

Objektovo orientované programovanie

Učiteľ:
Ing. Jozef Wagner PhD.

Učebnica:
<https://oop.wagjo.com/>

OPG

Teória 15

1. OOP princípy - opakovanie
2. Vzťahy medzi triedami
3. Prevody typov
4. Praktický príklad

Princípy OOP - opakovanie

Zapuzdrenie (anglicky encapsulation): Skrývanie interných dát objektu (private atribúty, modifikátory prístupu) a poskytovanie kontrolovaného prístupu cez verejné metódy (gettre/settre). Chráni dáta pred neoprávnenou zmenou. Atribúty modelujú HAS-A (objekt má) vzťah

Dedičnosť (anglicky inheritance): Podtrieda zdedí vlastnosti a metódy z rodičovskej triedy, umožňuje znovupoužitie kódu. Modeluje IS-A (objekt je) vzťah.

Princípy OOP - opakovanie

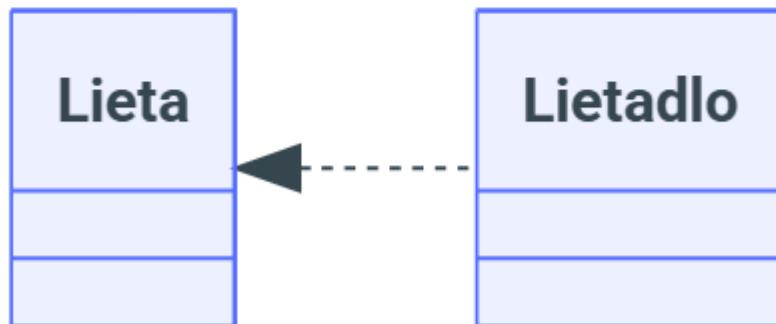
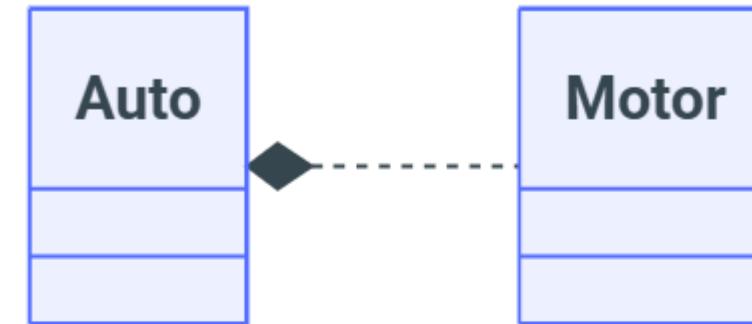
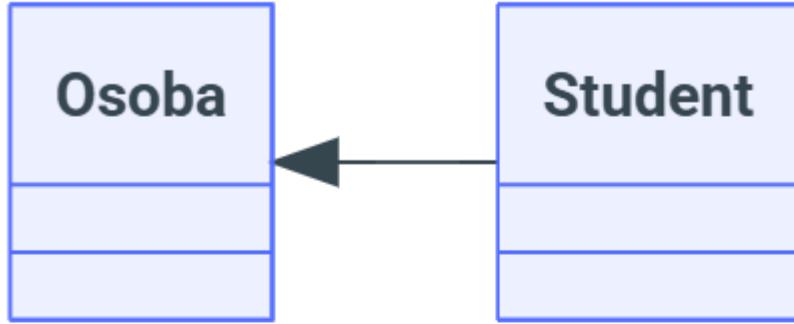
Polymorfizmus: Metódy sa správajú inak v závislosti od typu objektu. Poznáme statický polymorfizmus (preťažovanie metód - overloading), dynamický polymorfizmus (prekrývanie metód - overriding) a parametrický polymorfizmus (generické triedy)

Abstrakcia: Skrývanie zložitých detailov (AKO sa to robí) a zverejnenie len podstatného (ČO trieda robí). Pomocou abstraktných tried alebo rozhraní. Rozhrania modelujú CAN-DO vzťah (objekt vie)

Vzťahy medzi triedami

- Všeobecná **asociácia** - prepojenie
Učiteľ učí Žiaka, Pes žerie Granule
- **IS-A (je)** - vzťah dedičnosti - **extends**.
Auto je Vozidlo, Pes je Zviera
- **CAN-DO (vie)** - vzťah implementácie rozhrania - **implements**.
Lietadlo vie Lietat, ElektrickeAuto sa vie Nabíjať
- **HAS-A (má v sebe)** - vzťah **kompozície** - skladá sa z. **Silná väzba**, triedy bez seba nevedia existovať. Pomocou **atribútov**.
Človek má v sebe Srdce, Dom má v sebe Izby
- **HAS-A (má)** - vzťah **agregácie**. **Slabšia väzba**, trieda obsahuje iné objekty, ale vedia existovať aj samostatne. Pomocou **atribútov**.
Knižnica má Knihy, Auto má Pasažierov

Vzťahy medzi triedami



Casting tried - prevod typov

Upcasting (prevod nahor)

- Automatický prevod objektu z podtriedy na typ rodiča
- Je vždy bezpečný. Podtrieda vždy "je" aj nadradená trieda
- Používa sa veľmi často a automaticky, najmä pri práci s polymorfizmom

Downcasting (prevod nadol)

- Prevod objektu z nadradenej triedy späť na podtriedu.
- Je riskantný, a ak sa nepodarí, program vyhodí výnimku **ClassCastException**.
- Používa sa zriedka, ak potrebujeme pristúpiť k metódam podtriedy. Pred downcastingom sa odporúča vždy použiť operátor **instanceof**.

Praktický príklad

Hierarchia vozidiel