

Informatika 1

O predmete



Vyučujúci

- Prednáša:
 - doc. Ing. Ján Janech, PhD., A114, jan.janech@fri.uniza.sk
- Cvičí:
 - Ing. Alexander Brezáni, VB308, alexander.brezani@fri.uniza.sk
 - Ing. Michal Ďuračík, PhD., RA113, michal.duracik@fri.uniza.sk
 - Ing. Radoslav Gajdoš, GlobalLogic, radoslav.gajdos@globallogic.com
 - doc. Ing. Ján Janech, PhD. , RA114, jan.janech@fri.uniza.sk
 - doc. Ing. Marek Kvet, PhD., RA325, marek.kvet@fri.uniza.sk
 - Ing. Pavol Kysela, Scheidt & Bachmann, kysela.pavol@scheidt-bachmann.sk
 - Ing. Martin Mazúch, VB308, Martin.Mazuch@fri.uniza.sk
 - Ing. Matej Meško, PhD., RA124, Matej.Mesko@fri.uniza.sk
 - Ing. Adam Mračko, RA316, adam.mracko@fri.uniza.sk
 - Ing. Štefan Toth, PhD. RA124, Stefan.Toth@fri.uniza.sk
 - Ing. Monika Václavková, PhD., RA211, monika.vaclavkova@fri.uniza.sk

Cieľ predmetu

- úvod do programovania všeobecne
- dôraz na objektový prístup
- základné princípy tvorby algoritmov
- programovací jazyk Java – iba nástroj

Hodnotenie predmetu

Položka hodnotenia	Potrebný počet bodov	Max. počet bodov
Hodnotenie práce cez semester	25	50
Hodnotenie praktickou skúškou	25	50
Spolu	61 (na „E“)	100

Bodové hodnotenie	Hodnotenie známkou
<93, 100>	A – výborne
<85, 93)	B – veľmi dobre
<77, 85)	C – dobre
<69, 77)	D – uspokojivo
<61, 69)	E – dostatočne
<0, 61)	Fx – nevyhovel

Priebežné hodnotenie

Položka hodnotenia	Potrebný počet bodov	Max. počet bodov
Prvý test (~4. týždeň)	0	5
Druhý test (~8. týždeň)	0	5
Tretí test (~12. týždeň)	0	5
Semestrálna práca	5	20
Checkpointy	0	3
Domáce úlohy	0	6
Aktivity na cvičení	0	6
Spolu	25	50

Semestrálna práca

Položka hodnotenia	Očakávaný počet bodov	Max. počet bodov
1. Checkpoint (8. týždeň)	1	1
2. Checkpoint (10. týždeň)	2	2
Odovzdaná a obhájená semestrálna práca	10	20

Dochádzka

- prednášky – nepovinné
- cvičenia – nepovinné
 - POZOR! mizivá šanca spraviť predmet bez navštevovania cvičení (skúsenosti)
 - viac ako 2 absencie = vylúčenie z cvičenia
 - možnosť získať body aj bez účasti na cvičeniach
- pred cvičením treba vypracovať test

Konzultačné hodiny

Ing. Alexander Brezáni	Streda 12:00 – 14:00
Ing. Michal Ďuračík, PhD.	Streda 10:00 – 12:00
Ing. Radoslav Gajdoš	???
doc. Ing. Ján Janech, PhD.	Pondelok 12:00 – 13:00, Streda 12:00 – 13:00
doc. Ing. Marek Kvet, PhD.	Streda 11:00 – 12:00, Štvrtok 9:00 – 10:00
Ing. Pavol Kysela	???
Ing. Martin Mazúch	Streda 14:00-15:00
Ing. Matej Meško, PhD.	Pondelok 13:00 – 14:00
Ing. Adam Mračko	Pondelok 13:00 – 14:00
Ing. Štefan Toth, PhD.	Streda 13:00 – 15:00
Ing. Monika Václavková, PhD.	Utorok 13:30 – 14:30, Streda 12:30 – 13:30

Skúška

- max. 50 bodov
- potrebných aspoň 25 bodov
- praktická
- dostanete zadanie
- 120 minút
- posledný týždeň – príprava

Podvádzenie

- plagiáty semestrálnych prác
- plagiáty domácich úloh
- automatická kontrola na konci semestra
- podvody na skúške
- disciplinárne konanie
 - podmienené vylúčenie zo štúdia
 - vylúčenie zo štúdia

Informatika 1

Prednáška 1 – Princípy objektového programovania



Veci okolo nás



Televízor – ako sa používa?



