

PB162 Programování v jazyce Java

6. cvičenie

chyby v 2. úlohe

- odovzdať **načas** (-1 b)
- nemeňte triedy zadania, package! (-1 b)
- vhodné rozhrania (-1 b)
- konvencie
 - odsadenie kódu (-0,5 b)
 - private atribúty (-0,5 b)
- zbytočné
 - duplikácia kódu (-0,5 b)
 - atribúty (-0,5 b), lokálne premenné (-0 b)

dedičnosť (*inheritance*)

- 2. pilier OOP
- spôsob vyjadrenia vzťahu nadtyp–podtyp = generalizácia–špecializácia
- vzťah **ISA** (**is a special case of**)
- dnes
 - hierarchie typov
 - abstraktné triedy
 - dedičnosť medzi bežnými triedami

príklad

- poznáte?



príklad

- poznáte?
- do akej triedy patria?



Tatra 600 „Tatraplán“ (1947)



Škoda Octavia (1959)

príklad

- poznáte?



príklad

- do akej triedy patria?



Škoda Liaz 706 RT (1955)



Tatra 148 (1972)

príklad

- do akej triedy patria?
- a čo tie pred tým?



Škoda Liaz 706 RT (1955)



Tatra 148 (1972)

příklad

- společné rysy
- => společné dáta
- objem, výkon motoru
 - maximální rychlost'

osobný automobil

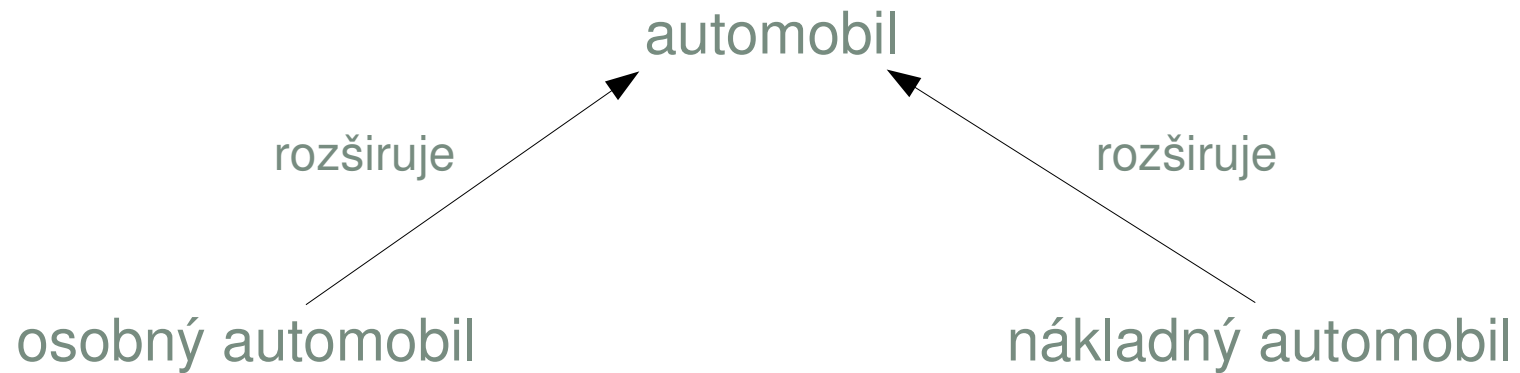


nákladný automobil



príklad

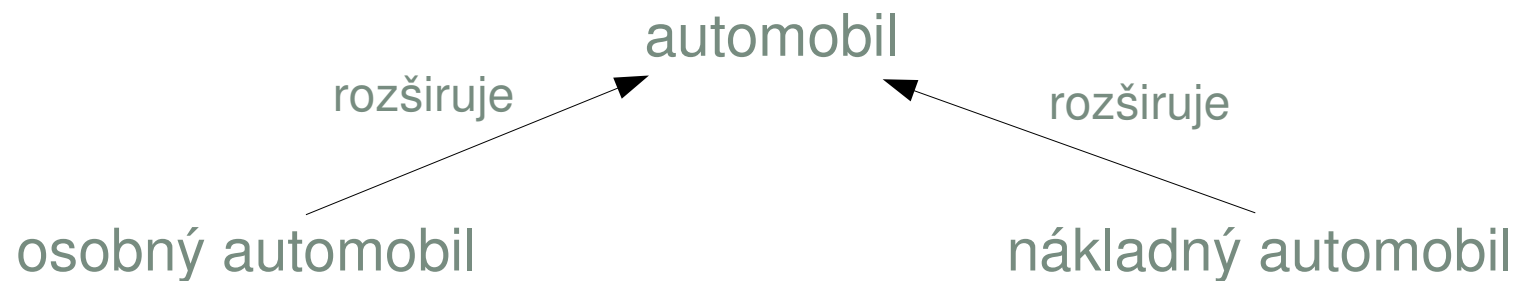
- vyčlenením týchto rysov do spoločného predka vytvoríme jednoduchú **hierarchiu tried**



príklad

- neexistuje automobil, ktorý nie je osobný ani nákladný (:-)

=> spoločný predok je **abstraktný**



príklad

- poznáte?



príklad

- do akej triedy patria?
- kam patria do predchádzajúcej hierarchie?



