



Algoritmus

Informatika 1

Z prednášky

- Čo je algoritmus a aké sú jeho základné vlastnosti?
- Čo je lokálna premenná?
- Aký je rozdiel medzi atribútom a lokálnou premennou?

Cieľ cvičenia

- Práca s lokálnymi premennými a logickými výrazmi.
- Príkaz vetvenia `if`.
- Návrh vlastnej triedy.

Lokálna premenná

- Syntax:

`typ_premennej` názov;

- Platí v rámci bloku (medzi { }), kde je definovaná.
- Môže byť hneď inicializovaná, napríklad:

`int` pom = 13;

Relačné a logické výrazy

- Relačné výrazy: $<$, $<=$, $>$, $>=$, $==$, $!=$.
- **Je rozdiel medzi prirad'ovacím príkazom $=$ a testom na rovnosť $==$!!!**
- Logické výrazy: $\&\&$, $||$, $!$

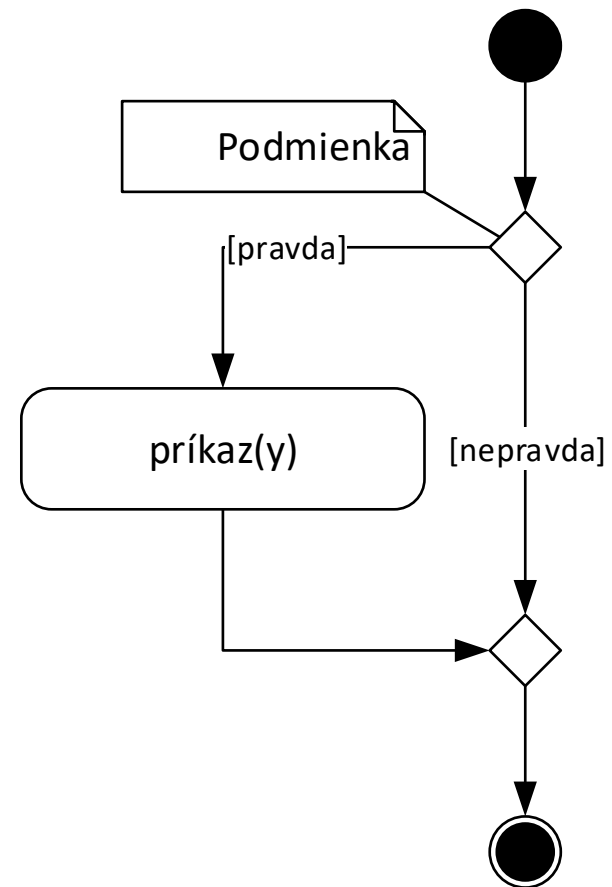
Logický súčin		
A	A && B	B
1	1	1
1	0	0
0	0	1
0	0	0

Logický súčet		
A	A B	B
1	1	1
1	1	0
0	1	1
0	0	0

Negácia	
A	!A
1	0
0	1

Vetvenie – neúplný príkaz `if`

```
if (logický_výraz) {  
    // vykoná sa,  
    // ak podmienka platí  
}
```

