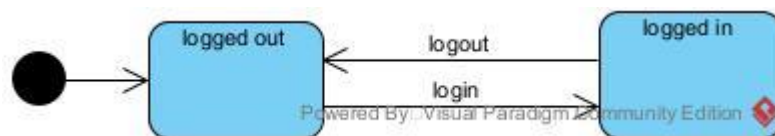
- Activity diagram – diagrama de activitate



- Sequence diagram – diagrama de secvențe



- State machine diagram – diagrama mașinii de stare



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

Sunt prezentate următoarele tipuri de diagrame: Structure Diagrams și Behavior Diagrams.

Diagramele de structura se concentrează pe elementele statice în timp ce cele de comportament se concentrează pe cele dinamice. Cele de structura ilustrează

organizarea internă a sistemului în timp ce cele de comportament ilustrează ce face și cum reacționează sistemul. Atât diagramele de structură cât și diagramele de comportament fac parte din Unified Modeling Language (UML) și ambele tipuri de diagrame servesc ca documentație pentru un sistem software.

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

Diagrama class:

Reprezintă clase și relațiile dintre ele. Specifică numărul de instanțe dintr-o clasă care pot fi asociate cu instanțe din alte clase.

Diagrama use case:

Reprezintă actori care sunt utilizatori externi care interacționează cu sistemul. Arată cazuri de utilizare, adică secvențele de acțiuni pe care actorii le pot efectua pentru a obține un anumit rezultat. Se utilizează linii pentru a conecta actori cu cazuri de utilizare, indicând care actori pot efectua ce cazuri de utilizare.

Diagrama activity:

O diagramă de activitate poate arăta pași necesari unui student pentru a participa la o prelegere și la o temă. Diagramele de activitate oferă mecanisme pentru a reprezenta atât fluxul de control, cât și fluxul de date din cadrul unui proces. Aceste mecanisme ajută la coordonarea acțiunilor care alcătuiesc activitatea generală.

Diagrama sequence:

Diagramele de secvență prezintă interacțiunile dintre obiecte pentru a finaliza o sarcină specifică. Accentul se pune pe ordinea cronologică a mesajelor trimise între obiecte. Diagramele de secvență oferă instrumente pentru a modela interacțiuni complexe prin controlul ordinii mesajelor și prin utilizarea modularizării.

Diagrama state-machine:

Obiectele trec prin diferite stări pe parcursul duratei lor de viață. Evenimentele, cum ar fi conectarea și deconectarea, declanșează tranziții între stări. Diagramele state-machine ne ajută să vizualizăm aceste stări și tranziții, oferind o imagine de ansamblu asupra comportamentului posibil al unui obiect.

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

