PB007 Softwarové inženýrství I

Cvičenie 12 – Package, Component, Deployment Diagrams

Valdemar Švábenský

Fakulta informatiky, Masarykova univerzita, Brno

8. decembra 2015

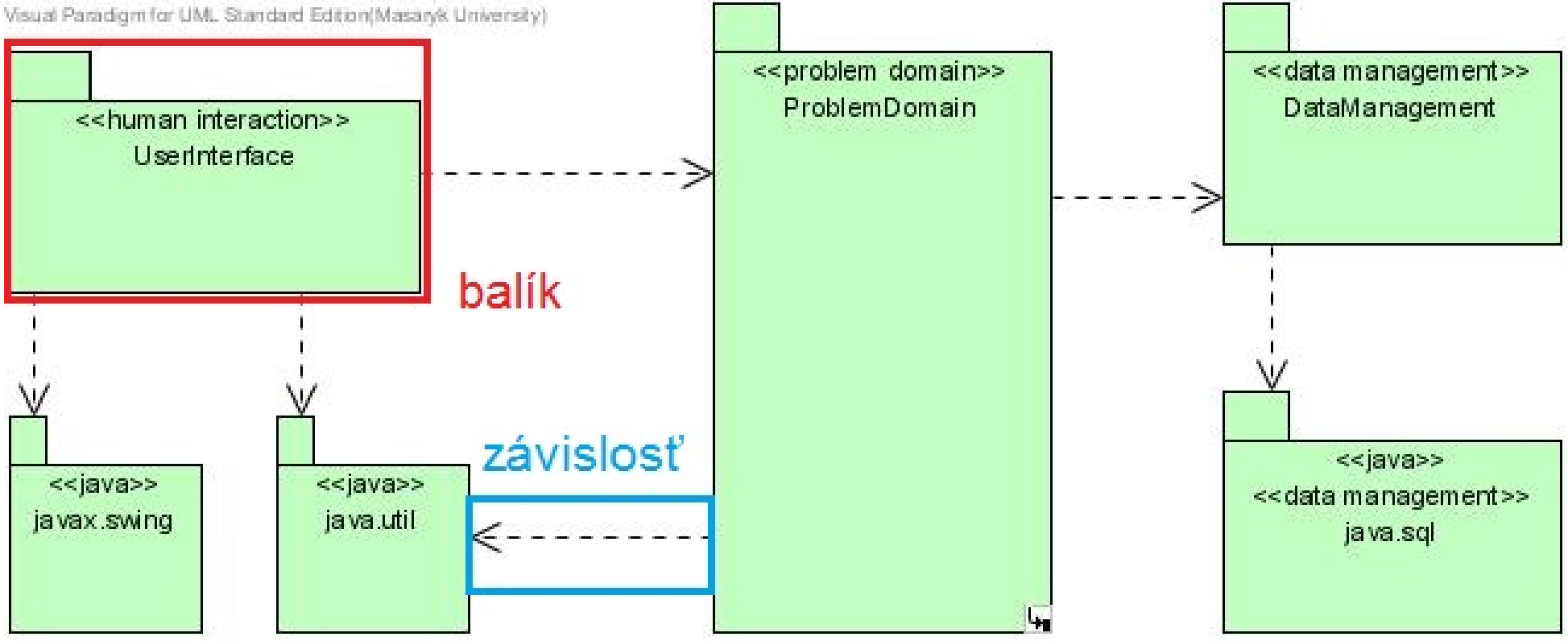
Obsah

1. Package Diagram (Diagram balíkov)
2. Component Diagram (Diagram komponentov)
3. Deployment Diagram (Diagram nasadenia)
4. Úlohy

Diagram balíkov

- Návrhy veľkých systémov obsahujú stovky tried
- Diagram balíkov logicky združuje prvky rôznych diagramov do skupín a popisuje závislosti medzi nimi
- Analógia s balíkom (package) v Jave
- Prvky diagramu:
  - **Balík** (Package) – mechanizmus pre združovanie sémanticky súvisiacich prvkov v diagramoch
  - **Závislosť** (Dependency) – relácia medzi balíkmi, kedy je klientský balík nejako závislý na zdrojovom balíku
    - «use» – Klientský balík využíva prvky zdrojového balíku (východzí typ závislosti)
    - «import» – Verejné prvky zdrojového balíka sú pridané ako verejné prvky do klientského balíku
    - «access» – Verejné prvky zdrojového balíka sú pridané ako súkromné prvky do klientského balíku