+ návod na použitie



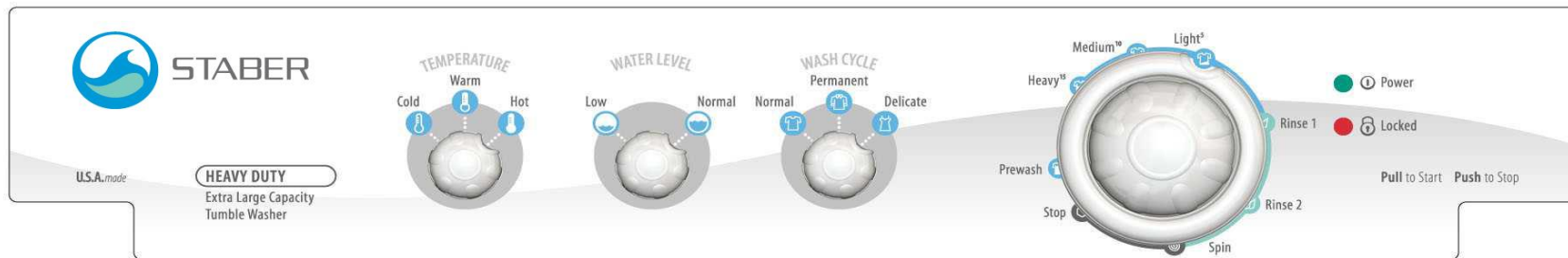
Televízor – čo je vo vnútri?

- z čoho sa skladá?
- ako funguje?
- musíme to vedieť, keď ho používame?

Veci okolo nás



Práčka – ako sa používa?



Veci okolo nás

- pohľad vonkajší
 - na čo nám slúži?
 - ako sa používa?
- pohľad vnútorný
 - z čoho sa skladá?
 - ako sa dá charakterizovať?
 - ako funguje?

Veci okolo nás



Objekt televízor



Televízor má vlastnosti

- vek
- veľkosť uhlopriečky
- farebný alebo čiernobiely
- typ obrazovky:
 - klasická (elektrónka),
 - tekuté kryštály (LCD),
 - OLED,
 - ...
- ...

Televízor má funkcie

- zobrazuje vysielanie TV stanice, alebo monoskop, alebo „sneženie“
- vypne sa alebo zapne sa na požiadanie
- zmení TV stanicu na požiadanie
- zmení hlasitosť na požiadanie
- ...

Objekt práčka



Práčka má vlastnosti

- rozmery
- hmotnosť dávky prádla
- trieda úspornosti
- značka
- plnenie zhora alebo zboku
- ...

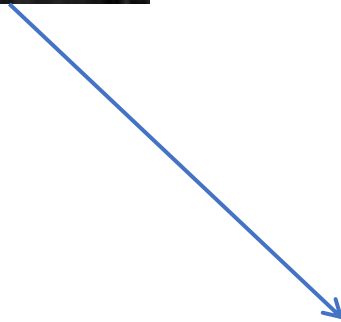
Práčka má funkcie

- perie
- zohrieva vodu
- plácha
- odstred'uje (žmýka)
- ...

Človek ako objekt

- komunikuje so svojim okolím?
 - má vlastnosti?
 - má svoje „časti“?
 - má „funkcie“?
-
- môžeme ho chápať ako objekt.
 - podobne s trochou fantázie aj ostatná príroda.
 - všetko okolo nás sú objekty.

Objekty spolupracujú



Objekty spolupracujú

- objekty spolupracujú tak, že si odovzdávajú potrebné informácie
 - človek stlačí tlačidlo na diaľkovom ovládači – číslo kanálu zvoleného TV programu
 - ovládač odošle informáciu do TV prijímača
 - TV prijímač naladí požadovaný TV program a zobrazuje ho človekovi

