

## Tema 1

- Ce este UML și pentru ce este utilizat?

UML, abreviere pentru Unified Modeling Language, reprezintă o sinteză a celor mai bune practici dezvoltate în decursul anilor în utilizarea limbajelor de modelare în ingineria software. Prin intermediul UML, avem posibilitatea de a prezenta diverse aspecte ale unui sistem software - cum ar fi cerințele, structurile de date, fluxurile de date și informații - într-un singur cadru, utilizând concepte orientate pe obiecte.

- Ce sunt modelele și care este utilitatea lor?

Un model evidențiază aspectele importante ale software-ului într-o formă clară de notare, care este cât mai simplă posibil, dar face abstracție de la irelevante detalii, la fel ca modelele din arhitectură, de exemplu, planurile de construcție.

Modelele pot fi utilizate într-o varietate de scopuri, diferențiem între modelele descriptive și cele prescriptive. Modelele descriptive sunt concepute pentru a ilustra o parte din realitate, facilitând înțelegerea anumitor aspecte. De exemplu, un model descriptiv al traficului urban poate arăta modul în care vehiculele circulă într-un oraș. Pe de altă parte, modelele prescriptive sunt utilizate pentru a oferi instrucțiuni sau ghiduri specifice pentru construcție sau dezvoltare. De exemplu, un model prescriptiv poate furniza un manual de construcție detaliat pentru dezvoltarea unui sistem software sau pentru a găsi rute în interiorul unui oraș pentru un utilizator non-local.

- Ce tipuri de diagrame sunt prezentate în [1] - Capitolul 2? Care sunt asemanarile și deosebirile dintre acestea?

Tipurile de diagrame prezentate în [1] sunt diagramele de structura și cele de comportament. **Asemanari:** Ambele sunt parte a limbajului de modelare UML și sunt folosite pentru proiectarea și modelarea sistemelor software. **Deosebiri:** Diagramele de structură se concentrează pe structura statică a sistemului, în timp ce diagramele de comportament se concentrează pe interacțiunea și comportamentul dinamic al sistemului. Diagramele de structură oferă o perspectivă statică, în timp ce diagramele de comportament oferă o perspectivă dinamică a sistemului.

- Caracterizați succint fiecare diagramă prezentată în [1] - Capitolul 2.

### ***Diagrama de clasă (Class diagram):***

Describe structura statică a sistemului, evidențiind clasele din sistem și relațiile între ele. Folosită pentru a modela entitățile din sistem, atributele și metodele lor, precum și relațiile dintre acestea.

### ***Diagrama de cazuri de utilizare (Use case diagram):***

Evidențiază interacțiunile dintre sistem și actorii săi (utilizatori sau alte sisteme externe). Folosită pentru a descrie funcționalitățile sistemului din perspectiva utilizatorilor săi, arătând scenariile de utilizare posibile.

***Diagrama de activități (Activity diagram):***

Modelează fluxurile de lucru sau procesele din sistem, evidențiind activitățile și fluxurile de control între ele. Folosită pentru a reprezenta comportamentul procedural sau detaliile algoritmice ale sistemului.

***Diagrama de secvență (Sequence diagram):***

Arată interacțiunile dintre obiectele din sistem într-un anumit scenariu de utilizare sau într-o secvență specifică de evenimente. Folosită pentru a evidenția ordinea și mesajele trimise între obiectele participante într-o secvență de acțiuni.

***Diagrama de automat finit de stare (State machine diagram):***

Modelează stările și tranzițiile posibile ale unui obiect sau a unui sistem în timpul ciclului său de viață. Folosită pentru a descrie comportamentul dinamic al sistemului și modul în care acesta răspunde la evenimente și schimbări de stări.