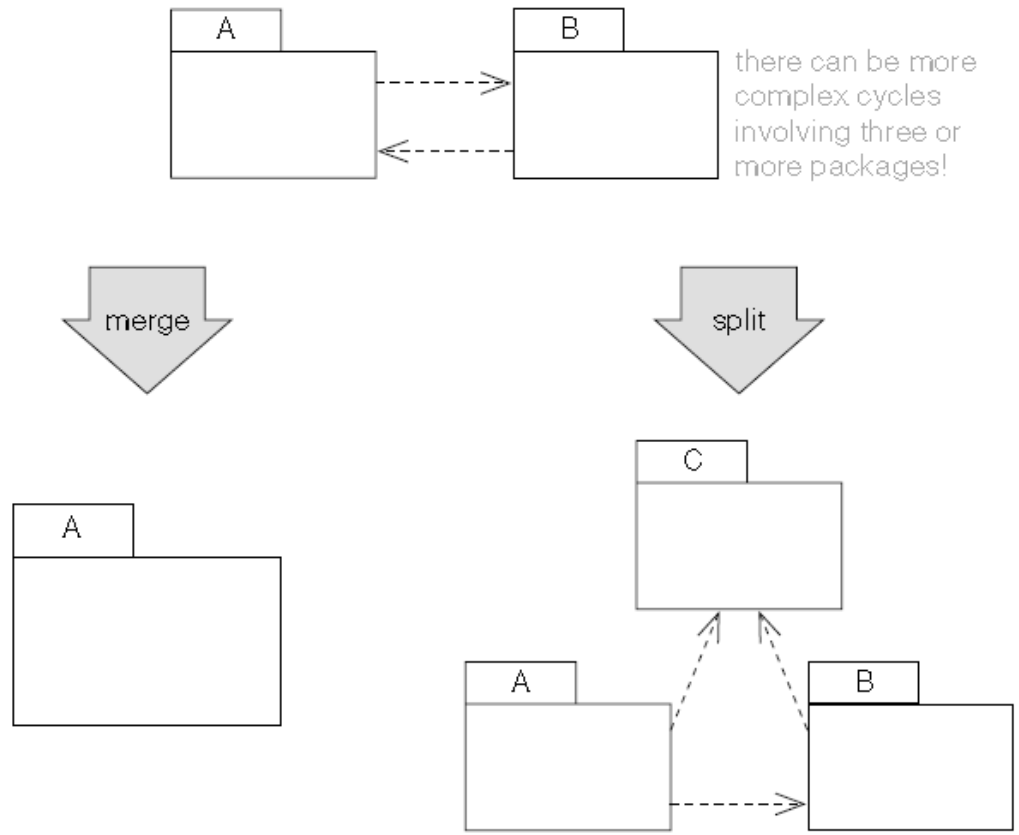
Diagram balíkov – ukážka



Zdroj: https://is.muni.cz/auth/el/1433/podzim2015/PB007/um/sem/cz_files/11/11_StudiumPackage.jpg

