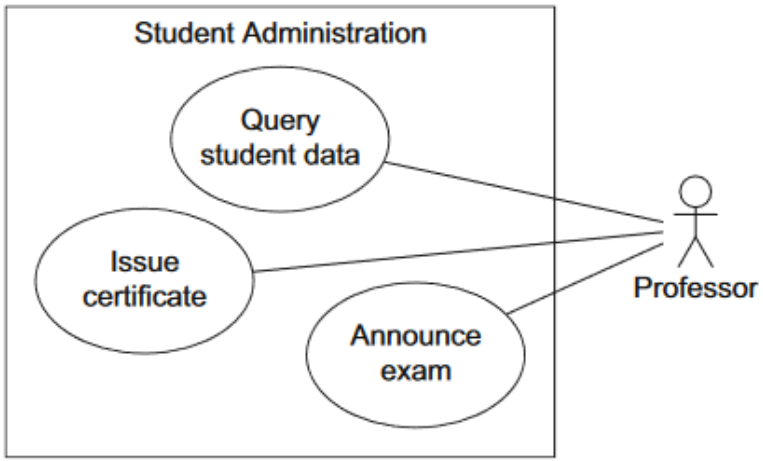
Unified Modeling Language (UML)

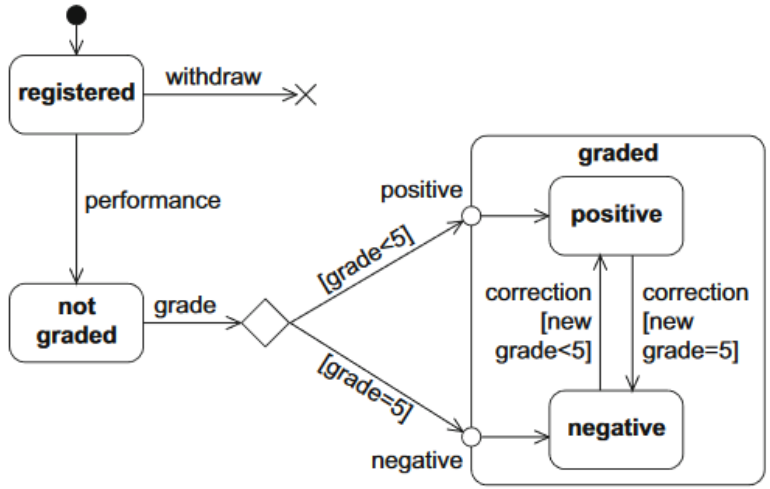
autoři: Bc. Štěpán Bezděk, Bc. Daniel Zahálka
předmět: 18OOP
fakulta: Fakulta jaderná a fyzikálně inženýrská
České vysoké učení technické v Praze

Co je to UML?

- standardizovaný vizuální modelovací jazyk
- způsob vizualizace návrhu systému

- UML metamodel - abstraktní model, který definuje strukturu a pravidla pro vytváření modelů v jazyce UML