Objekty spolupracujú

všeobecne budeme hovoriť, že
objekty si posielajú správy

Základné pojmy OP

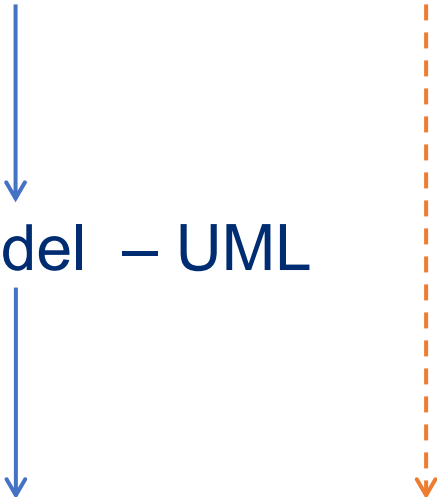
- objekt
- správa

Modelovanie objektov

Objekt reálneho sveta vs. objekt v OOP

- model je zjednodušením reality – uvažuje iba stránky podstatné pre riešený problém
- model – fyzikálny, matematický, programový, ...
- programový objekt modeluje objekty reálneho sveta
- program je množina objektov, ktoré spolu komunikujú pomocou správ
- programy sa píšu v programovacích jazykoch
- my budeme používať jazyk Java [džava]

Postup pri tvorbe programu

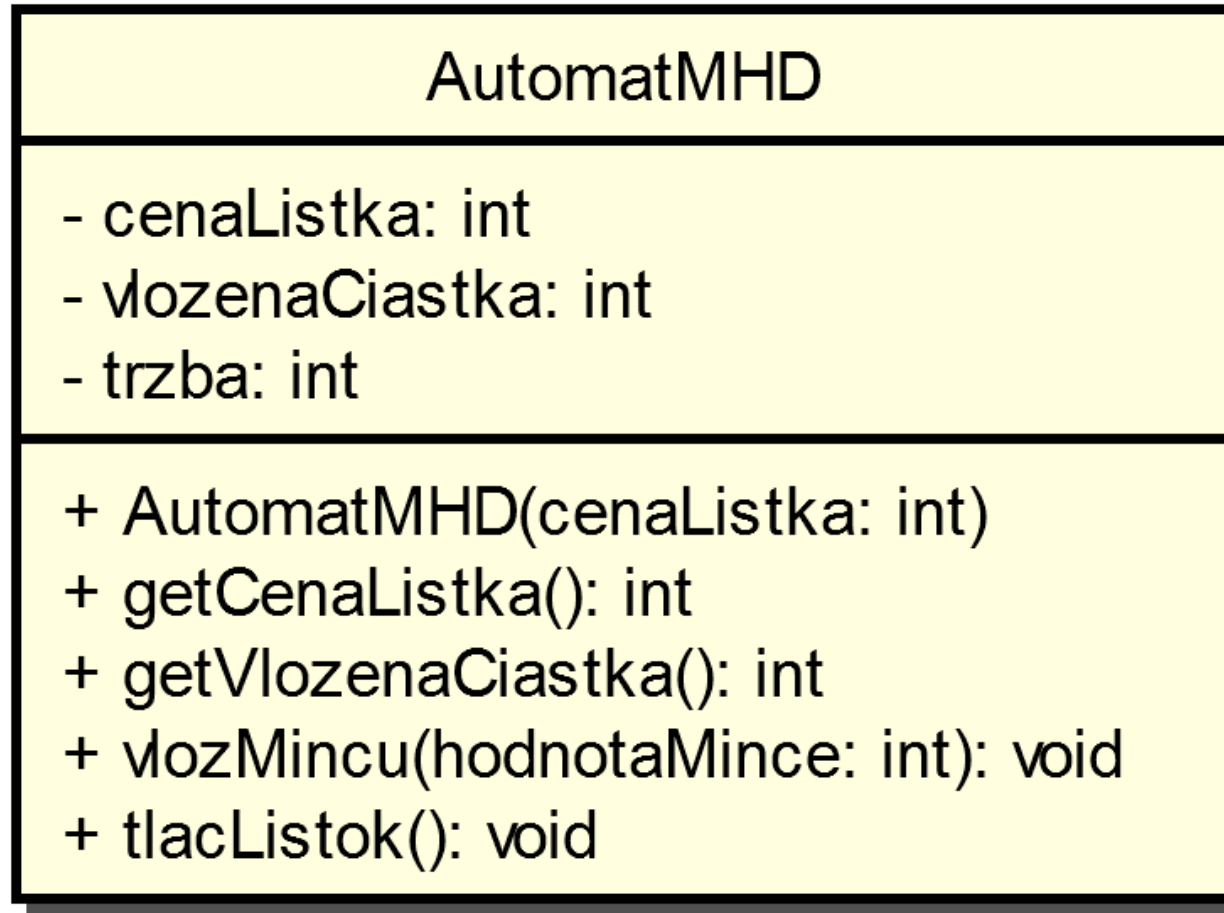
- objekt reálneho sveta
 - logický model – UML
 - softvérový model – program v Java
- 
- ```
graph TD; A[• objekt reálneho sveta] -- solid blue arrow --> B[• logický model – UML]; B -- solid blue arrow --> C[• softvérový model – program v Java]; A -. dashed orange arrow .-> C;
```