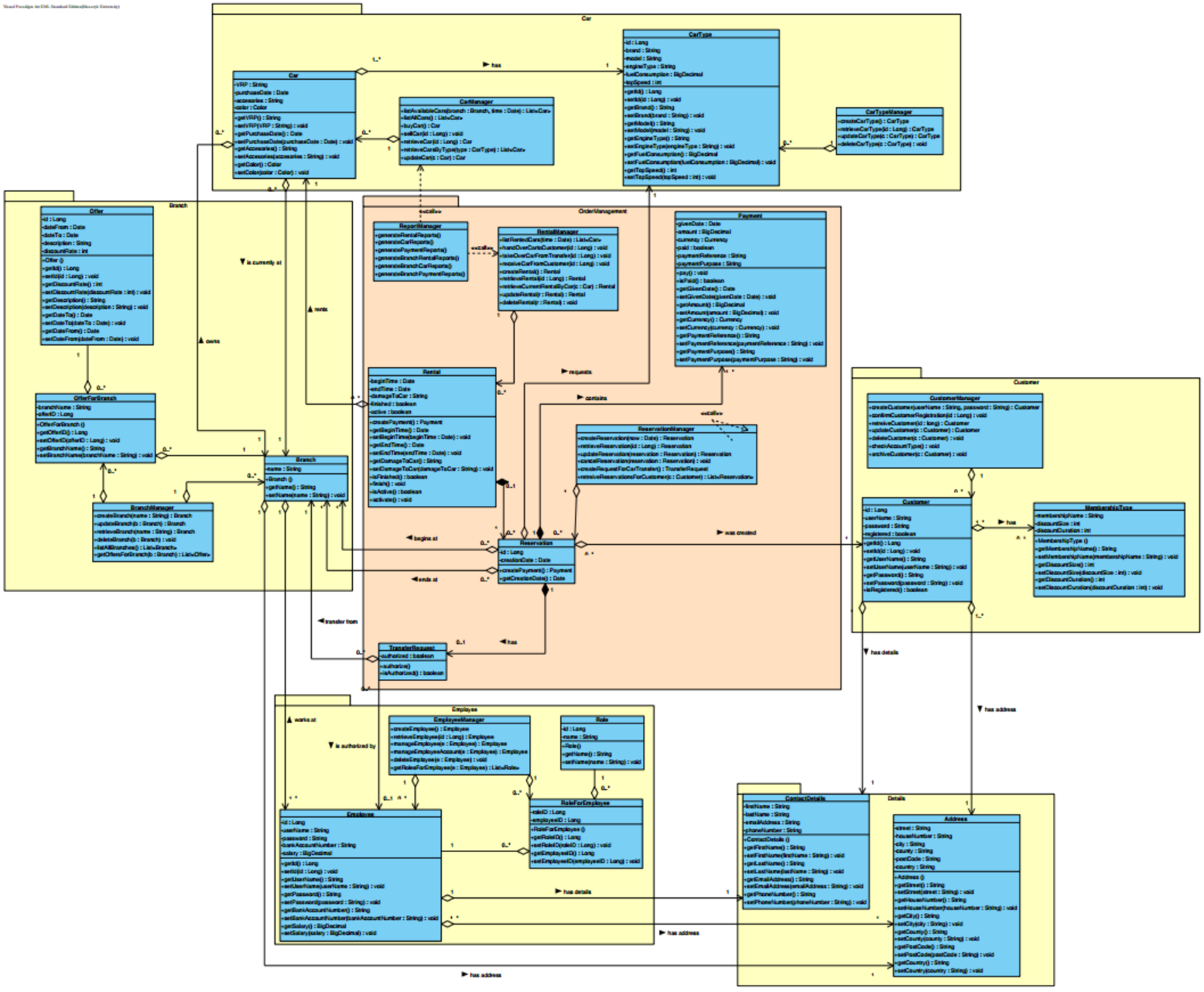
Diagram balíkov – závislosti

- Závislosť medzi balíkmi $\Longleftrightarrow$ existuje závislosť medzi dvoma prvkami týchto balíkov
- Závislosť je tranzitívna
- Nesmie dochádzať k cyklickým závislostiam:

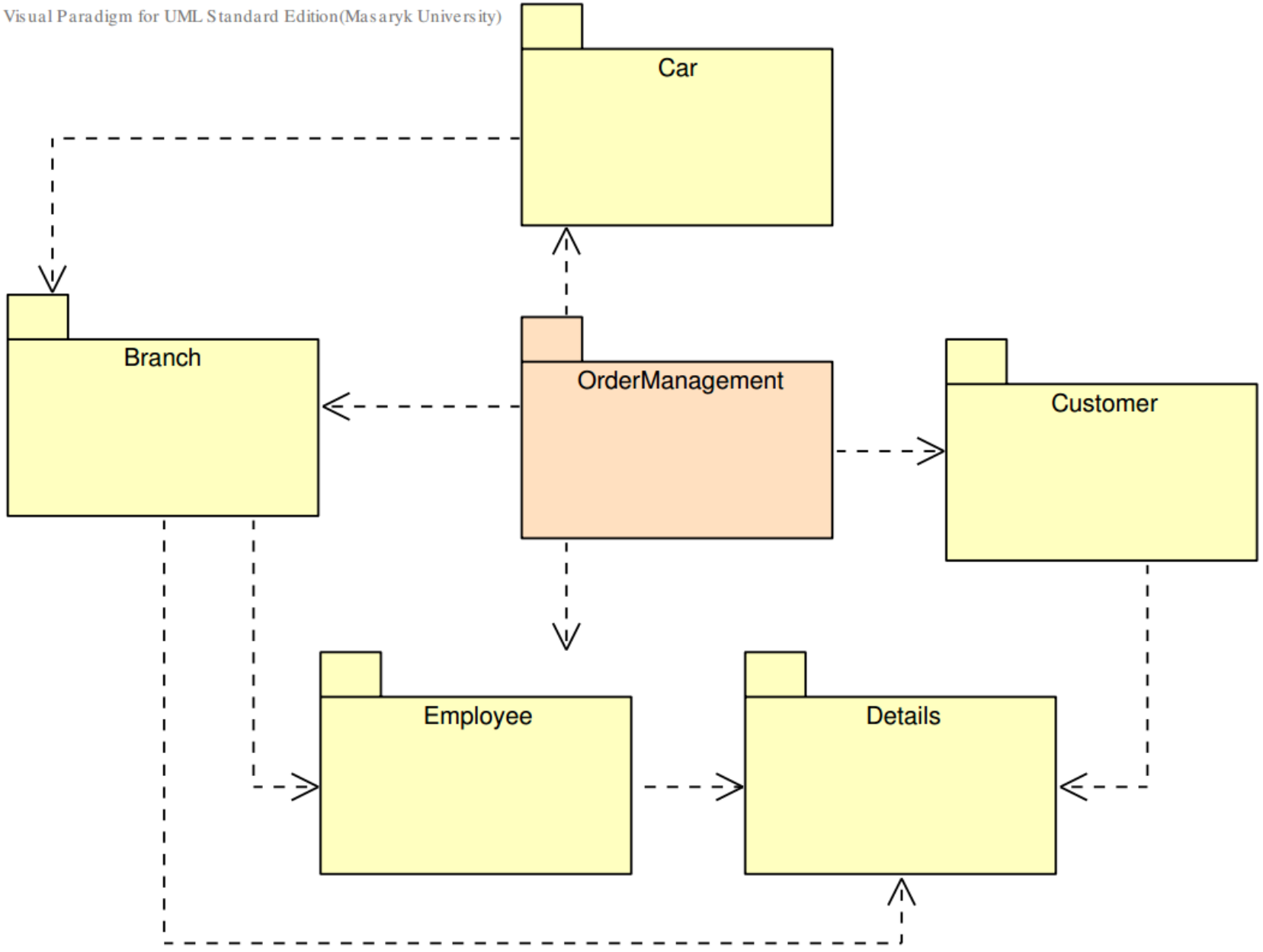


Zdroj: https://praveenthomasln.wordpress.com/tag/uml-packages/

Diagram balíkov a tried (ilustračný obrázok)



