

Informatika 1

O predmete

- Prednáša (Lectures):
 - doc. Ing. Ján Janech, PhD., A114, jan.janech@kst.uniza.sk
- Cvičí (Seminars):
 - doc. Ing. Ján Janech, PhD. , RA114, jan.janech@kst.uniza.sk
 - RNDr. Peter Varša, PhD., RA209, peter.varsa@fri.uniza.sk
 - Ing. Monika Václavková, PhD., RA211, monika.vaclavkova@fri.uniza.sk
 - Ing. Michal Varga, PhD., RA319, michal.varga@fri.uniza.sk
 - Ing. Ján Boháčik, PhD., RA210, jan.bohacik@fri.uniza.sk
 - Ing. Matej Meško, PhD., RA124, matej.mesko@fri.uniza.sk
 - Ing. Marek Kvet, PhD., RA315, marek.kvet@fri.uniza.sk
 - Mgr. Iveta Jančigová PhD., RA124, iveta.jancigova@fri.uniza.sk
 - Ing. František Kajánek, RA316, frantisek.kajanek@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	Bodové hodnotenie
Priebežné hodnotenie	max. 50 bodov
Bonusové body	max. 15 bodov
Skúška	max. 50 bodov
Spolu	max. 115 bodov

Bodové hodnotenie	Hodnotenie známkou
<93, 115>	A – výborne
<85, 93)	B – veľmi dobre
<77, 85)	C – dobre
<69, 77)	D – uspokojivo
<61, 69)	E – dostatočne
<0, 61)	Fx – nevyhovelo

Priebežné hodnotenie

Položka hodnotenia	Potrebný počet bodov	Max. počet bodov
Prvý test (~4. týždeň)	0	10
Druhý test (~8. týždeň)	0	10
Tretí test (~12. týždeň)	0	10
Semestrálna práca	5	20
Body za otázky	0	5
Mimoriadne aktivity	0	10
Spolu	25	65

Dochádzka

- prednášky – nepovinné
- cvičenia – nepovinné
 - POZOR! mizivá šanca spraviť predmet bez navštevovania cvičení (skúsenosti)
 - nemožnosť získavať bonusové body za mimoriadne aktivity
 - viacnásobná absencia = vylúčenie z cvičenia

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
 - podmiennečné vylúčenie zo štúdia
 - vylúčenie zo štúdia

1

Princípy objektového programovania

Veci okolo nás



Televízor – ako sa používa?



+ návod na použitie



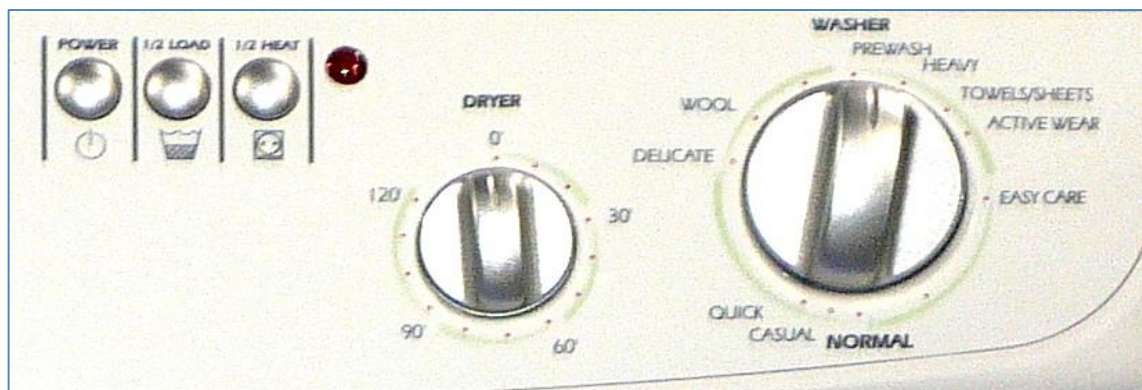
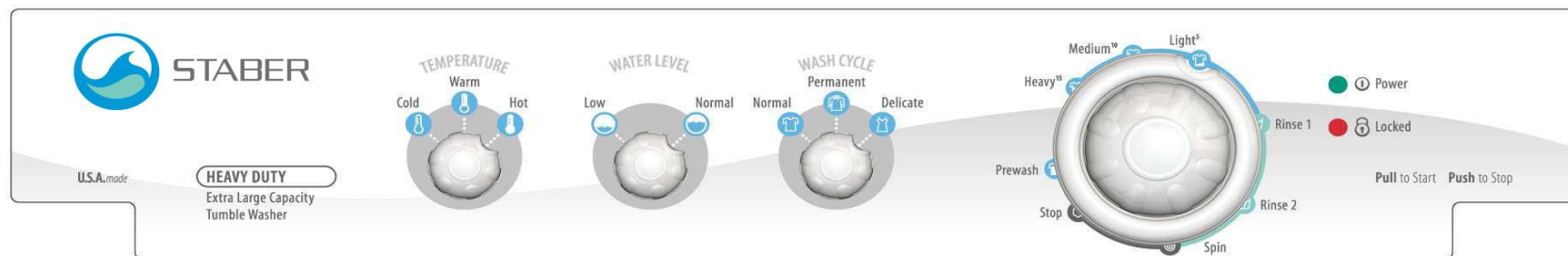
Televízor – čo je vo vnútri?