Využití UML

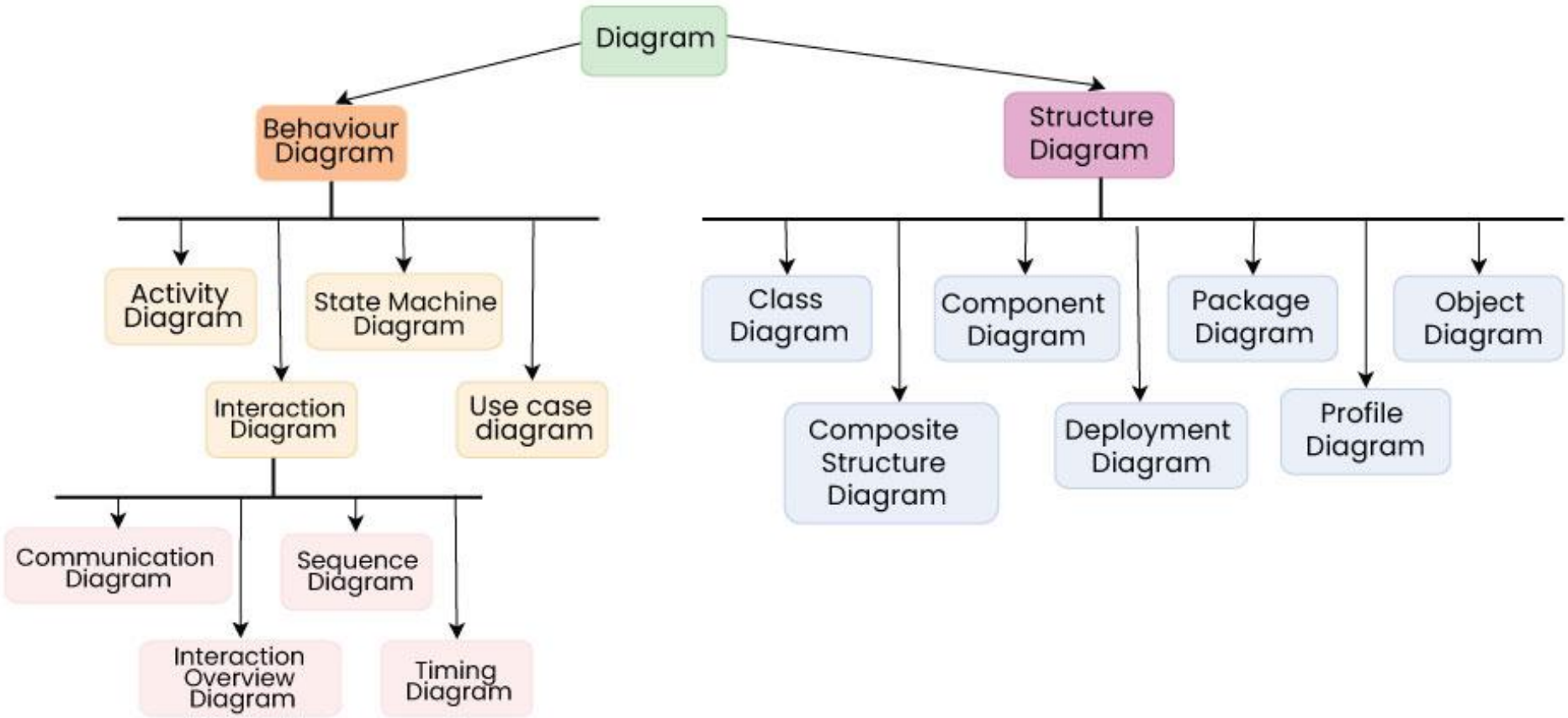
- UML není vázán na konkrétní vývojový nástroj, konkrétní programovací jazyk nebo konkrétní cílovou platformu, na které musí být vyvíjený systém používán

- jde o jasný a stručný způsob komunikace mezi týmy lidí
- nezbytný pro komunikaci s neprogramátory o základních požadavcích, funkcích a procesech systému
- úspora času při správné vizualizaci procesů

Diagram

- v UML je model znázorněn graficky ve formě **diagramů**
  - poskytují pohled na část reality popsanou modelem
  - vyjadřují, kteří uživatelé používají kterou funkci
  - znázorňují strukturu systému, ale bez specifikace konkrétní implementce
  - představují podporované a zakázané procesy
- v aktuální verzi 2.5.1 UML nabízí 14 standardizovaných diagramů
- popisují buď strukturu, nebo chování systému
- popsány jsou podrobně ve specifikaci: `https://www.omg.org/spec/UML/2.5.1/PDF`

Typy UML diagramů



UML Diagrams

Strukturální diagramy

- zaměřují se na statickou strukturu systému
- představují **prvky systému** (např. třídy, objekty, komponenty)
- jejich **vztahy**
  - bez ohledu na časové aspekty
- jejich **vlastnosti**

- UML nabízí **sedm typů** diagramů pro modelování struktury systému z různých pohledů:
- *Class diagram* – Diagram tříd
- *Object diagram* – Diagram objektů
- *Component diagram* – Diagram komponent
- *Composite structure diagram* – Diagram složené struktury
- *Deployment diagram* – Diagram nasazení
- *Package diagram* – Diagram balíčků
- *Profile diagram* – Profilový diagram

Diagramy chování

- zaměřují se na dynamické aspekty systému
- popisují chování systému, jeho prvků a jejich interakce v průběhu času
- modelují tok činností, změny stavů a interakce mezi objekty a komponentami

- UML nabízí také **sedm typů** diagramů chování v různých úrovních systému:
- *Use case diagram* – Diagram případů užití
- *Activity diagram* – Diagram aktivit
- *State machine diagram* – Stavový diagram
- *Sequence diagram* – Sekvenční diagram
- *Communication diagram* – Komunikační diagram
- *Interaction overview diagram* – Přehledový diagram interakcí
- *Timing diagram* – Časový diagram

Diagram tříd

- Nejrozšířenějším diagramem
- Notace:
  - Název třídy
  - Atributy – představují datové členy třídy
  - Metody – představují chování nebo funkčnost třídy
- Přístupnost
  - $+$ veřejné – viditelné pro všechny třídy
  - $-$ privátní – viditelné pouze v rámci třídy
  - $\#$ chráněné – viditelné pro podtřídy
  - $\sim$ viditelné ve stejném balíčku