# Reálny objekt



# UML model



# Softvérový model

```
/**
 * Konštruktor vytvorí automat, ktorý bude
 * tlačit cestovné listky pevnej ceny.
 * Cena je určená parametrom cenaListka.
 * Pozor - cena listka musí byť kladné celé
 * číslo a táto podmienka sa nekontroluje.
 */
public AutomatMHD(int cenaListka) {
 this.cenaListka = cenaListka;
 this.vlozenaCiastka = 0;
 this.trzba = 0;
}
```

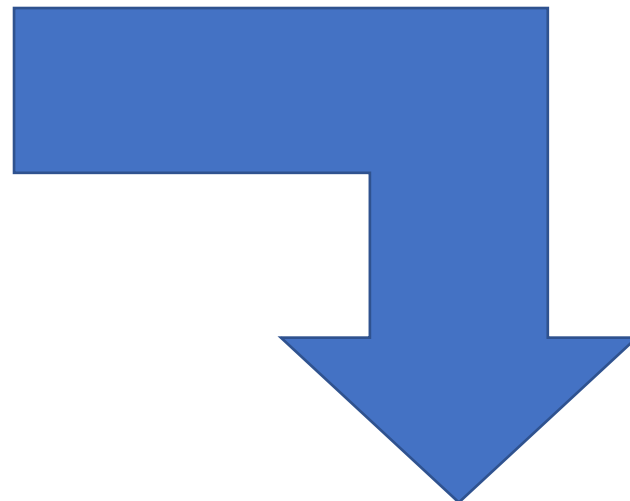
# Používanie programu

- program v dvoch tvaroch
  - zdrojový kód (zdrojový text) – zapisuje aj číta človek – počítač mu nerozumie
  - strojový/cieľový kód – podľa neho pracuje počítač – človek nie je schopný jednoducho prečítať, alebo zmeniť
- prevod zo zdrojového do strojového kódu zabezpečuje špeciálny program – prekladač (kompilátor)
- počítač vykonáva program v strojovom kóde

# Používanie programu

```
/**
 * Konstruktor vytvori automat, ktorý bude
 * tlačit cestovné listky pevnej ceny.
 * Cena je určená parametrom cenaListka.
 * Pozor - cena listka musí byť kladné celé
 * číslo a táto podmienka sa nekontroluje.
 */
public AutomatMHD(int cenaListka) {
 this.cenaListka = cenaListka;
 this.vlozenaCiastka = 0;
 this.trzba = 0;
}
```