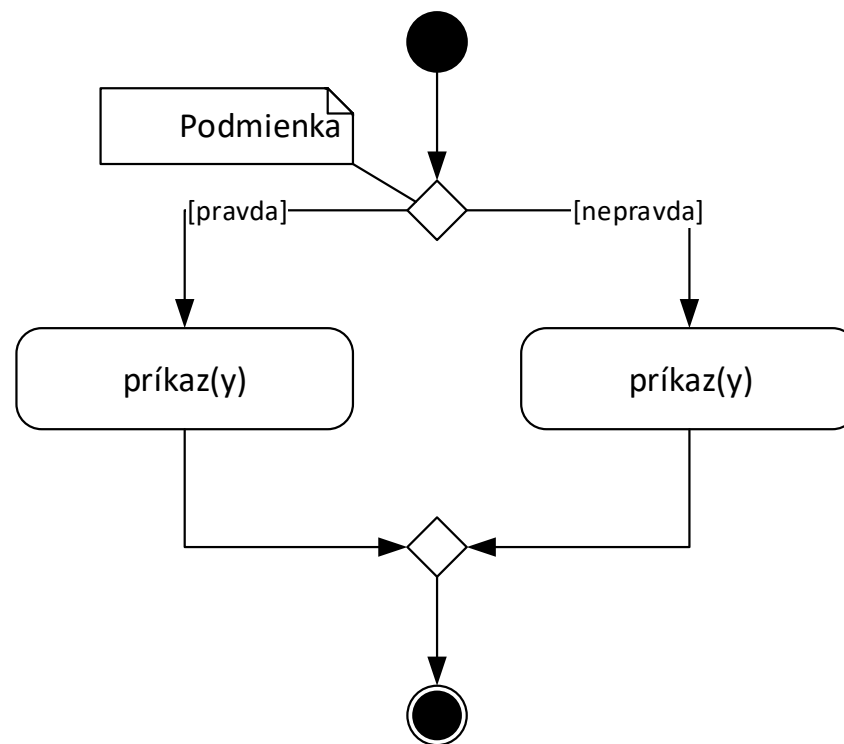


Vetvenie – neúplný príkaz `if`

- Napíšte metódu, ktorá preberie tri parametre a vypíše do terminálu hodnotu rozdielu prvých dvoch parametrov, ak je tento rozdiel aspoň veľkosti tretieho parametra.

Vetvenie – úplný príkaz `if`

```
if (logický_výraz) {  
    // vykoná sa,  
    // ak podmienka platí  
} else {  
    // vykoná sa,  
    // ak podmienka neplatí  
}
```



Vetvenie – úplný príkaz `if`

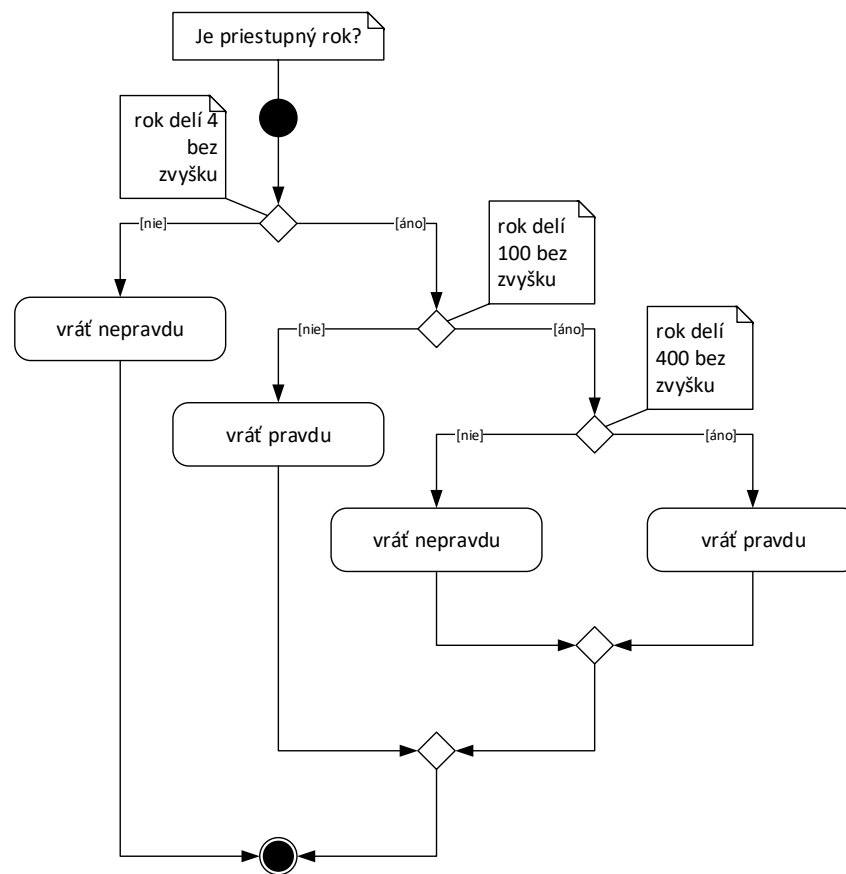
- Napíšte metódu, ktorá vráti najväčšie z troch čísel, ktoré preberie ako parametre.