Výsledný samostatný diagram balíkov

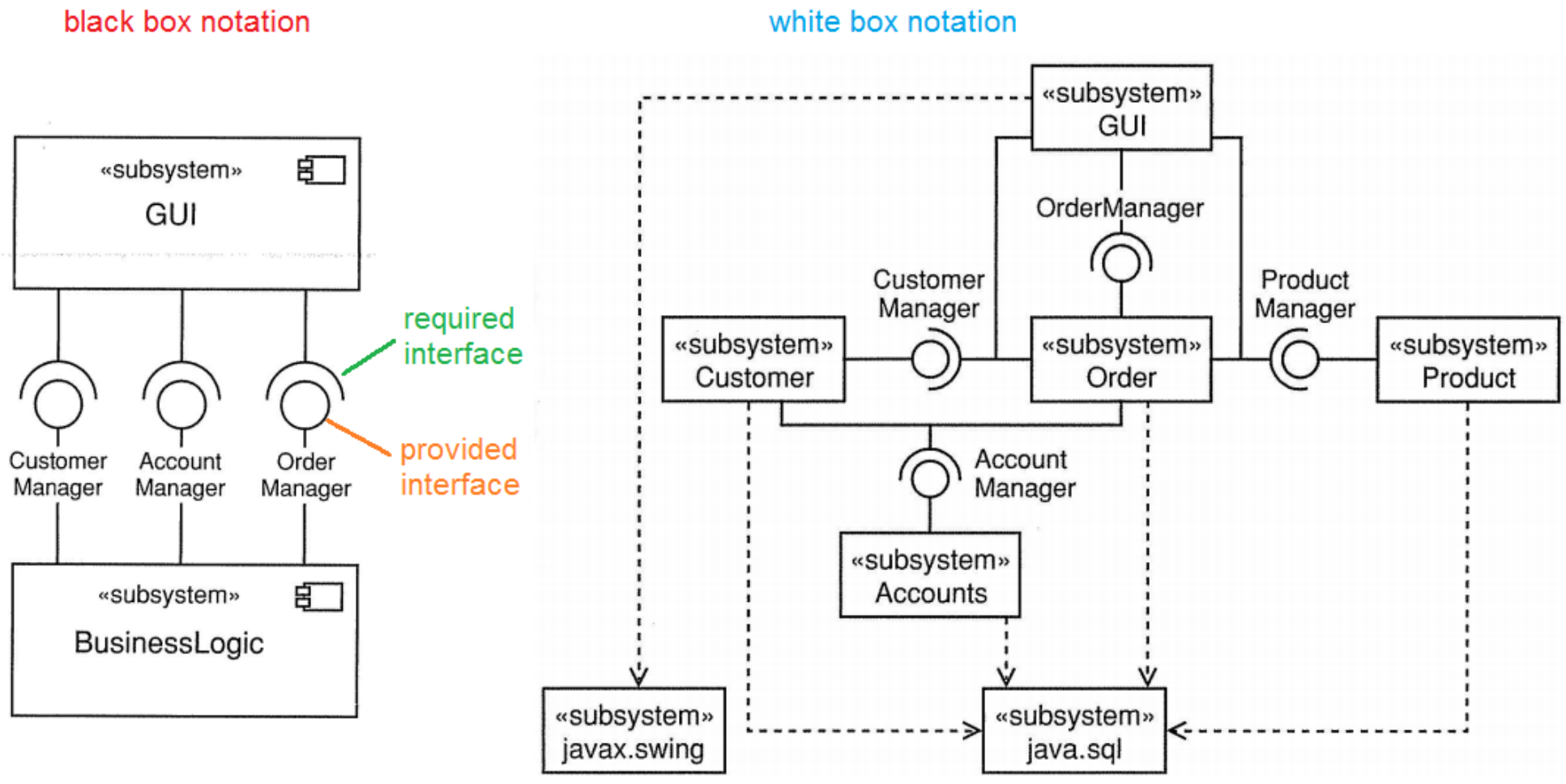


1. Package Diagram (Diagram balíkov)
2. Component Diagram (Diagram komponentov)
3. Deployment Diagram (Diagram nasadenia)
4. Úlohy

Diagram komponentov

- Komponent = modulárna časť systému, ktorej externé chovanie je úplne definované **poskytovanými** a **požadovanými rozhraniami**
  - Modulárna = môže byť nahradená iným komponentom, ak podporuje rovnaký protokol
- Black-box pohľad: vnútro komponentu je skryté
- White-box pohľad: vnútro komponentu je rozpracované

