

Objektovo Orientované programovanie

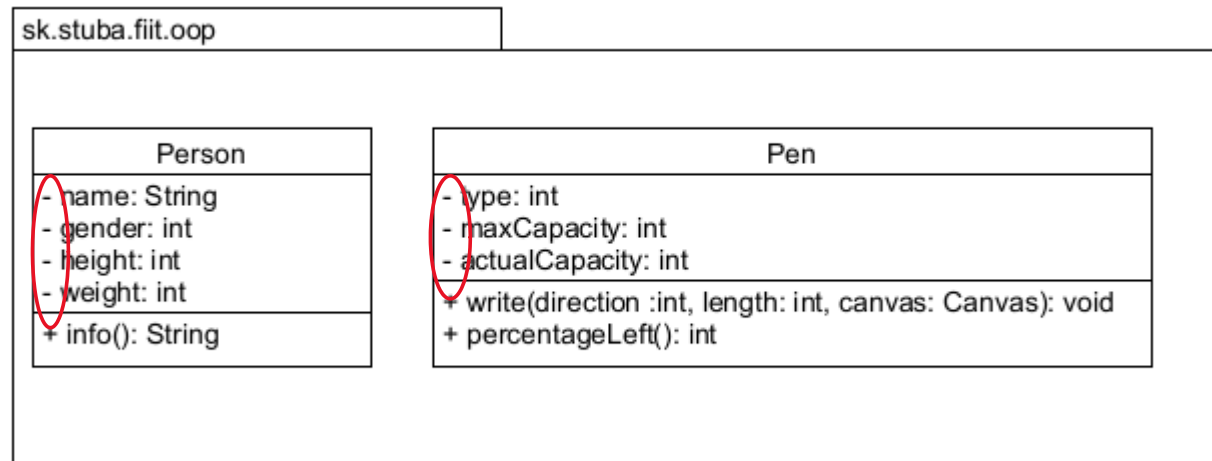
2. Prednáška
LS 2025/2026
Autor: Juraj Petrík

Back to the roots

- Vráťme sa ešte k základným princípom OOP

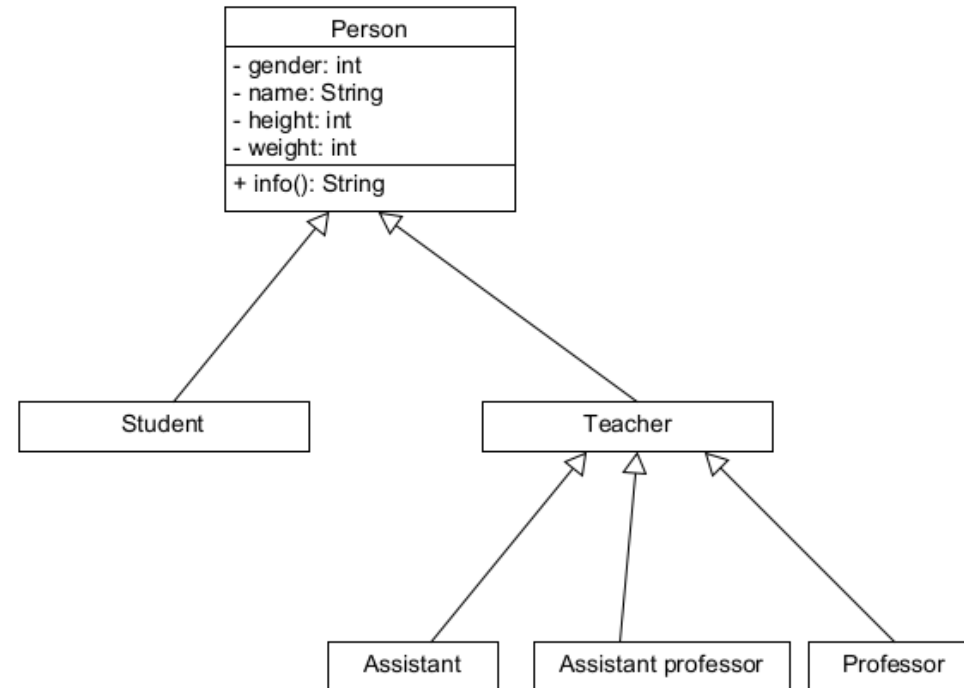
Enkapsulácia (zapuzdrenie)

- Nemanipulujeme priamo s dátami, ale pomocou metód
- V Jave napr. aj keyword `private` (pomáha docieľiť)
- Mutátory, ale nie len to – ide o celkový koncept, že mimo triedy nepracujem priamo s dátami



Dedičnosť

- Umožňuje „odvodzovanie“ tried z iných tried
- Vzniká strom
- V Jave keyword extends