Projekt datum

- Navrhните triedu `Datum`, ktorá bude reprezentovať dátum a bude poskytovať nasledujúcu funkčnosť:
 - Vytvorí sa v zadaný dátum (napr. 24.9.2015).
 - Vie vrátiť a nastaviť svoj deň, mesiac a rok.
 - Vie zistiť, či sa nachádza v priestupnom roku.
 - Vie zistiť počet dní v aktuálnom mesiaci.
 - Vie vrátiť počet dní, koľko uplynulo od začiatku roku po daný dátum (seba).
 - Vie zistiť, koľko dní ostáva do konca roku.
 - Vie pričítať a odčítať od seba počet dní a na základe toho upraviť dátum (napr. 24.12.2015 + 10 = 3.1.2016).
- Vytvorte UML diagram tried (aj vnútorný pohľad) pre triedu `Datum`.
- Implementujte konštruktor, gettre a settre.

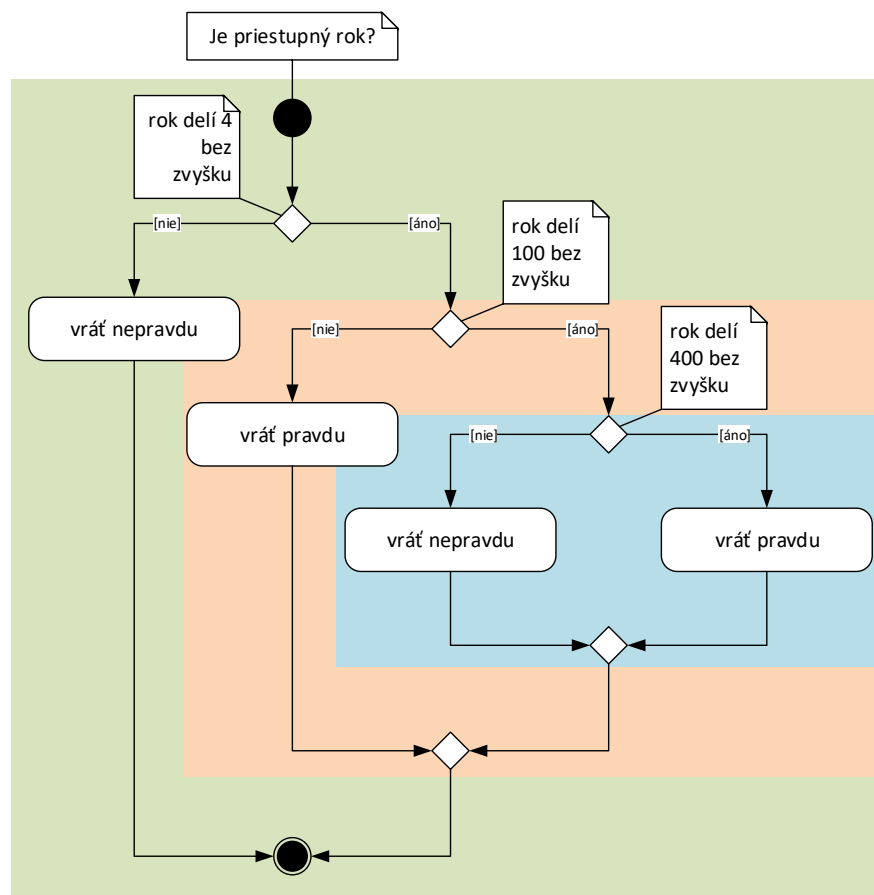
Test na priestupný rok

- Rok je priestupný vtedy, keď je deliteľný 4 bez zvyšku.
- Ak je deliteľný 100 bez zvyšku, tak musí byť aj 400, inak nie je priestupný.
- Ako budú vyzerat' kódy jednotlivých podmienok?