Diagram komponentov – ukážka



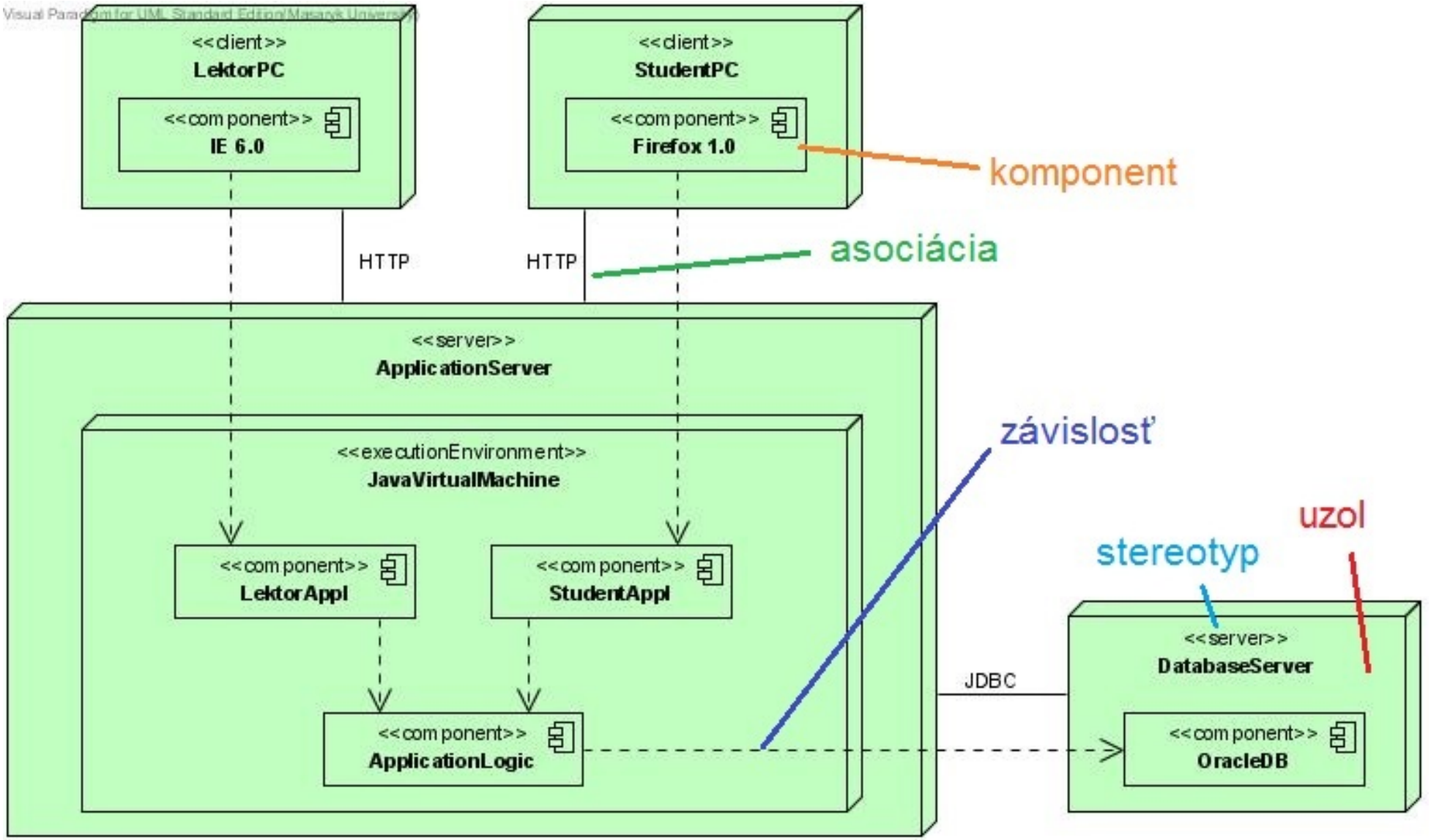
Zdroj: https://is.muni.cz/auth/el/1433/podzim2015/PB007/um/lec/08-Architecture.pdf

1. Package Diagram (Diagram balíkov)
2. Component Diagram (Diagram komponentov)
3. Deployment Diagram (Diagram nasadenia)
4. Úlohy

