51. UML – charakteristika jazyka a rozšiřitelnost

- UML (Unified Modeling Language) – nástroj a jazyk pro specifikování požadavků, analýzu, vizualizaci, návrh a modelování software a různých systémů a také obchodních procesů (viz UP)
- z čeho se skládá:
  - prvky pro modelování
  - vztahy – vazby prvků
  - diagramy – pohledy do modelu – nad jedním modelem může být více pohledů/diagramů
- mechanismy UML:
  - specifikace – grafické a textové prvky
  - doplňky
  - obecná dělení – rozhraní vs. implementace – co prvek dělá vs. jak to dělá
  - mechanismy rozšiřitelnosti – jak lze UML rozšiřovat
    - omezení – podmínka nebo pravidlo vztahující se k objektu
      - $\{x == 1\}$
    - stereotyp – nové, uživatelem definované prvky odvozené ze stávajících UML prvků
      - << subsystem >>
    - připojené hodnoty – definice nových vlastností prvků a zavedení klíčových slov pro ně
      - keyword = value

- pohled UML na architekturu – chápe ji jako strukturu jednotlivých částí a jejich propojení a interakci a hlavní principy návrhu systému

=> 4+1 pohled na architekturu:

- Logický – slovník domény – tříd objektů atd.
  - diagram tříd, objektů, balíčků, stavového automatu
- Procesní – procesy jako aktivní třídy (: instance tříd)
  - diagram aktivit, sekvencí, komunikace
- Implementační – soubory tvořící kód systému a jejich závislosti
  - diagram balíčků a komponent
- Nasazení – nasazení SW na HW
  - diagram nasazení
- Případů použití – požadavky na systém a jeho používání
  - diagram případů použití (Use Case Diagram)

MODELOVÁNÍ STRUKTURY
- diagram tříd
- objektů
- komponent
- složené struktury
- balíčků
- nasazení (SW na HW)

MODELOVÁNÍ CHOVÁNÍ
- diagram případů užití
- diagramy interakce tříd mezi sebou
  - sekvence
  - komunikace
  - stavový
  - aktivity

MODELOVÁNÍ STRUKTURY

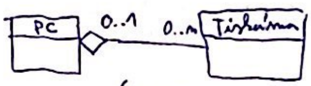
1.) Diagram tříd

- třídy (množiny objektů stejného typu/stejné třídy se stejnými atributy a metodami (operacemi) a jejich struktura a závislosti a vztahy mezi nimi
- asociace jsou tzv. vazby a vztahy
- může být jednosměrná nebo obousměrná (—— nebo ——>)

→ statický pohled na strukturu tříd

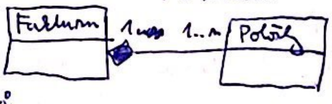
- generalizace × specializace —▷
- agregace × kompozice

⇒

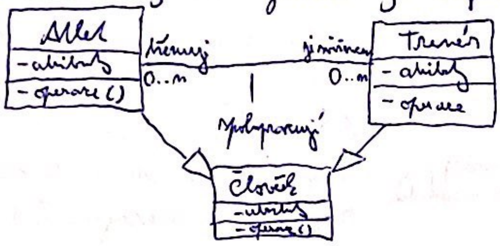


↳ PC může existovat bez tiskárny

- v asociacích je uvedena kardinalita na každé straně a její název a také mohou být role jednotlivých objektů:



→ faktura nemůže existovat bez položek

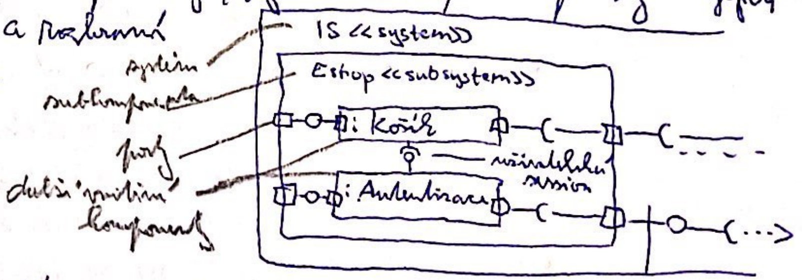


2.) Diagram objektů (snímek)