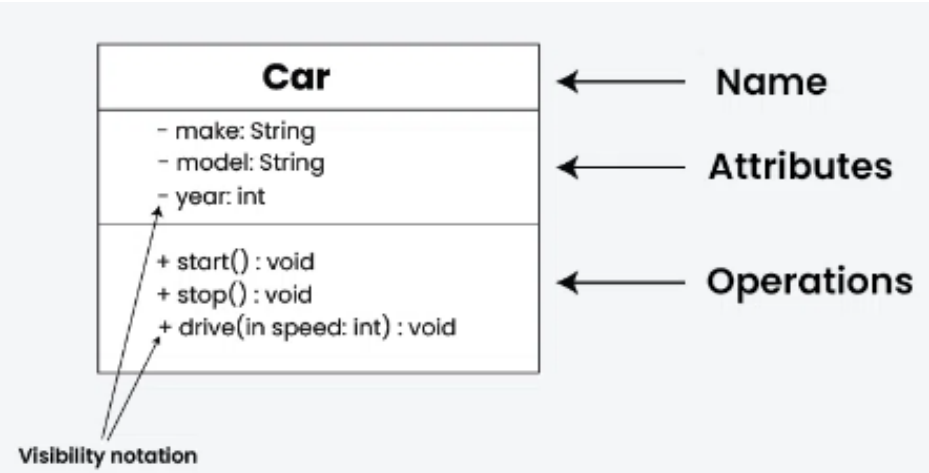


Diagram tříd

- Vztahy mezi třídami
  - Asociace – objekty třídy mohou být spojeny s objekty druhé
  - Navigovatelná asociace – specializovaný typ asociace
  - Dědičnost – třída dědí vlastnosti a metody jiné třídy
  - Implementace/Realizace – třída implementuje rozhraní nebo abstraktní třídu
  - Závislost – změna jedné třídy může ovlivnit druhou
  - Agregace – reprezentuje celek a jeho části
  - Složení – podobná agregaci, ale části nemohou existovat mimo celek

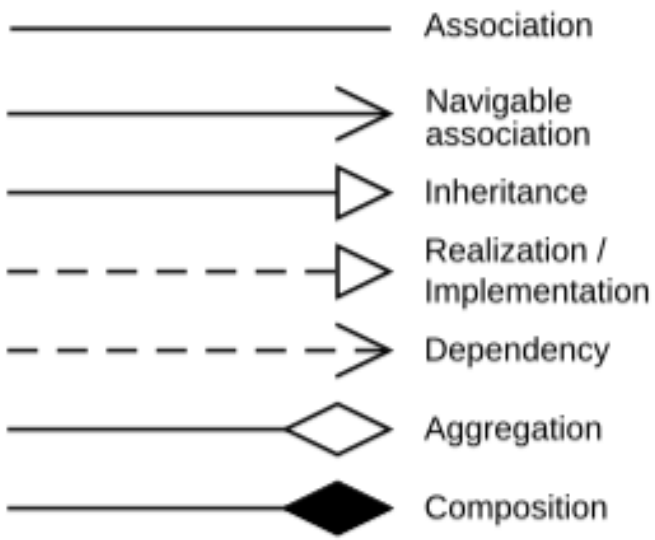


Diagram objektů

- Snímek obrazovky instancí v systému a vztahu, který mezi nimi existuje
- Zápis:
  - Specifikace objektu – Atributy a hodnoty
  - Odkaz – vyjádření vztahu mezi dvěma objekty
  - Závislostní vztahy – ukazuje, kdy jeden prvek závisí na jiném prvku
  - Asociace – jeden objekt odkazuje na členy druhého objektu
  - Agregace – specifická forma sdružování
  - Složení