Škoda 706 RTO LUX (1956)



Škoda ŠD (1963)

příklad

- poznáte?



príklad

- do akej triedy patria?
 - viem, každá do inej :-)



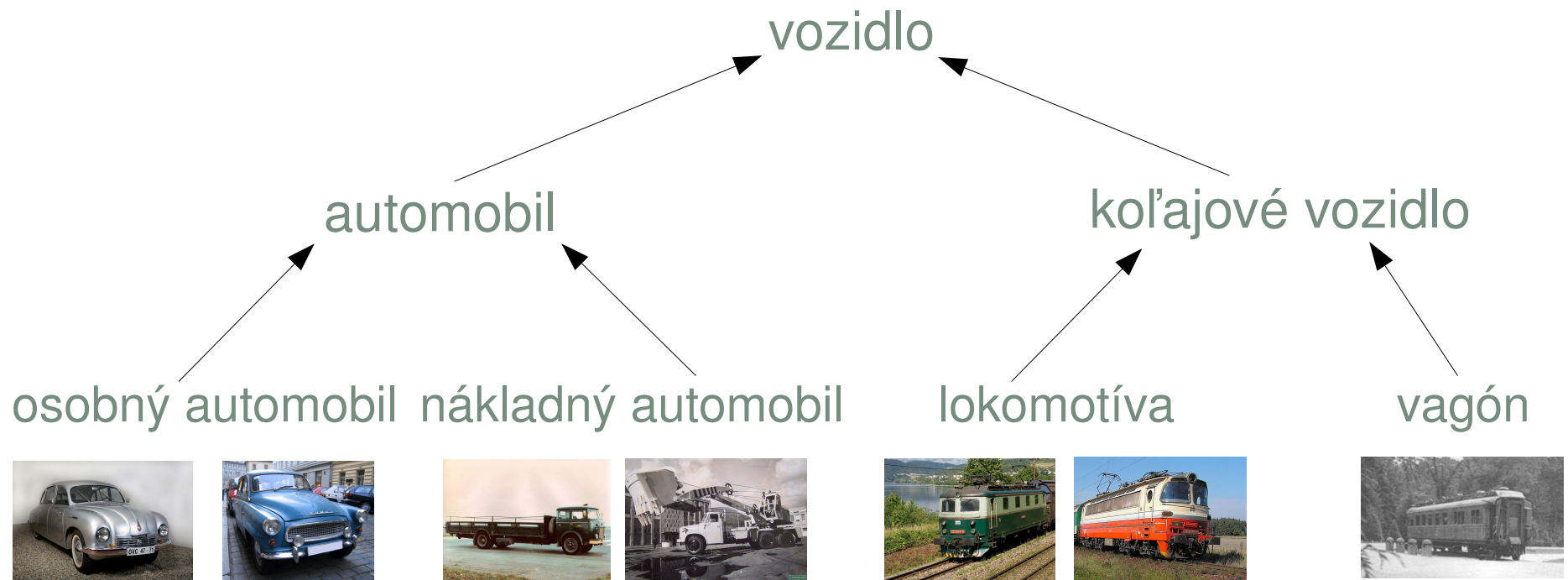
E 469.1 121 004 „Bobina“ (1960)



S 489.0 230 099 „Laminátka“ (1967)

príklad

- hierarchia tried
 - neúplná, ako vždy



dedičnosť

1. implementácia rozhrania

2. rozšírenie abstraktnej triedy

3. rozšírenie triedy

– hrozí zneužitie dedičnosti, lebo „sa to hodí“, aj keď nejde o vzťah ISA

- Circle extends Point

- spoločné x, y; kružnica pridá r

- District extends Municipality

- spoločné: počet obyvateľov, rozloha...

– **NEPOUŽÍVAŤ** – ide o vzťah **PART OF**, NIE ISA!

abstraktná trieda

- definuje stav i správanie
 - rozhranie nedefinovalo stav ani implementáciu správania
- **ale:** nemožno vytvárať inštancie
- trieda určená na rozširovanie
- **použitie** – ak existuje nadtyp
 - spoločné atribúty a metódy pre podtypy
 - nikdy neexistuje samostatne, vždy len v nejakom podtype

dedičnosť – syntax

```
abstract class Vehicle {  
    protected double weight; //lepiej pouzít private!!  
    public abstract void unload();  
    public getWeight() {  
        return weight;  
    }  
}
```

```
abstract class Automobile extends Vehicle {  
    protected String spz; //lepiej pouzít private!!  
}
```

```
class Truck extends Automobile{  
    protected double payload; //lepiej pouzít private!!  
    public void unload() {  
        ...  
    }  
    ...  
}
```

dedičnosť – volanie metód predka

```
abstract class Vehicle {
    protected double weight; //lepiej pouzít private!!
    public Vehicle (double weight) {
        this.weight = weight;
    }
}

abstract class Automobile extends Vehicle {
    protected String spz; //lepiej pouzít private!!
    public Automobile (String spz, double weight) {
        super(weight);
        this.spz = spz;
    }
}

class Truck extends Automobile {
    protected double payload; //lepiej pouzít private!!
    public void unload() {
        super.openDoor(...) {
            ...
        }
    }
}
```

viditeľnosť – zhrnutie

- člen triedy (trieda, atribút, metóda)
 - **public** – verejný
 - bez modifikátora – *package-private*
 - **protected** – viditeľný z triedy a jej potomkov
 - **private** – viditeľný len z triedy
- <http://java.sun.com/docs/books/tutorial/java/java>

šachový príklad

- figúrky (veža, strelec, dáma)
 - pohybujú sa podľa pravidiel
 - **veža**: `newColumn == column || newRow == row`
- všetky majú spoločné:
 - polohu (`column, row`)
 - pravidlá – „neujst“ zo šachovnice
- doprogramovať ďalšiu figúrku – príprava na písomku

1. vnútrosemestrálna písomka

- budúci týždeň
- najprv uvažovať, potom programovať
 - ceruzka, papier
 - prečítajte **celé** (naozaj celé...) zadanie
 - značte si, kreslite diagramy tried, **uvažujte** (*návrh*)
 - programujte (*kódovanie*)
- hlavná činnosť „programátora“ je **premýšľanie**