Test na priestupný rok

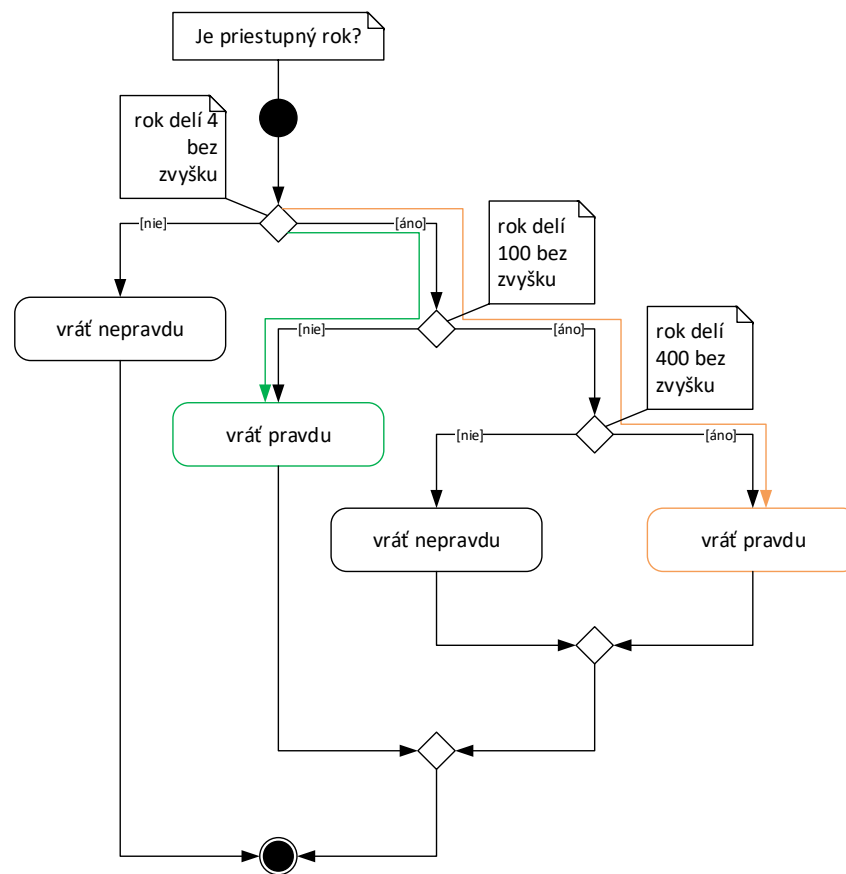
```
public boolean jePriestupny() {  
    if (this.rok % 4 == 0)  
        if (this.rok % 100 == 0)  
            if (this.rok % 400 == 0)  
                return true;  
            else  
                return false;  
        else  
            return true;  
    else  
        return false;  
}
```



Test na priestupný rok

- Existujú dve cesty k pravde; ak je rok priestupný, musí platiť aspoň jedna z nich.