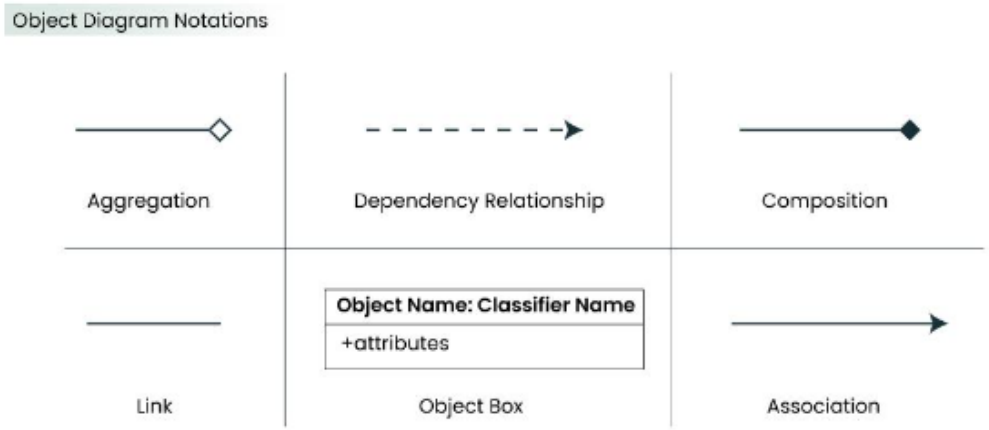


Diagram případů užití

- Představuje interakci mezi aktéry (uživateli nebo externími systémy) a zvažovaným systémem za účelem dosažení konkrétních cílů.
- Prvky diagramu
  - Agent/Aktér – externí entity, které interagují se systémem; mohou zahrnovat uživatele, jiné systémy nebo hardwarová zařízení
  - Případ užití – konkrétní věci, které systém umí
  - Hranice systému – definuje, co je uvnitř systému a co je vně

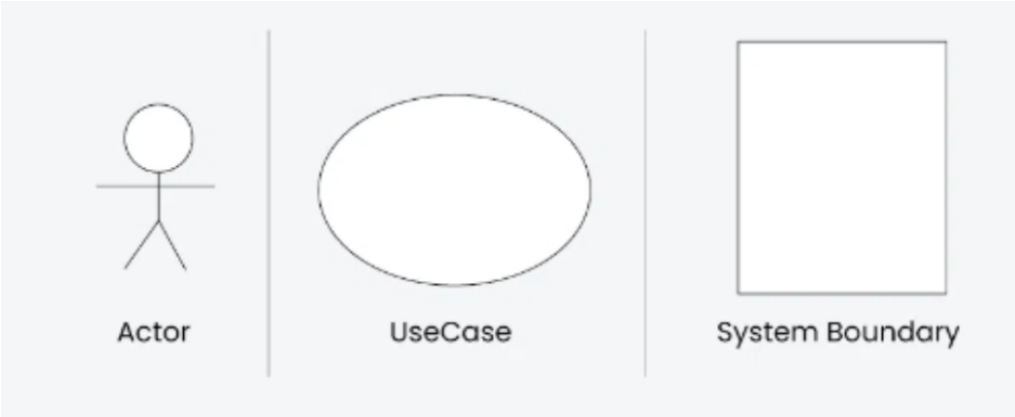
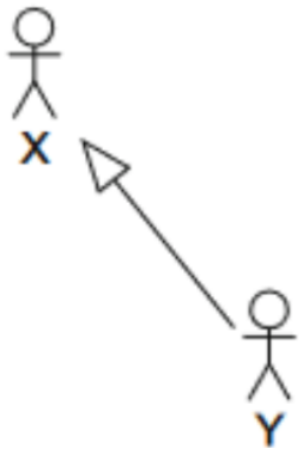


Diagram případů užití

- Vztahy mezi aktéry:
  - Aktéři mají často společné vlastnosti a některé případy užití mohou být použity různými aktéry.