# preklad

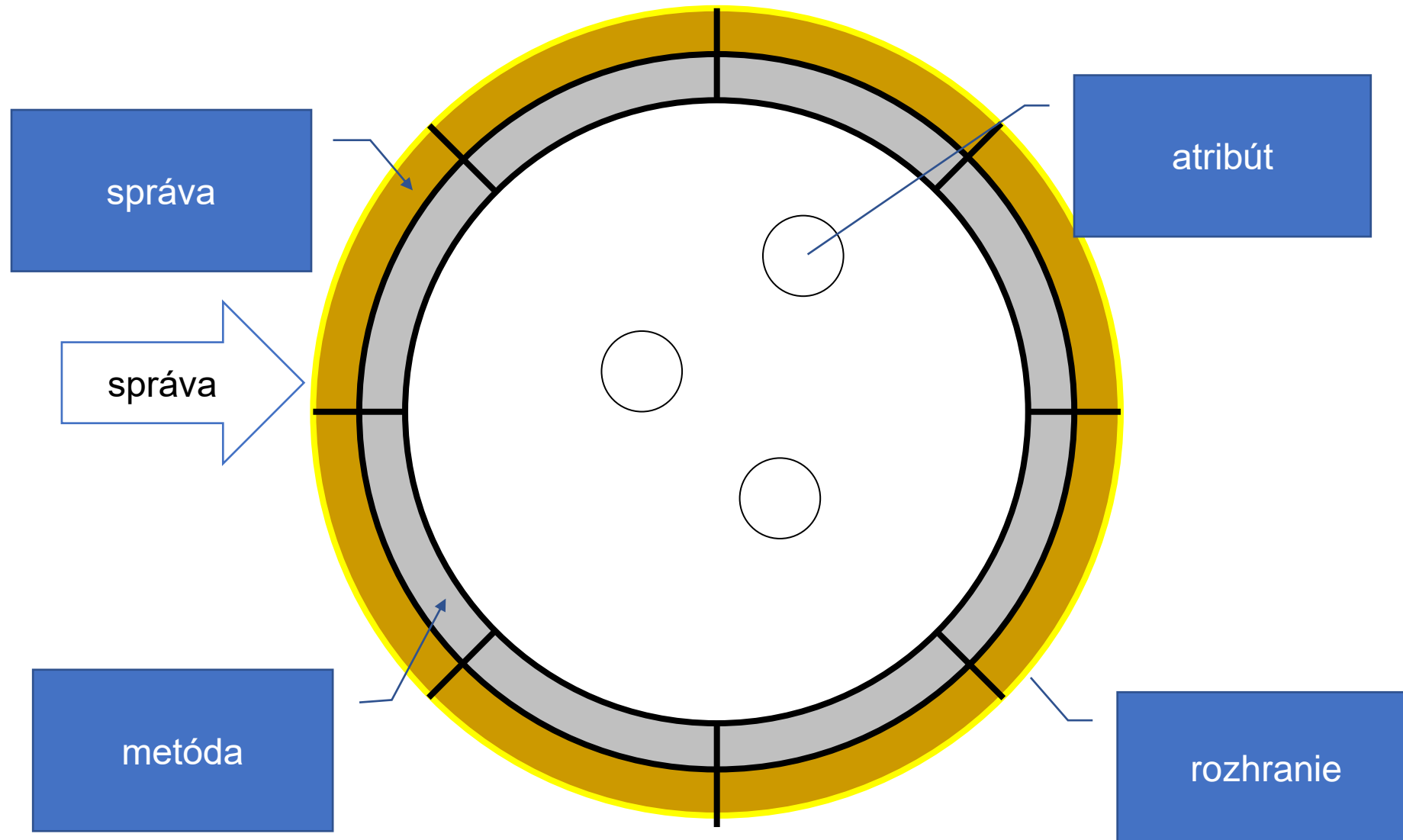


|           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |                   |
|-----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-------------------|
| 000000A0: | 65 | 72 | 54 | 61 | 62 | 6C | 65 | 01 | 00 | 12 | 4C | 6F | 63 | 61 | 6C | 56 | erTableLocalV     |
| 000000B0: | 61 | 72 | 69 | 61 | 62 | 6C | 65 | 54 | 61 | 62 | 6C | 65 | 01 | 00 | 04 | 74 | variableTablet    |
| 000000C0: | 68 | 69 | 73 | 01 | 00 | 0C | 4C | 41 | 75 | 74 | 6F | 6D | 61 | 74 | 4D | 48 | hisLocalAutomatMH |
| 000000D0: | 44 | 3B | 01 | 00 | 0C | 70 | 61 | 43 | 65 | 6E | 61 | 4C | 69 | 73 | 74 | 6B | D;paCenaListk     |
| 000000E0: | 61 | 01 | 00 | 0D | 64 | 61 | 6A | 43 | 65 | 6E | 75 | 4C | 69 | 73 | 74 | 6B | aadajCenuListk    |
| 000000F0: | 61 | 01 | 00 | 03 | 28 | 29 | 49 | 01 | 00 | 11 | 64 | 61 | 6A | 56 | 6C | 6F | a()IdajUlo        |
| 00000100: | 7A | 65 | 6E | 75 | 43 | 69 | 61 | 73 | 74 | 6B | 75 | 01 | 00 | 03 | 28 | 29 | zenuCiastku()     |
| 00000110: | 46 | 01 | 00 | 09 | 76 | 6C | 6F | 7A | 4D | 69 | 6E | 63 | 75 | 01 | 00 | 0E | FvlozMincu        |
| 00000120: | 70 | 61 | 48 | 6F | 64 | 6E | 6F | 74 | 61 | 4D | 69 | 6E | 63 | 65 | 01 | 00 | paHodnotaMince    |
| 00000130: | 0A | 74 | 6C | 61 | 63 | 4C | 69 | 73 | 74 | 6F | 6B | 01 | 00 | 03 | 28 | 29 | tlacListok()      |
| 00000140: | 56 | 01 | 00 | 0A | 74 | 6C | 61 | 63 | 4C | 69 | 73 | 74 | 6B | 61 | 01 | 00 | UtlacListka       |
| 00000150: | 0A | 53 | 6F | 75 | 72 | 63 | 65 | 46 | 69 | 6C | 65 | 01 | 00 | 0F | 41 | 75 | SourceFileAu      |
| 00000160: | 74 | 6F | 6D | 61 | 74 | 4D | 48 | 44 | 2E | 6A | 61 | 76 | 61 | 0C | 00 | 16 | tomatMHD.java     |
| 00000170: | 00 | 25 | 0C | 00 | 12 | 00 | 13 | 0C | 00 | 14 | 00 | 13 | 0C | 00 | 15 | 00 | %                 |

