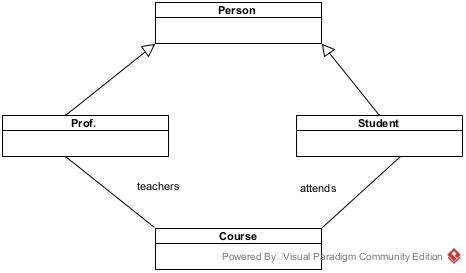
TEMA 1

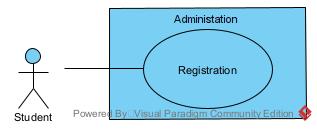
* Ce este UML si pentru ce este utilizat?
* UML, prescurtare de la Unified Modeling Language, este un limbaj standard utilizat pentru a descrie modele si specificatii pentru software.expand\_more. A fost creat de catre consortiul Object Management Group (OMG) cu scopul de a standardiza diversele sisteme de notare si abordari pentru proiectarea software-ului.expand\_more
* UML este un limbaj vizual, ceea ce inseamna ca foloseste simboluri grafice pentru a reprezenta concepte precum clase, obiecte, relatii si procese. Acest lucru il face un instrument util pentru comunicarea designului software intre diferiti stakeholders, cum ar fi dezvoltatori, designeri si clienti.
* UML este utilizat in principal pentru proiectarea software orientat pe obiecte, dar poate fi folosit si pentru alte tipuri de sisteme, cum ar fi sisteme de business sau sisteme hardware.expand\_more.
* Ce sunt modelele si care este utilitatea lor?
* Modelele sunt reprezentari simplificate ale unor obiecte, sisteme sau concepte din lumea reala. Ele ne ajuta sa intelegem, comunicam si analizam aceste lucruri intr-un mod mai usor si mai clar.
* Familiarizarea cu mediul de lucru mai sus amintit prin crearea anumitor diagrame din sectiunile 2.3.1 s, i 2.3.2 din [1].

Mai exact:

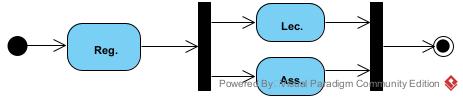
-Class Diagram



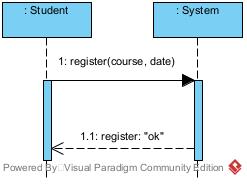
-Use Case Diagram



-Activity Diagram



-Sequence diagram



-State Machine Diagram



* Ce tipuri de diagrame sunt prezentate in [1] – Capitolul 2? Care sunte asemanarile si deosebirile intre acestea?
* In aceste capitole sunt prezentate Structure Diagrams(diagramele de structura) si Behavior Diagrams(diagramele de comportament).
* Asemanari:
* Ambele tipuri de diagrame fac parte din UML, atat diagramele de structura, cat si diagramele de comportament sunt elemente importante ale UML.
* Ambele tipuri de diagrame pot fi utilizate pentru a reprezenta diferite aspecte ale unui sistem software.
* Ambele tipuri de diagrame pot fi utilizate pentru a imbunatati comunicarea si colaborarea dintre echipele de dezvoltare: Diagramele ofera o reprezentare vizuala a designului software-ului, care poate fi usor inteleasa de catre toti membrii echipei.
* Deosebiri:
* Tipul de informatii reprezentate:

Diagrame de structura: Reprezinta structura statica a unui sistem software, incluzand clase, obiecte, relatii si atribute.

Diagrame de comportament: Reprezinta comportamentul dinamic al unui sistem software, incluzand interactiuni, fluxuri de control si stari.

* Scopul principal:

Diagrame de structura: Documentarea structurii interne a software-ului, facilitand intelegerea modului in care sunt organizate componentele sale.

Diagrame de comportament: Descrierea modului in care se comporta sistemul software in diferite scenarii, permitand identificarea si rezolvarea problemelor potentiale.

* Tipuri de diagrame:

Diagrame de structura: Exemple includ diagrame de clase, diagrame de obiecte, diagrame de componente si diagrame de implementare.

Diagrame de comportament: Exemple includ diagrame de cazuri de utilizare, diagrame de secventa, diagrame de activitate, diagrame de stare si diagrame de interactiune.