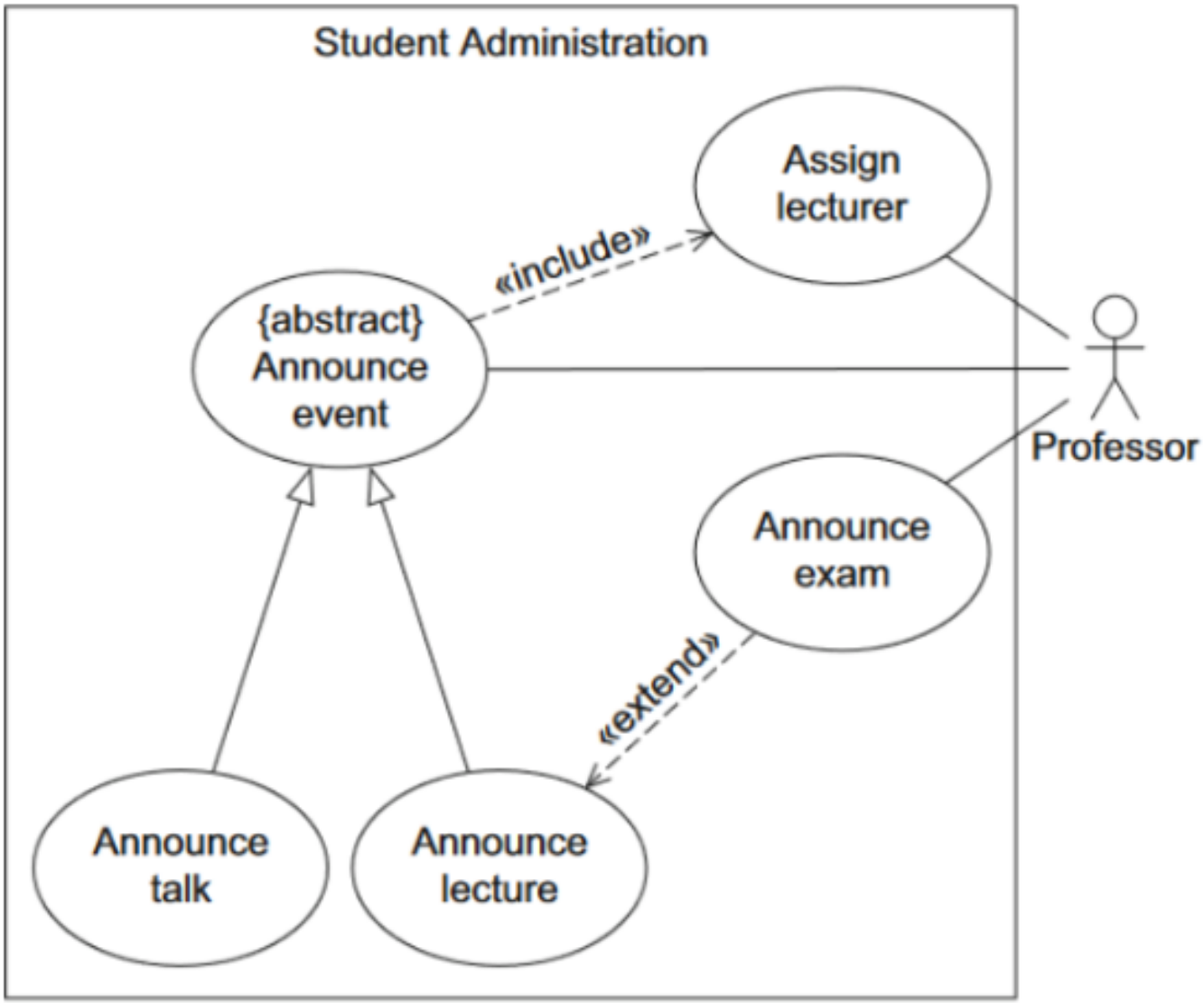


- Vztahy mezi prvky
  - Asociativní vztah – představuje komunikaci nebo interakci mezi aktérem a případem užití
  - Include – označuje, že případ užití zahrnuje funkce jiného případu užití
  - Rozšíření – ilustruje, že případ užití lze za určitých podmínek rozšířit o další případ užití
  - Zobecnění – společné vlastnosti a společné chování různých případů užití lze seskupit do nadřazeného případu užití

Diagram případů užití

- Postup návrhu diagramu:
  - Chcete-li identifikovat aktéry, kteří se objevují v diagramu případů užití, musíte odpovědět na následující otázky:
    - Kdo používá hlavní případy užití?
    - Kdo potřebuje podporu pro svou každodenní práci?
    - Kdo je zodpovědný za správu systému?
    - Jaká jsou externí zařízení/(softwarové) systémy, se kterými musí systém komunikovat?
    - Kdo má zájem na výsledcích systému?
  - Jakmile znáte účastníky, můžete odvodit případy užití tak, že se zeptáte na následující otázky o účastnících:
    - Jaké jsou hlavní úkoly, které musí aktér vykonávat?
    - Chce se aktér dotazovat nebo dokonce měnit informace obsažené v systému?
    - Chce aktér informovat systém o změnách v jiných systémech?
    - Má být aktér informován o neočekávaných událostech v systému?

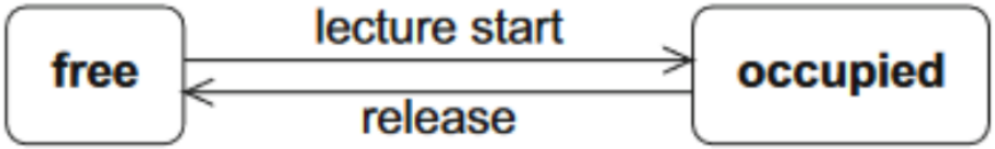
Diagram případů užití



Stavový diagram

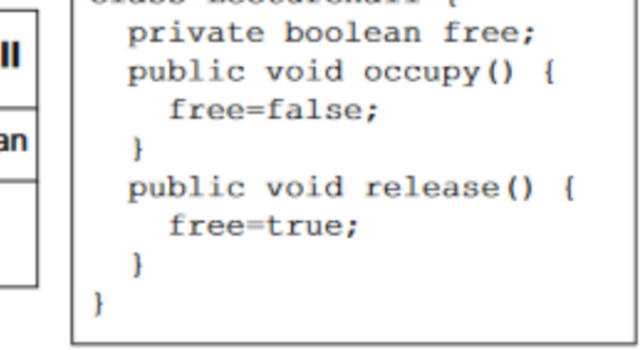
- Stavový diagram se používá k modelování dynamického chování třídy v reakci na čas a měnící se vnější podněty

- jednoduchý příklad: posluchárna, může být v jednom ze dvou stavů: volná nebo obsazená.