- vychází z diagramu tříd ale ukazuje objekty (konkrétní instance tříd) a jejich vztahy v nějakém časovém okamžiku (snapshot)

3.) Diagram komponent

- implementační diagram
- ukazuje rozložení systému na komponenty, jejich další podkomponenty a jejich vzájemné vztahy, závislosti a rozhraní
- je zapouzdřena vnitřní struktura komponent – ta se neřeší

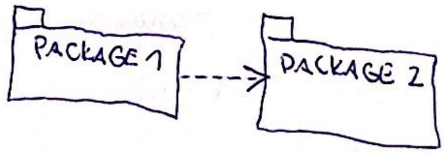


4.) Diagram složené struktury

- z čeho se systém skládá a jaké jsou vazby – třeba výherní bankomat má hlavní procesor, displej, paměť, klávesnici, modem, šifrovací procesor, tiskárnu, čtečku karet, kameru, ...)

5.) Diagram balíčků

- modeluje balíčky a jejich závislosti



6.) Diagram nasazení – implementační diagram

- nasazení SW na HW
- jaké záznamy budou v DB, co bude za soubory, kde budou fyzicky soubory jaké knihovny, jaký konkrétní HW bude použit atd...
- třeba i co bude instalováno na nějakém konkrétním serveru (BSD Solaris) a co naopak pouze na PC uživatele – klientská část

MODELOVÁNÍ CHOVÁNÍ

1. Diagram případů užití (Use Case Diagram)

- základní model pro analýzu požadavků
- může obsahovat aktéry a případy užití a jejich propojení (Případ UC)
- představuje požadavky na systém a co má systém umět a co mohou jednotliví uživatelé dělat
- aktéři mohou mít specializaci x generalizaci – jeden vychází z druhého (mohou to mít i případy užití ale nepoužívá se)
- vazby «include» - - -> – případ užití zahrnuje druhý
  - «extend» <- - - - – může se provést i ten druhý v toku (to jsou tzv. body pro rozšíření)

Specifikace případů užití
- textová specifikace konkrétního případu užití (název UC, popis, aktéři, precondition, postcondition, tok řízení, ...)

2. Diagram sekvence

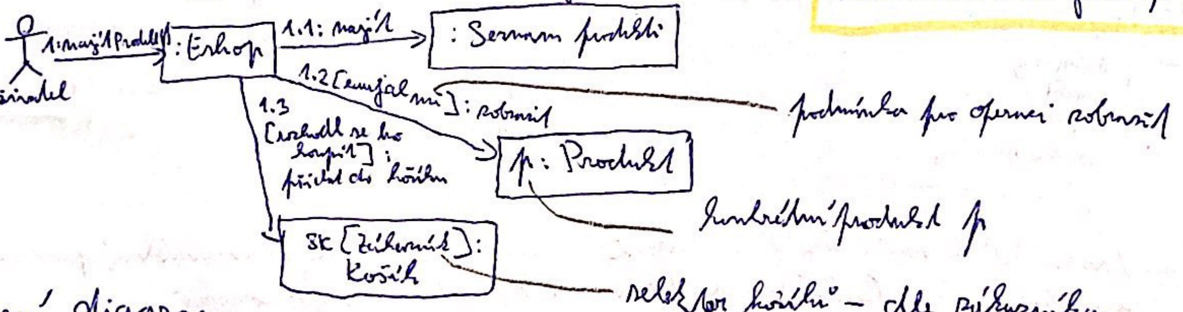
- instance tříd – konkrétní objekty [:A]
- čára života ⋮ a aktivní doba [ ]
- posílání zpráv mezi objekty – typy zpráv:
- může posílat zprávu i sám sobě
- v rozšířeném jsou cykly LOOP a podmínky OPT a ALT

- - - -> [:A] = konstruktor
——> = synchronní volání
<- - - - - = callback, vrací hodnotu atd..
——> = asynchronní volání