# Teoretický model objektu

- rozhranie – zoznam správ, ktoré prijíma
- verejná časť – majú prístupné (vidia) všetky objekty v okolí
- atribúty
- metódy
- neverejné časti, má prístupné (vidí) len objekt sám

# Ilustrácia štruktúry objektu



# Rozhranie objektu

- zoznam všetkých správ
- ponuka služieb poskytovaných objektom
- potrebujeme poznať, ak chceme objekt použiť
- jeden objekt komunikuje s iným objektom cez jeho rozhranie



# Atribúty objektu

- atribúty objektu modelujú jeho vlastnosti a štruktúru
- atribút má názov a hodnotu
- príklady:
  - výška práčky: 80 cm
  - vek človeka: 19 rokov
  - pohlavie človeka: žena
  - značka televízora: Tesla
  - občan SR: áno

# Stav objektu

- aktuálne hodnoty všetkých atribútov objektu nazývame stav objektu.
- zmenou hodnoty ktoréhokoľvek atribútu sa mení stav objektu.
- kedy sú dva objekty v rovnakom stave?
- na základe čoho sa zmení stav objektu?

# Chovanie objektu

- objekt zachováva svoj stav, kým nie je nútený zmeniť ho na základe prijatej správy.

# Metódy objektu

- reakcia objektu na doručení správu – metóda
- metóda – algoritmus (predpis), čo urobiť, keď objekt príjme správu
- metódy určujú možné chovanie objektu

# Správa → metóda

- každej správe z rozhrania objektu je priradená práve jedna metóda objektu
- protokol – zobrazenie množiny správ na množinu metód
- v programovacích jazykoch (až na výnimky) správa a metóda majú rovnakú štruktúru
- poslanie správy – volanie metódy

# Objekt, trieda a inštancia

Objekt

Trieda



Objekt

Inštancia



Objekt

Inštancia



# Objekt, trieda a inštancia

- továreň – objekt, ktorý vyrába iné objekty – výrobky
  - (často rovnako pomenované ako továreň)
- trieda – objekt, ktorý vytvára rovnaké objekty
- inštancia – objekt, ktorý vyrobila trieda
- príklad:
  - škoda
    - trieda na autá značky Škoda
  - škoda\_ZA765CK, škoda\_KM583AZ
    - inštancie triedy Škoda

# Trieda ako továreň

- objekt – prijíma správy
- špeciálny objekt – vytvára (svoje) inštancie
- kedy trieda vytvorí novú inštanciu?
- špeciálna správa – žiadosť o novú inštanciu