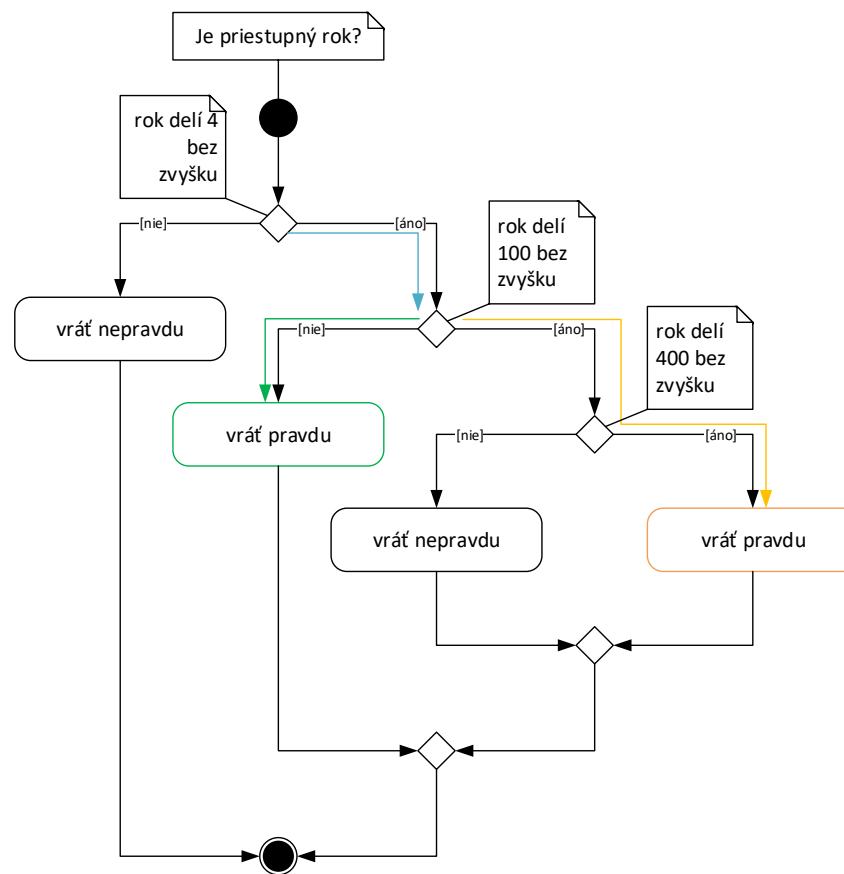
```
public boolean jePriestupny() {  
    if (( this.rok % 4 == 0 &&  
         this.rok % 100 != 0 )  
        || // alebo  
        (this.rok % 4 == 0 &&  
         this.rok % 100 == 0 &&  
         this.rok % 400 == 0 ))  
        return true;  
    else  
        return false;  
}
```



Test na priestupný rok

- „Spoločnú cestu“ môžeme vyjadriť logickým súčinom.

```
public boolean jePriestupny() {  
    if ((this.rok % 4 == 0)  
        &&  
        (this.rok % 100 != 0 ||  
         this.rok % 400 == 0))  
        return true;  
    else  
        return false;  
}
```



Test na priestupný rok

```
public boolean jePriestupny() {  
    if ((this.rok % 4 == 0) &&  
        (this.rok % 100 != 0 || this.rok % 400 == 0))  
        return true;  
    else  
        return false;  
}
```

```
public boolean jePriestupny() {  
    return (this.rok % 4 == 0) &&  
        (this.rok % 100 != 0 || this.rok % 400 == 0)  
}
```