3. Diagram komunikace – zase objekty [:A]

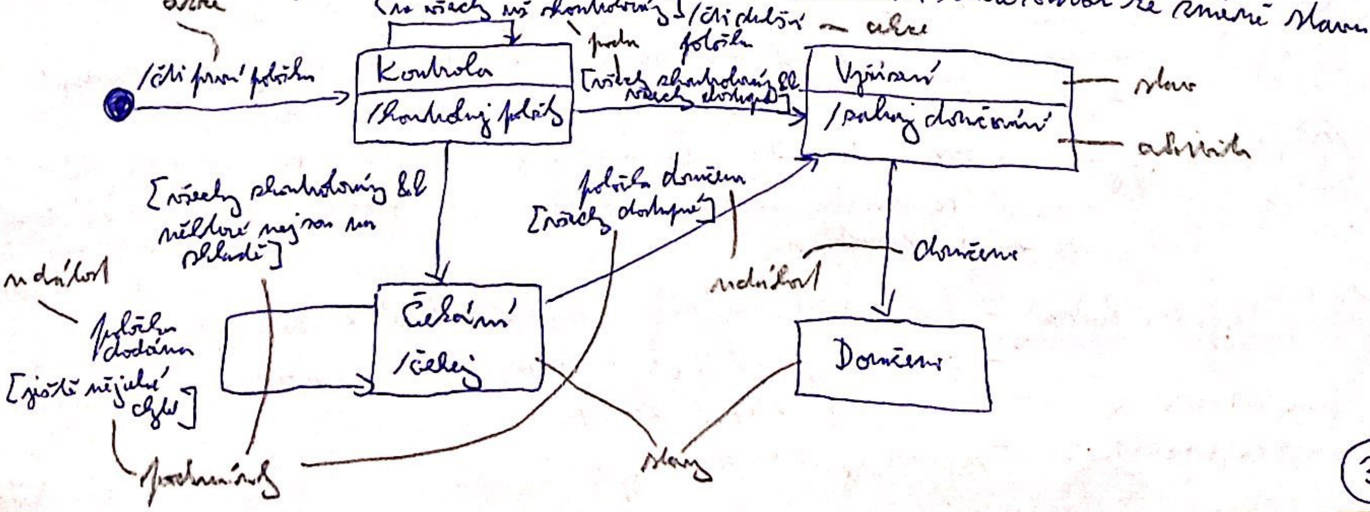
- statické vazby mezi objekty které se využívají pro komunikaci
- potenciální vazby pro komunikaci se číslují

– říká se mu i diagram spolupráce



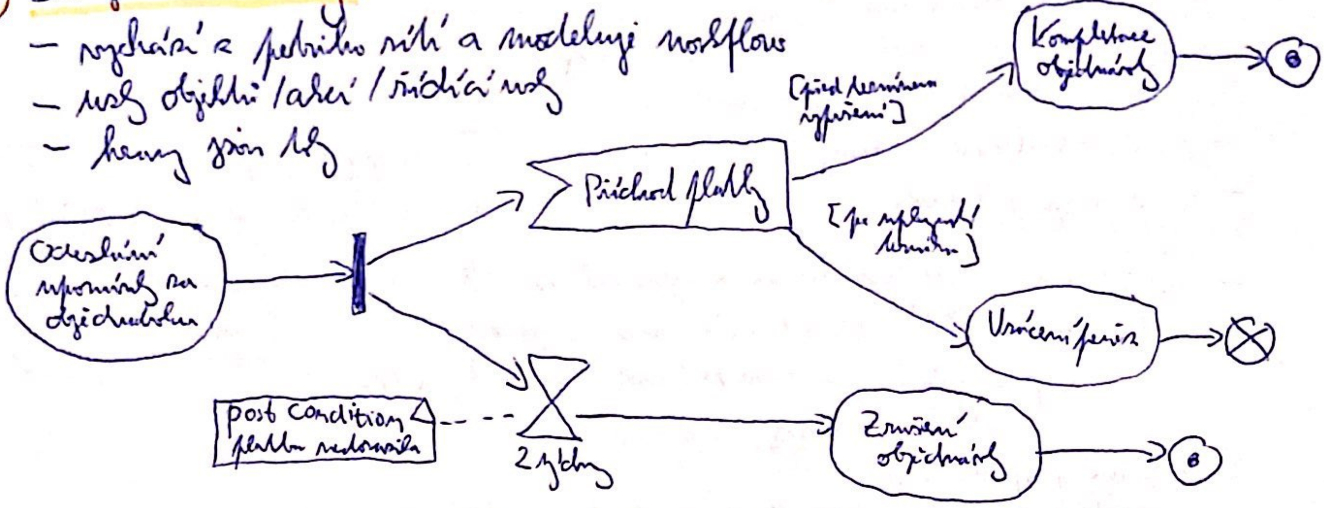
4. Stavový diagram

- vychází z konečného automatu a reprezentuje stavy systému a jak se lze dostat z jednoho stavu do druhého – události které vedou ke změně stavu