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

Veci okolo nás



Práčka – ako sa používa?



Veci okolo nás



Veci okolo nás



Veci okolo nás



Veci okolo nás



Veci okolo nás



Veci okolo nás



Veci okolo nás

- dajú sa nejako spoločne charakterizovať?
- čo sa dá povedať, aby sme to mohli aplikovať na každú vec?

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),
 - plazma
- ...

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
- ...

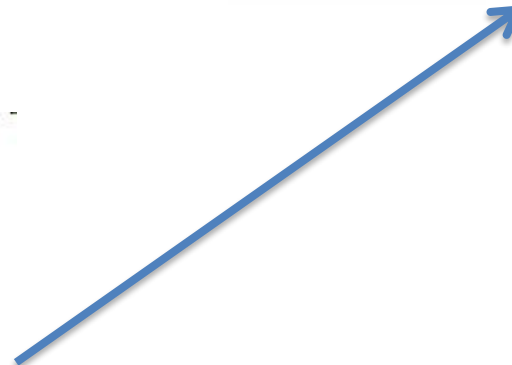
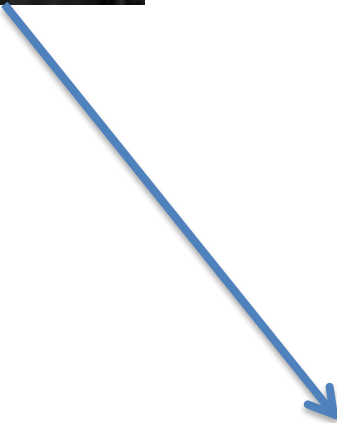
Práčka má funkcie

- 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 ostaná 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

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 píšú v programovacích jazykoch
- my budeme používať jazyk Java [džava]

Postup pri tvorbe programu

- objekt reálneho sveta



- logický model – UML



- softvérový model – program v Java



Reálny objekt



AutomatMHD

- cenaListka: int
- vlozenaCiastka: int
- trzba: int

- + AutomatMHD(cenaListka: int)
- + getCenaListka(): int
- + getVlozenaCiastka(): int
- + vložMincu(hodnotaMince: int): void
- + tlačListok(): void

Softvérový model

```
/**  
 * Konstruktor vytvori automat, ktory bude  
 * tlacit cestovne listky pevnej ceny.  
 * Cena je urcena parametrom cenaListka.  
 * Pozor - cena listka musi byt kladne cele  
 * cislo a tato podmienka sa nekontroluje.  
 */  
public AutomatMHD(int cenaListka) {  
    this.cenaListka = cenaListka;  
    this.vlozenaCiastka = 0;  
    this.trzba = 0;  
}
```