Diagram nasadenia

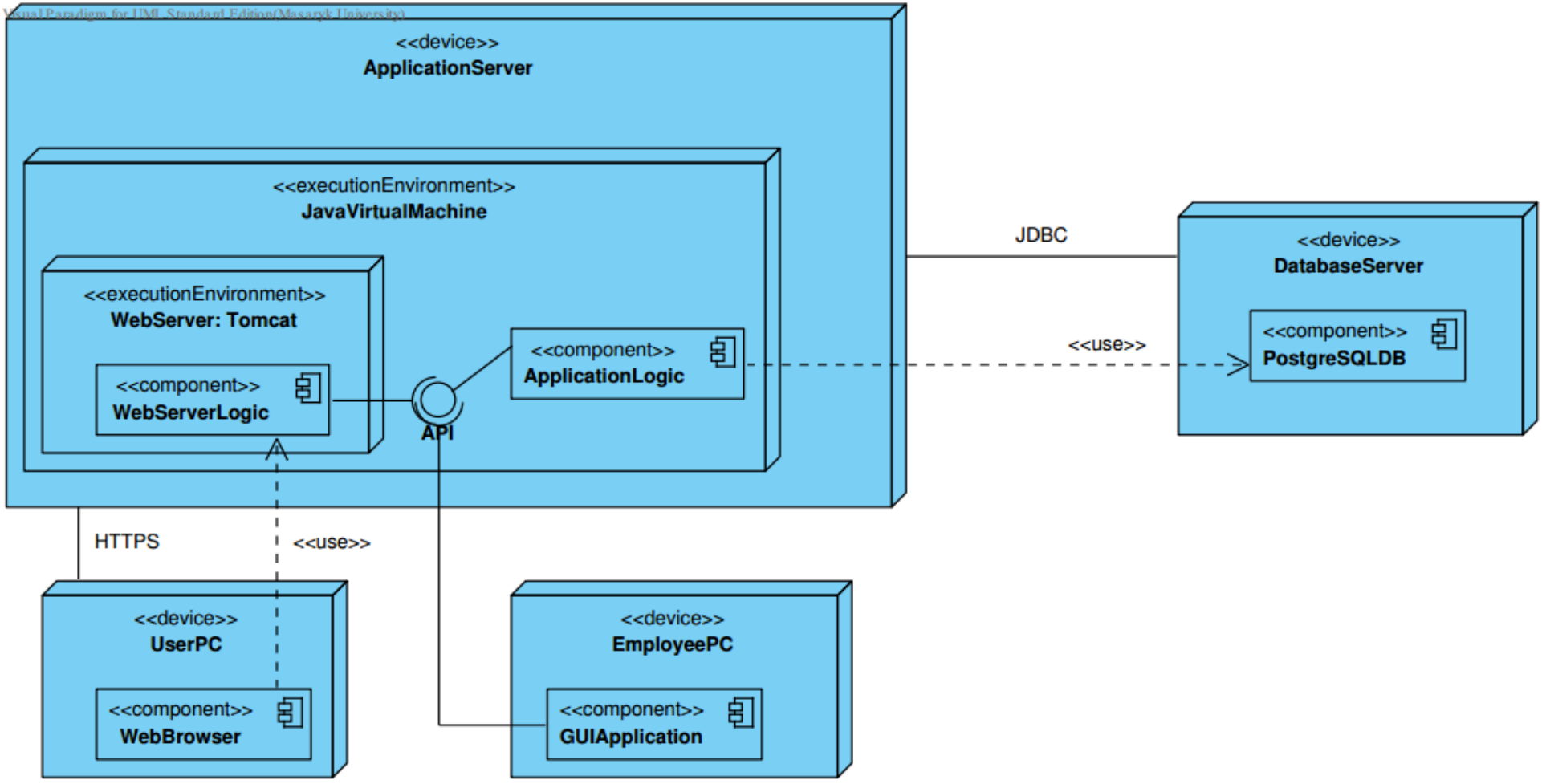
- Diagram nasadenia zobrazuje, ako bude architektúra SW mapovaná na architektúru HW
- Prvky:
  - **Uzly** (nodes) – typ výpočtového zdroja, na ktorý budú umiestnené jednotlivé časti systému
    - «device» – zariadenie, HW
    - «execution environment» – SW prostredie
  - **Artefakty / Komponenty** – SW nasadený na uzloch
    - Artefakt – fyzická úroveň, napr. súbor
    - Komponent – logická úroveň, zoskupenie artefaktov
  - **Rozhrania** pre komunikáciu s komponentami
  - **Asociácie** – spojenia (komunikačné kanály) medzi uzlami (+ názov komunikačného protokolu)
  - **Závislosti** medzi artefaktmi / komponentmi

Diagram nasadenia – ukážka



Zdroj: https://is.muni.cz/auth/el/1433/podzim2015/PB007/um/sem/cz_files/12/12_StudiumDeployment.jpg

Diagram nasadenia – ukážka 2



1. Package Diagram (Diagram balíkov)
2. Component Diagram (Diagram komponentov)
3. Deployment Diagram (Diagram nasadenia)
4. Úlohy

Úlohy

- Do diagramu tried doplňte balíky a presuňte do nich jednotlivé triedy
- Následne vytvorte samostatný **diagram balíkov**, ktorý bude obsahovať iba samotné balíky a závislosti medzi nimi
- Rozmyslite si, z akých komponentov bude pozostávať váš systém a cez aké rozhrania budú spolu komunikovať
- Vytvorte **diagram nasadenia**, ktorý bude dané komponenty mapovať na jednotlivé HW uzly
- Finalizujte projekt – odstráňte prázdne diagramy, skontrolujte aktuálnosť všetkých diagramov
- Vygenerujte PDF report a vložte ho do odovzdávarne „Week 12“ (skupiny 09, 10)
  - Názov v tvare *priezvisko1-priezvisko2-priezvisko3.pdf*
  - Odovzdáva jeden človek za svoj tím
  - Deadline: 13.12. 23:59