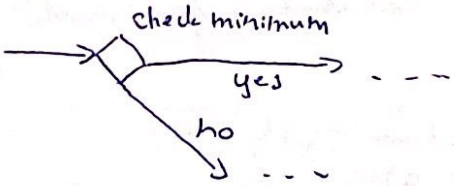


5. Diagram aktivit

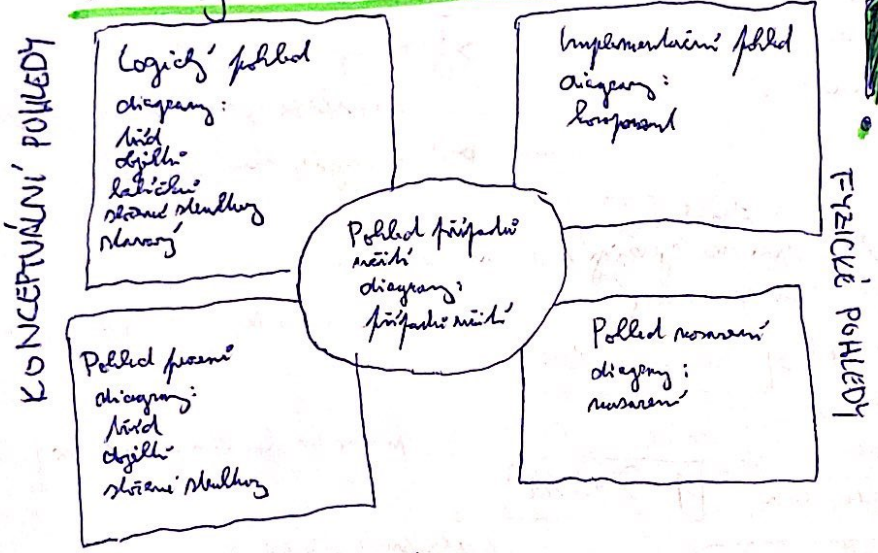
- vychází z Petriho sítí a modelují workflow
- tok objektů / akcí / řídící tok
- hrany jsou toky



- podmínky mohou být i



Pohledy UML na architekturu



! máme model nějakého SW nebo systému nebo architektury a diagramy jsou pohledy na ten model a ty pohledy se dělí na logické, implementační, nasazení, procesní a případů užití

⟹ 4+1 pohled

- tvůrci UML věděli že UML nemůže obsáhnout všechny univerzální prostředky pro modelování jakéhokoliv systému a jeho potřeb a požadavků

⟹ proto lze jazyk UML rozšiřovat pomocí prostředků rozšiřitelnosti

1. **Omezení** {...}
   - textový popis omezení, dodržení nějaké podmínky, nějaké pravidlo které musí být dodrženo, nějaký předpoklad
   - jsou nějaká předdefinovaná omezení v UML ale lze vytvářet vlastní
   - lze použít jazyk omezení ale není to vyjadřováno striktně

2. **Stereotypy** <<...>>
   - je to odvození nového prvku ze stávajícího prvku UML – jsou i nějaké předdefinované stereotypy <<subsystem>> pro komponenty atd..
   - vezme se původní prvek a to že je to nový a odvozený se značí stereotypem

3. **Připojené hodnoty**
   keyword=value
   {autor = ZK, verze = 1.0}
   - prvky v UML mají nějaké vlastnosti které se zobrazují v diagramu
   - pomocí připojených hodnot lze zavést nové vlastnosti pro prvky nebo tak jak se používají klíčová slova pro ně
   - uplatňují se u stereotypů