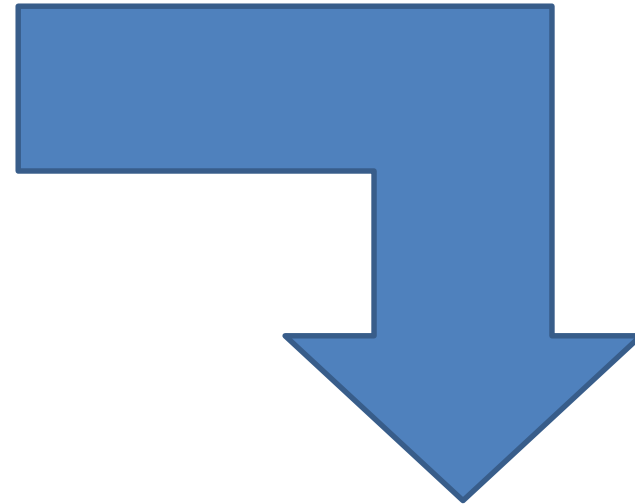

Používanie programu

- 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

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

preklad

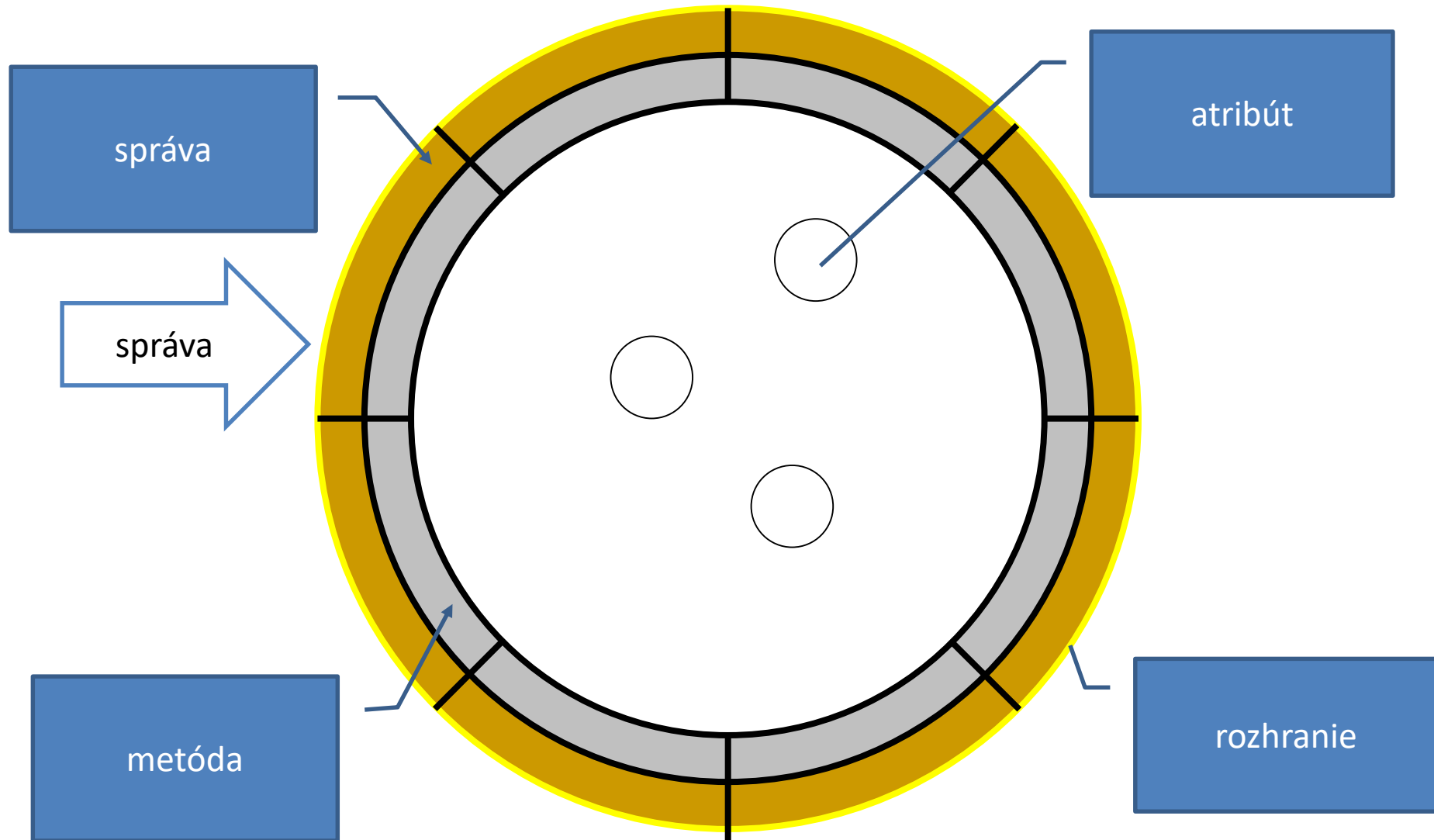


000000A0:	65	72	54	61	62	6C	65	01	00	12	4C	6F	63	61	6C	56		erTable■■■■LocalU
000000B0:	61	72	69	61	62	6C	65	54	61	62	6C	65	01	00	04	74		variableTable■■■t
000000C0:	68	69	73	01	00	0C	4C	41	75	74	6F	6D	61	74	4D	48		his■■■■LAutomatMH
000000D0:	44	3B	01	00	0C	70	61	43	65	6E	61	4C	69	73	74	6B		D;■■■paCenaListk
000000E0:	61	01	00	0D	64	61	6A	43	65	6E	75	4C	69	73	74	6B		a■■■dajCenuListk
000000F0:	61	01	00	03	28	29	49	01	00	11	64	61	6A	56	6C	6F		a■■■()I■■■dajVlo
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		F■■■vlozMincu■■■
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		U■■■tlacListka■■■
00000150:	0A	53	6F	75	72	63	65	46	69	6C	65	01	00	0F	41	75		■SourceFile■■■Au
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