Stavový diagram

- Stejně jako každý jiný diagram i stavový diagram modeluje pouze tu část systému, která je pro daný účel nezbytná nebo relevantní.
- Pokud se však již nacházíte v pozdní fázi vývojového procesu, je výhodná reprezentace blízká kódu.

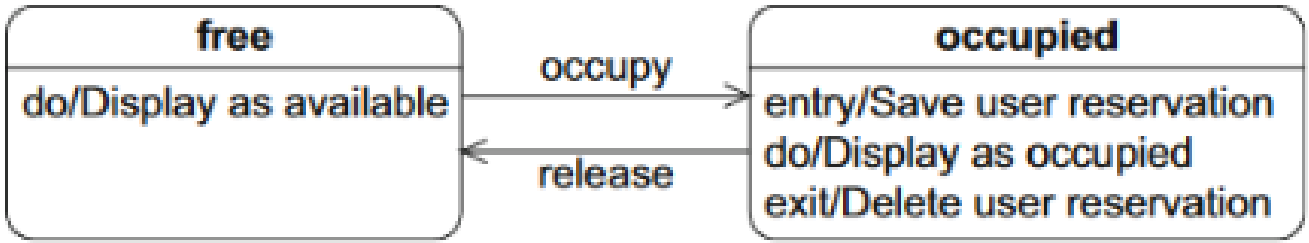



```
class LectureHall {
  private boolean free;
  public void occupy() {
    free=false;
  }
  public void release() {
    free=true;
  }
}
```

Stavový diagram

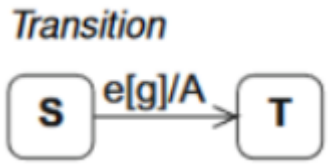
- Stavový diagram je graf se stavy jako uzly a přechody mezi stavy jako hrany.
- V diagramu je stav zobrazen jako obdélník se zaoblenými rohy a je označen názvem stavu.

- Když je objekt v určitém stavu, může tento objekt provádět všechny interní činnosti uvedené v tomto stavu.



Stavový diagram

- Přechod z jednoho stavu do druhého se označuje jako změna stavu nebo jednoduše přechod.



- Událost (event) (nazývanou také „spouštěč“), spustí přechod stavu
- Podmínka (guard) (nazývaná také „strážní podmínka“), umožňuje provedení přechodu
- Činnosti (actions) (nazývané také „účinky“ nebo „efekty“), se provedou během změny cílového stavu

- Typy stavů:
  - výchozí stav
  - rozhodovací uzel
  - konečný stav

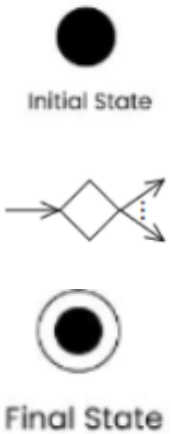


Diagram aktivit

- Používají se k ilustraci toku řízení v systému a odkazují na kroky spojené s prováděním případu užití.

- Uživatelsky definované chování ve formě aktivit.
  - Může popisovat implementaci případu užití.
  - Obsahem aktivity je orientovaný graf.
  - Uzly představují komponenty aktivity:
    - akce, datová úložiště a řídicí prvky
  - Hrany představují tok řízení nebo tok objektů.
    - možné cesty provádění aktivity

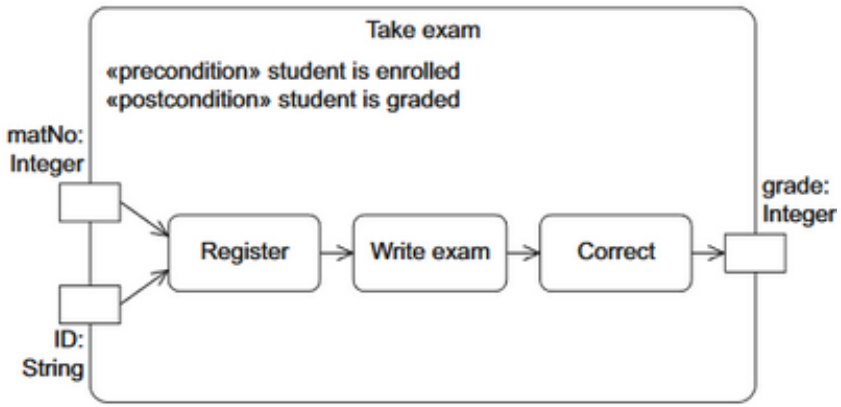


Diagram aktivit

- Akce
  - Základními prvky aktivit.
  - Zadání libovolného uživatelského chování.
- Akce zpracovávají vstupní hodnoty a vytvářejí výstupní hodnoty.
  - schopny provádět výpočty
  - načítat data z paměti
  - měnit aktuální stav systému
- Akce založené na událostech (Event-based akce)
- čekání na událost (a), čekání na čas (b)