Algoritmus

DRUHÁ ČASŤ

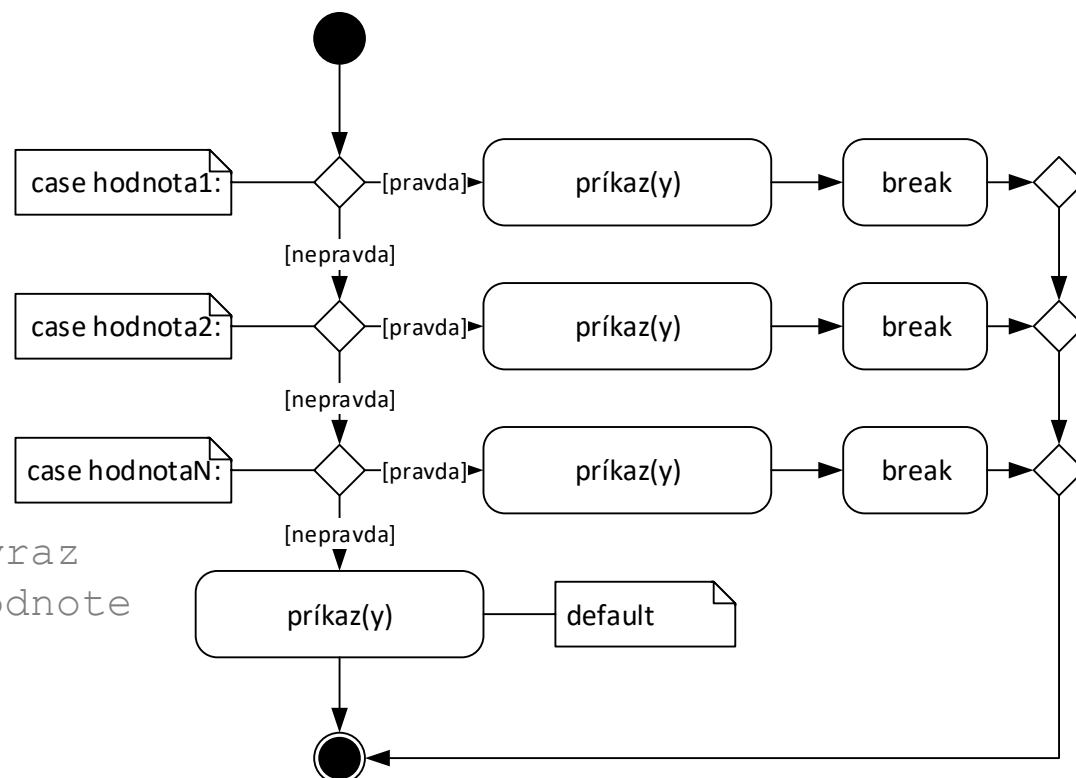


Cieľ cvičenia

- Príkaz vetvenia switch.

Vetvenie –príkaz `switch`

```
switch(výraz) {  
  case hodnota:  
    // vykoná sa, ak  
    // výraz = hodnota  
    break;  
  
  default:  
    // vykoná sa, ak sa výraz  
    // nerovnal žiadnej hodnote  
}
```

