1. 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. UML permite prezentarea aspectelor variate ale unui sistem software (cerințele funcționale și nefuncționale, structurile de date, fluxurile de date și fluxurile de informații) într-un cadru unic, utilizând concepte orientate pe obiecte.

1. Ce sunt modelele și care este utilitatea lor?

Modelele ne permit să descriem sistemele eficient și elegant. Noțiunea de „model” a fost folosită în diverse ramuri ale științei pentru o descriere simplificată a faptelor complexe din realitate. Modelele suportă o reprezentare a unui sistem care se reduce la

esențiale pentru a minimiza complexitatea sistemului de a aspecte de vârstă. Un sistem este de obicei descris nu printr-o singură vedere ci printr-o serie de opinii care împreună produc o imagine de ansamblu unificată.

Modelele pot fi folosite în diverse scopuri. Astfel distingem modelele descriptive de cele prescriptive. Modelele descriptive arată o parte din realitate pentru a face un anumit aspect mai ușor de înțeles. De exemplu, o hartă a orașului descrie un oraș în așa fel încât să ajute un turist să găsească rute în interiorul orașului. În schimb, modelele prescriptive sunt folosite pentru a oferi un manual de construcție pentru sistemul de dezvoltat.

Modelele sunt folosite ca o schiță pentru a comunica anumite aspecte ale unui sistem într-un mod simplu. Aici, modelul nu este o mapare completă a sistemul. Schițele se disting de fapt prin selectivitatea lor, așa cum se reduc la aspectele esenţiale pentru rezolvarea unei probleme. Schițele fac adesea vizibile soluțiile alternative. Astfel, modelele sunt folosite și ca bază de discuție.

4. Familiarizarea cu mediul de lucru mai sus amintit prin crearea anumitor diagrame din

sectiunile 2.3.1 și 2.3.2 din [1] (class diagram – diagrama de clasă, use case diagram

– diagrama de cazuri de utilizare, activity diagram – diagrama de activitate, sequence

diagram – diagrama de secvente, state machine diagram – diagrama mașinii de stare).

Class diagram: A diagram of a person's course

Description automatically generated

Use case diagram: A diagram of a student and a student

Description automatically generated

Activity diagram: A diagram of a flowchart

Description automatically generated

Sequence diagram: A diagram of a student

Description automatically generated

State machine diagram: A diagram of a login

Description automatically generated

5. Ce tipuri de diagrame sunt prezentate în [1] – Capitolul 2? Care sunt asemănarile și

deosebirile?

In capitolul 2 sunt prezentate: diagrama de cazuri de utilizare, diagrama mașinii de stare, diagrama de activitate, diagrama de secvente, diagrama de comunicare, diagrama de timp si diagrama de prezentare a interactiunii. Toate acestea se aseamana prin faptul ca fac parte din categoria diagrame UML, sunt folosite pentru analiza si proiectare si prin utilizarea componentelor UML. Ceea ce le diferentiaza este rolul fiecareia, diagrama de clase specifica structurile de date si de obiecte, diagrama de obiecte reprezinta un instantaneu concret al starii sistemului, diagrama de pachete este ffolosita pentru organizarea modelelor complexe, diagrama structurii compozitiei permite o descompunere ierarhica a componentelor, diagrama de implementare reprezinta topologia hardware, diagrama de profil permite extinderea UML, diagrama de componente modeleaza componente independente si executabile diagrama de cazuri de utilizare are rolul de a specifica cerintele unui sistem, diagrama mașinii de stare reprezinta comportamentul permis al unui obiect, diagarama de activitate modeleaza procese de orice tip, diagrama de secventa descrie interactiunile dintre obiecte pentru a indeplini o anumita sarcina. Diagrama de comunicare pune accent pe relatiile de comunicare, diagrama de timp arata schimbarile de stare, iar diagrama de prezentare a interactiunii modeleaza conexiunile dintre diferite procesee de interactiune.

6. Caracterizați succint fiecare diagramele din [1] – Capitolul 2 (cele 5 menționate mai

sus)

**Diagrama de clase** specifica structurile de date si de obiecte ale unui system, **diagrama de cazuri de utilizare** are rolul de a specifica cerintele unui sistem, **diagrama mașinii de stare** reprezinta comportamentul permis al unui obiect, **diagarama de activitate** modeleaza procese de orice tip, **diagrama de secventa** descrie interactiunile dintre obiecte pentru a indeplini o anumita sarcina.

7. Folosind Visual Paradigm, realizați diagramele de clasă:

A diagram of a program

Description automatically generated

A diagram of a computer

Description automatically generated