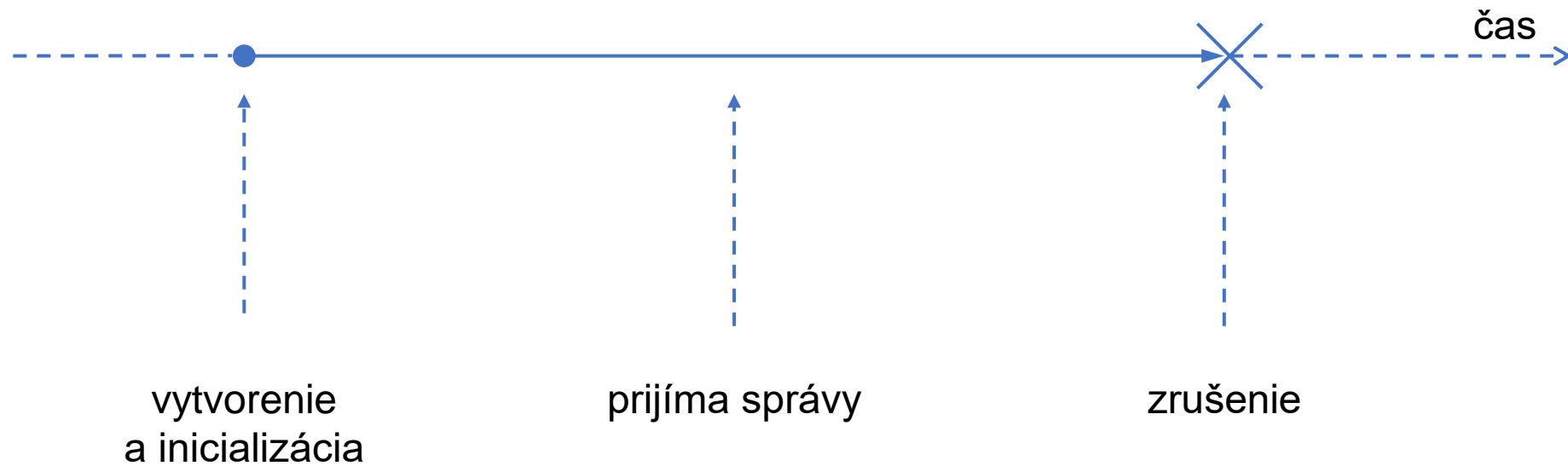
# Trieda ako typ

- na triedu sa dá pozerieť aj ako na typ objektu
- škoda je typ všetkých áut, ktoré boli vyrobené továrňou Škoda

# Životný cyklus inštancie

- príklad:
  - továrneň vyrobí nové auto
  - auto používa jeho vlastník (vlastníci)
  - posledný vlastník odovzdá staré auto do zberných surovín
  - alebo na mieste, kde ho prevezme zberná služba

# Ilustrácia životného cyklu



# Životný cyklus inštancie

- časový interval existencie inštancie
- na začiatku inštancia vzniká – vytvorí ju trieda
  - v okamihu vzniku objektu musí byť definovaný aj jeho stav – začiatočný stav
  - v programovacích jazykoch je za nastavenie začiatočného stavu zodpovedná špeciálna metóda nazývaná konštruktor
- inštancia žije – poskytuje služby
- na konci sa inštancia na požiadanie zničí, alebo ju zruší zberná služba – garbage collector

# Názov objektu

- názov objektu je jeho pomenovanie, pomocou ktorého sa naň odvolávame
- objekt môže mať aj viac názvov
  - Ja, ty, Jožko, pán majster – všetko sa vzťahuje na rovnaký objekt
- rovnaký názov môže v rôznych okamihoch odkazovať na rôzne objekty
  - obsluhovaný klient – najskôr je to pán Novák a potom ho vystrieda pani Krásna, atď.
- názov nie je unikátny

# Štruktúra správy

- adresát.selektor(parameter1, parameter2, ...)
- príklad správy
- kruhModry.posunVodorovne(100)

# Časti správy

- adresát – objekt, ktorému je správa určená
  - kruhModry
- názov správy – selektor, naznačuje, o čo adresáta žiadame
  - posunVodorovne
- parametre – spresňujú žiadosť
  - (100)
- návratová hodnota – možná odpoveď objektu na správu

# Literatúra

- BARNES, D., KÖLLING, M.: Objects First with Java: A Practical Introduction Using BlueJ. 3rd edition. Pearson. 2006. ISBN 978-0-13-197629-0.
- PECINOVSKÝ, R.: Java 7, učebnice objektové architektury pro začátečníky. Praha: Grada publishing. 2012. ISBN 978-80-247-3665-5.
- PELÁNEK, R.: Programátorská cvičebnice, Algoritmy v příkladech. Praha: Computer press. 2012. ISBN 978-80-251-3751-2.
- materiály na predmetovom Moodle