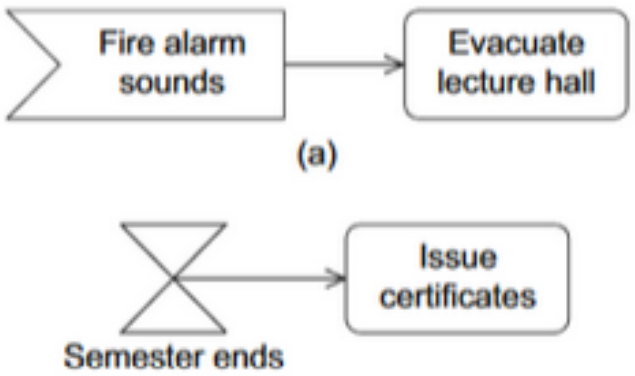
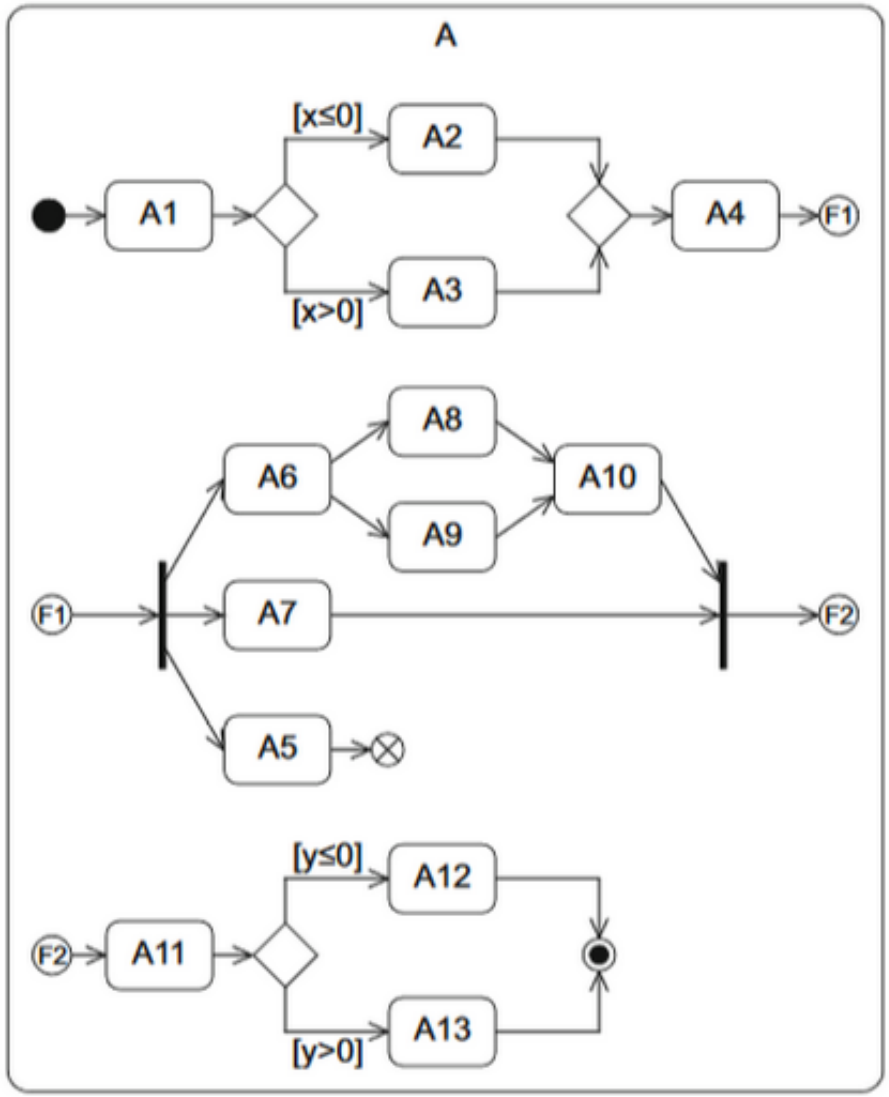
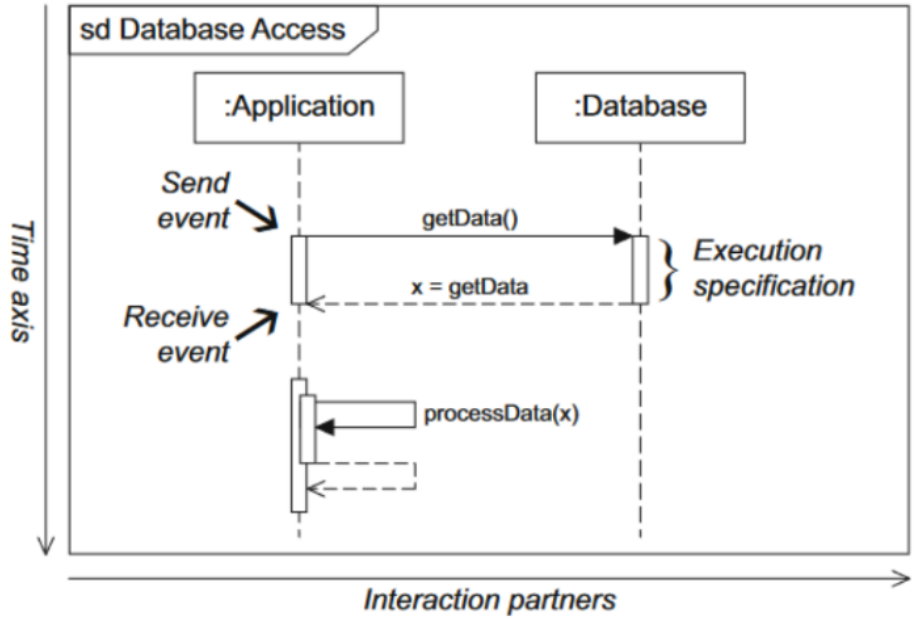