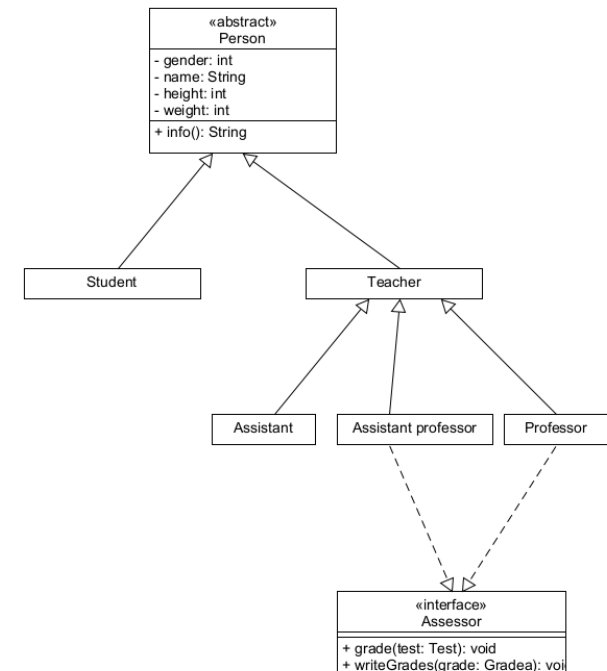


Polymorfizmus

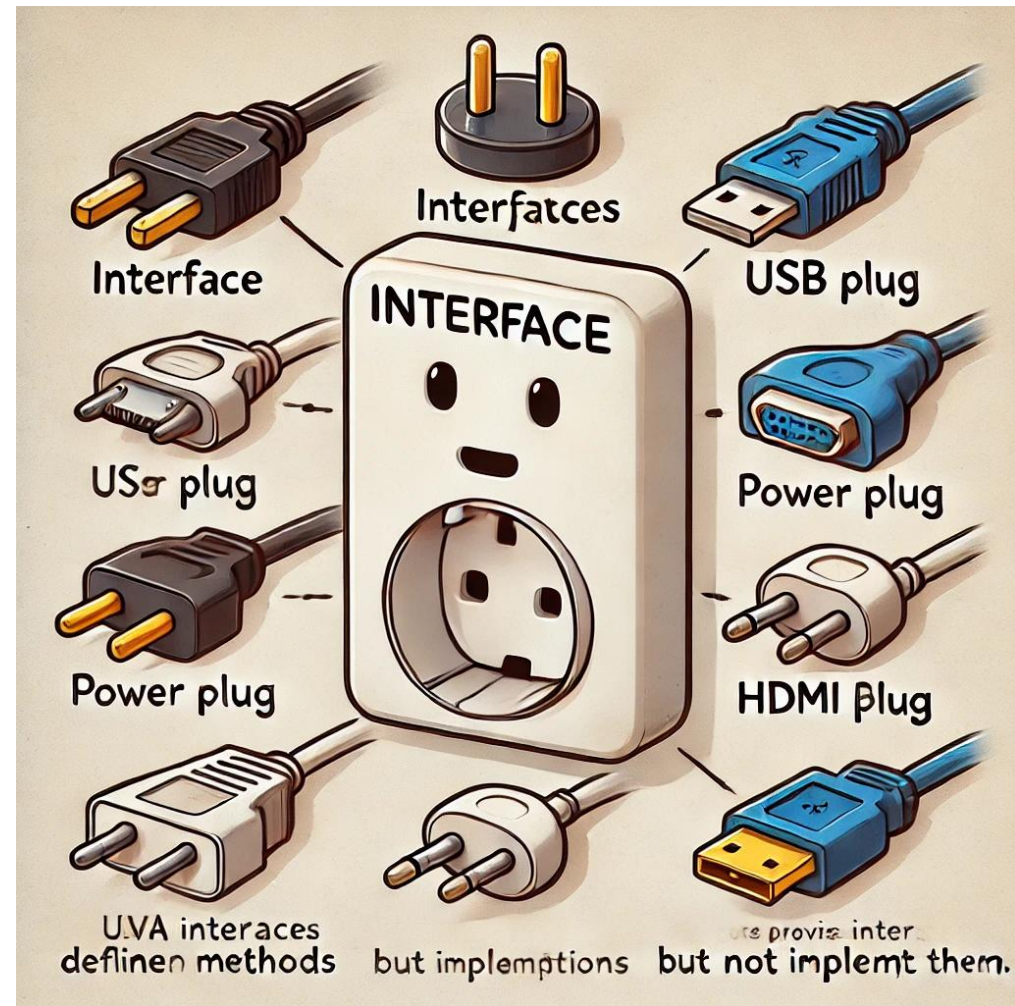
- Compile time
 - Method overloading – rovnaké meno metód, rôzne parametre
- Runtime
 - Method overriding – pri dedení, špecifickosť správania
 - *Virtual functions*

Abstrakcia

- “Skrývanie“ internej implementácie
- Abstrakcia je na úrovni návrhu (design), enkapsulácia na úrovni implementácie
- V Jave napr. (aj pomáha) Abstract class a Interface



Interface



Interface II.



Triedy II.

- Konštruktory:
 - Inicializácia objektu
 - Nededia sa
 - Ak nie je žiadny, je automaticky vytvorený prázdny
- this vs super

Metódy

- Java je **pass by value (copy)**
- Ľubovoľný počet a typ parametrov
- Vraciame max jeden „objekt“

Atribúty triedy vs lokálne premenné

- Atribúty triedy (instance variables) sú deklarované v triede, nie metóde
- Lokálne premenné sú deklarované v metóde
- Atribúty triedy majú vždy default hodnotu
- Lokálne premenné musia byť inicializované pred použitím
- == vs equals()

Modifikátory prístupu

- Triedy
 - Public – od všadiaľ
 - *Default* – iba z rovnakého balíku
- Atribúty, metódy, konštruktory
 - Public – prístupné pre všetky triedy
 - Private – iba vo vlastnej triede
 - *Default* – iba z rovnakého balíku
 - Protected – iba z rovnakého balíku a podtried

Návrh a implementácia triedy

- Čo má trieda robiť?
- Aké atribúty?
- Aké metódy?
- *Testy*
- Pseudokód pre metódy
- Implementácia
- Testovanie
- Ladenie a oprava

Quiz time