* Caracterizati cele 5 tipuri de diagrame de mai sus.

1. **Diagrama de clase:**

* Reprezinta structura statica a sistemului.
* Identifica clasele, atributele si relatiile dintre ele.
* Permite intelegerea organizarii interne a software-ului.

1. **Diagrama de secventa:**

* Descrie interactiunea dintre obiecte in timp.
* Ilustreaza ordinea in care se transmit mesajele intre obiecte.
* Folosita pentru a analiza fluxul de control al sistemului.

1. **Diagrama de cazuri de utilizare:**

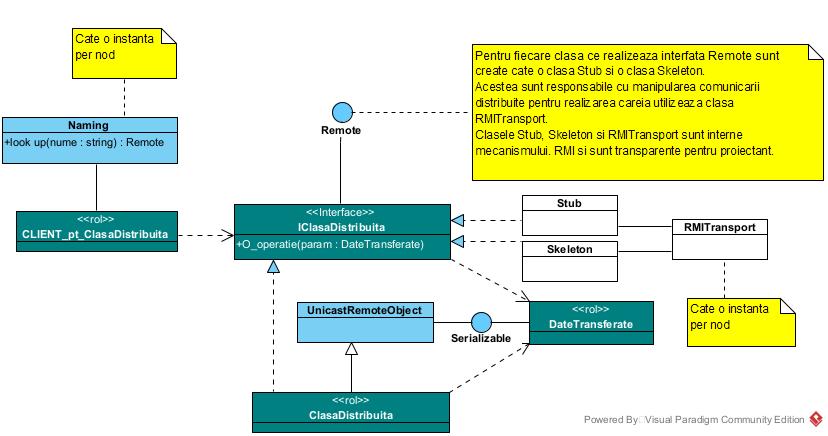
* Prezinta functionalitatea sistemului din perspectiva utilizatorilor.
* Identifica actorii (utilizatori si alte sisteme) si cazurile de utilizare (actiuni pe care le pot realiza).
* Permite intelegerea modului in care utilizatorii interactioneaza cu sistemul.

1. **Diagrama de activitate:**

* Reprezinta fluxul de lucru al unui proces sau caz de utilizare.
* Identifica etapele si actiunile implicate in realizarea unui proces.
* Folosita pentru a modela logica de afaceri a sistemului.

1. **Diagrama de stare:**

* Descrie comportamentul dinamic al unui obiect.
* Identifica starile posibile ale obiectului si tranzitiile dintre ele.
* Folosita pentru a modela modul in care un obiect reactioneaza la stimuli externi.



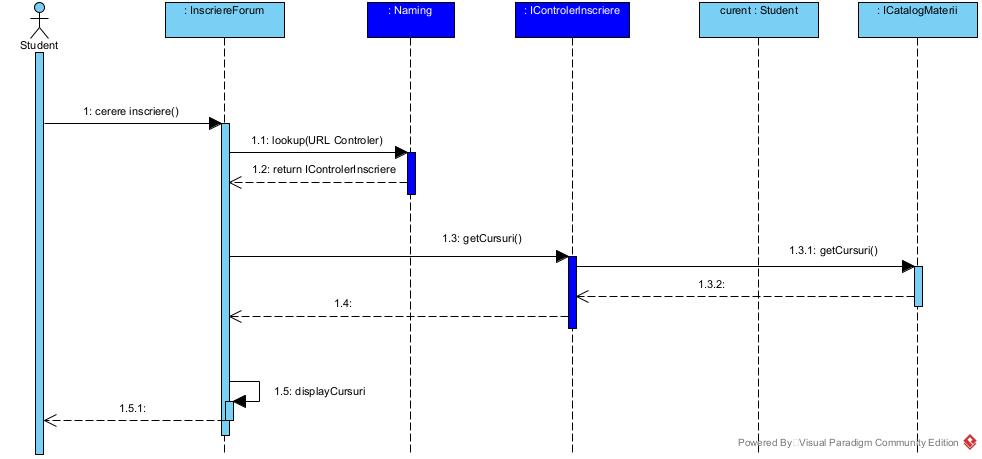
 Figura 1: O diagrama de clase

Figura 2: O diagrama de secvente