Diagram aktivit



Sekvenční diagram

- Znázorňuje interakci mezi objekty v sekvenčním pořadí, tj. pořadí, ve kterém se tyto interakce odehrávají.
- Prvky diagramu:
  - Partneři interakce/Aktéři – třída nebo instance třídy provádějící komunikaci / osoba nebo stroj provádějící komunikaci
  - Lifeline(doba existence) – popisuje dobu fungování aktéra
  - Zpráva – Asynchronní/Synchronní/Odpověď



Vytvoření UML

1. Identifikujte účel
2. Identifikujte prvky a vztahy
3. Vyberte vhodný typ diagramu UML
4. Vytvořte hrubý náčrt
5. Vyberte modelovací nástroj UML
6. Vytvořte diagram
7. Definujte vlastnosti prvku
8. Přidejte poznámky a komentáře
9. Ověřte a zkontrolujte
10. Upřesněte a opakujte

Nástroje pro tvorbu UML diagramů

- Lucidchart – webový nástroj; umožňuje více uživatelům pracovat na diagramech v reálném čase
- Draw.io – bezplatný webový nástroj; Integruje se s různými službami cloudového úložiště a lze jej používat offline
- Visual Paradigm – poskytuje komplexní sadu nástrojů pro vývoj softwaru, včetně diagramů UML
- StarUML – open-source modelovací nástroj UML
- Papyrus – open-source modelovací nástroj UML
- PlantUML – textový nástroj, který vám umožňuje vytvářet diagramy UML pomocí jednoduché a člověkem čitelné syntaxe

Děkujeme za pozornost

Zdroje

- SEIDL, Martina; SCHOLZ, Marion; HUEMER, Christian a KAPPEL, Gerti, 2015. UML @ Classroom. Springer Cham. ISBN 978-3-319-12742-2. Dostupné také z: https://doi.org/10.1007/978-3-319-12742-2.
- GeeksforGeeks, 2024. Unified Modeling Language (UML) Diagrams. Dostupné z: https://www.geeksforgeeks.org/unified-modeling-language-uml-introduction/
- GeeksforGeeks, 2024. Class Diagram — Unified Modeling Language (UML). Dostupné z: https://www.geeksforgeeks.org/unified-modeling-language-uml-class-diagrams/
- Visual Paradigm, 2024. What is Class Diagram? Dostupné z: https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-class-diagram/
- Visual Paradigm, 2024. What is Composite Structure Diagram? Dostupné z: https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-composite-structure-diagram/
- Visual Paradigm, 2024. What is Object Diagram? Dostupné z: https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-object-diagram/
- Visual Paradigm, 2024. What is Profile Diagram? Dostupné z: https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-profile-diagram/