„Každý hmotný bod v inerciálnej sústave zotrváva v pokoji alebo v rovnomernom priamočiarom pohybe, kým nie je nútený vonkajšími silami tento svoj stav zmeniť.“

Isaac Newton

- 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

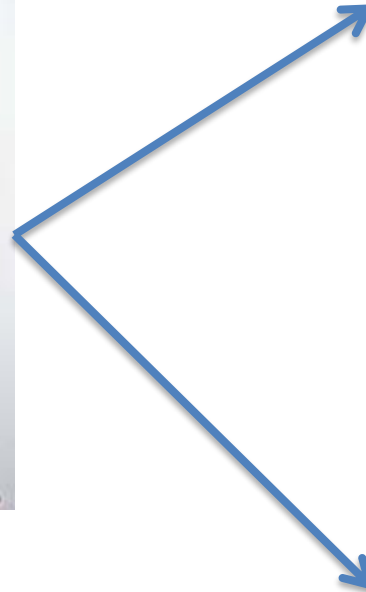


Inštancia

Objekt



Inštancia



Objekt, trieda a inštancia

- továreň – objekt, ktorý vyrába iné objekty – výrobky
 - (často rovnako pomenovane ako továreň)
- trieda – objekt, ktorý vytvára rovnaké objekty
- inštancia – objekt, ktorý vyrobila trieda
- príklad:
 - škoda
 - trieda na auta značky Škoda
 - škoda_ZA765CK, škoda_KM583AZ
 - inštancie triedy Škoda

Trieda ako továrneň

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

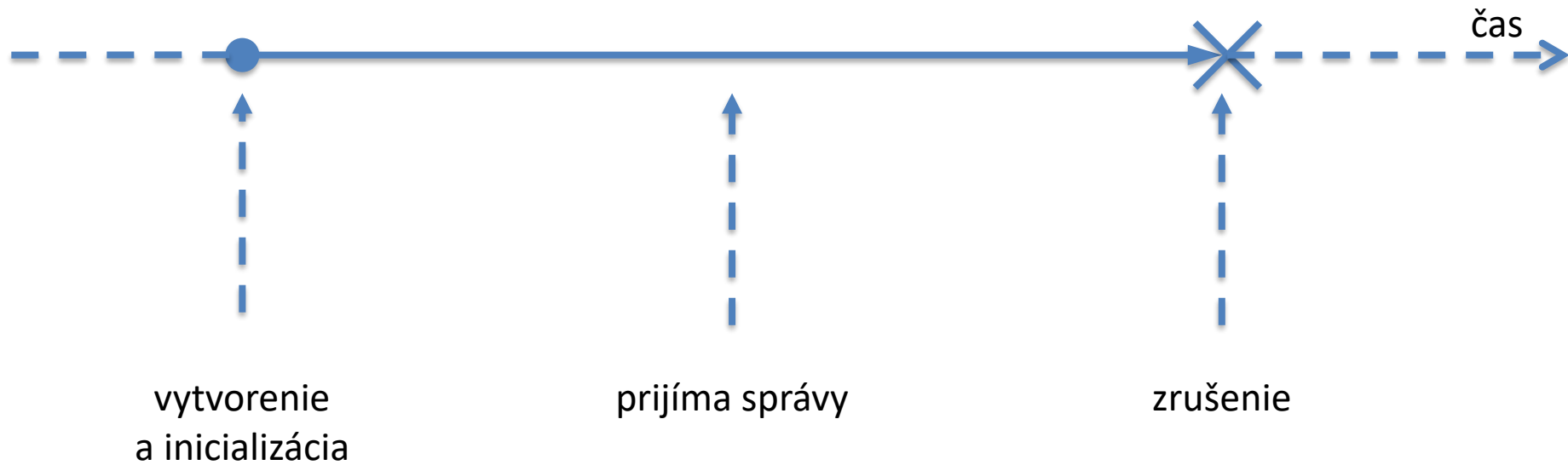
Trieda ako typ

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

Životný cyklus inštancie

- príklad:
 - továreň 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

Identita objektu

- identita objektu – vyjadruje vnútornú vlastnosť objektu, fakt jeho existencie a jednoznačnú odlíšiteľnosť od iných akokoľvek podobných objektov
- na vyjadrenie svojej identity objekty používajú jednoznačnú identifikáciu – jednoznačné vonkajšie označenie určitého objektu, pomocou ktorého ho dokážu odlíšiť od všetkých podobných
 - Môžeme modelovať pomocou atribútu

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 na rozdiel od identity 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

- 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 <http://vzdelavanie.uniza.sk/>

Vďaka za pozornosť