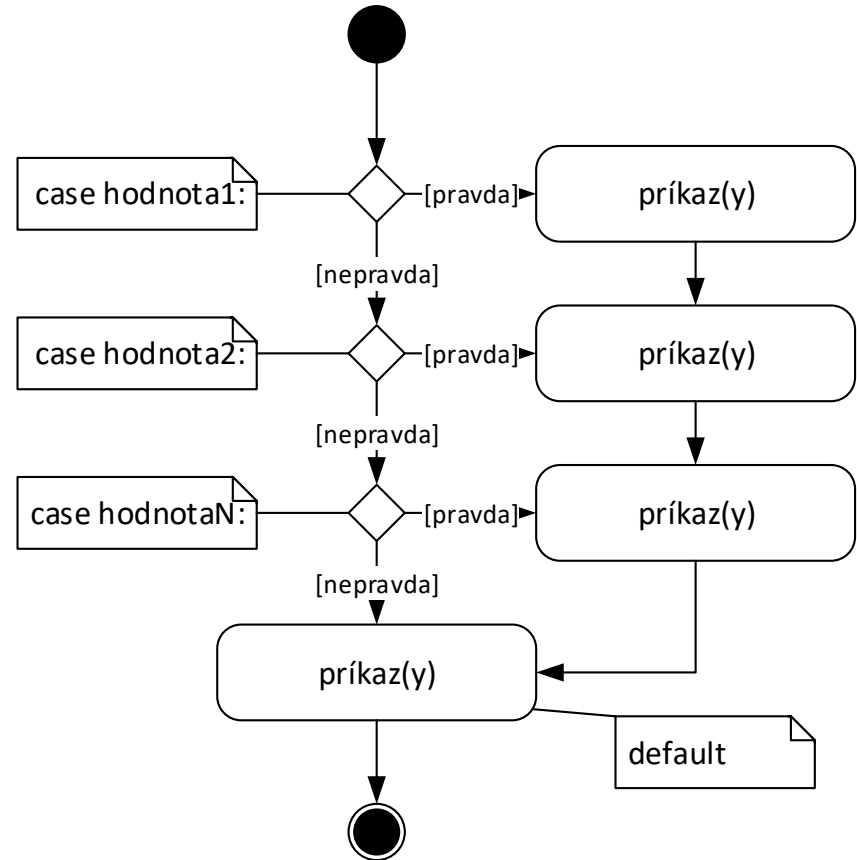


Vetvenie –príkaz `switch`

- Implementujte metódu na získanie počtu dní v aktuálnom mesiaci.

Vetvenie –príkaz `switch` bez príkazu `break`

- Príkaz `break` vyskočí za najbližšiu `}` zátvorku.
- Ak sa `break` neuvedie na konci v `case` návesti, vykonávanie pokračuje ďalej (nevyskočí za `switch`!)
- Používať iba veľmi výnimočne (ideálne iba v `switch`), narúša čitateľnosť kódu.



Vetvenie –príkaz `switch` bez príkazu `break`

- Implementujte metódu na zistenie počtu dní od začiatku roku a počtu dní do konca roku.

Algoritmus

DOMÁCE CVIČENIA

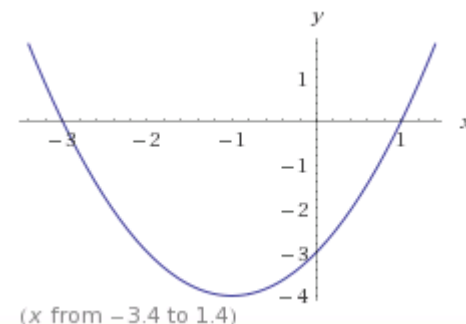


Datum – pričítanie počtu dní

- Napíšte diagram aktivít metódy na navýšenie počtu dní.
- Implementujte riešenie s využitím **doteraz známych** príkazov.
- Otestujte metódu na rôznych príkladoch (prejdenie cez mesiac, rok, viac rokov).

Kvadratická rovnica

- Vašou úlohou je navrhnuť, namodelovať a implementovať triedu `KvadratickaRovnica`, ktorá bude schopná riešiť kvadratické rovnice v tvare **$ax^2 + bx + c = 0$** , pričom hodnoty a , b , c preberie inštancia ako parametre pri svojom vzniku.
- `KvadratickaRovnica` bude vedieť vrátiť a zmeniť hodnoty a , b , c (ak by príslušný setter spôsobil, že sa prestane jednať o kvadratickú rovnicu, tak nič nenastaví a informuje používateľa výpisom chyby v terminály).
- `KvadratickaRovnica` musí vedieť reagovať na nasledujúce správy:
 - Vrátiť x -ovú súradnicu vrcholu paraboly.
 - Vrátiť y -ovú súradnicu vrcholu paraboly.
 - Vyriešiť rovnicu, teda vypočítať a vrátiť hodnotu koreňov, pričom:
 - ak sa pri výpočte zistí, že zadaná kvadratická rovnica nemá riešenie, vypíše sa táto správa do terminálu,
 - ak existujú dva korene rovnice, vráti sa koreň, ktorý je bližšie k 0.
 - Otestovať, či dané Y patrí do oboru hodnôt rovnice.



Upozornenie

- Tieto študijné materiály sú určené výhradne pre študentov predmetu 5BI137 Informatika 1 na Fakulte riadenia a informatiky Žilinskej univerzity v Žiline.
- Reprodukovanie, šírenie (i častí) materiálov bez písomného súhlasu autora nie je dovolené.

Ing. Michal Varga, PhD.
Katedra informatiky
Fakulta riadenia a informatiky
Žilinská univerzita v Žiline
Michal.Varga@fri.uniza.sk

