



Pracovný list 3

Kód jazyka JAVA - Editor zdrojového kódu. Vytvorenie tried, vzťahy medzi triedami.

Úloha 1 – Zobrazenie kódu iných tried

Zobrazte kód už existujúcich tried, prezrite si ho a pýtajte sa

Úloha 2 – Skopírovanie existujúceho kódu do triedy *Cinka1*

Kód tried triedy skopírujeme a dáme ho do triedy *Cinka1* v prostredí BLUEJ. Kódy triedy *Cinka1* sú aj v súbore : *Cinka1.txt*.

Po doplnení kódu je potrebné triedu skompilovať – preložiť (cez tlačidlo *Preložiť*)

Pre činku, môžeme doplniť nasledujúci kód: (kód je popísaný poznámkami)

```

/*****
*
* Trieda {@code Cinka1} slúži na zobrazenie cinky...
* @author      {kabetka} --> POZNÁMKY - meno autora a verzia
* @version     {verzia: 1.00}
*****
*/
public class Cinka1 //--> ZAČIATOK DEFINOVANIA TRIEDY
{
    /* ***** Definovanie atribútov ***** */
    // Rozmery cinky sú dané ako atribúty triedy --> POZNÁMKY
    private int poziciaX, poziciaY; //--> ATRIBÚTY
    // Atribút definuje zobrazenie - či je činka zobrazená
    private boolean jeZobrazeny; //--> ATRIBÚT
    // Cinka sa sklada z dvoch elíps - gulaL - ľavá, gulaP - pravá
    a nakoniec ich spája zrdka - obdĺžnik
    private Elipsa gulaL; //--> ATRIBÚTY ELIPSA, OBDLZNIK
    private Elipsa gulaP;
    private Obdlznik zrdka;

/*****
*
* Konštruktory objektov triedy Cinka1 // --> POZNÁMKY
*/
//Konštruktor s parametrami

public Cinka1(int poziciaX, int poziciaY )//--> KONŠTRUKTOR s
parametrami
{

```

Príloha E

```
/* Tu definujeme činku z dvoch elíps a pripíšeme vzdialenosť pozíciuX
a pozíciuY a k nim obdlžnik */
//Vytvorí sa objekty ELIPSA a OBDLZNIK

    this.poziciaX = poziciaX;  this.poziciaY = poziciaY;
    gulaL = new Elipsa ( poziciaX + 0, poziciaY + 0, 50, 50, 1
);
    zrdka = new Obdlznik( poziciaX + 25, poziciaY + 23, 150, 4, 4
);
    gulaP = new Elipsa ( poziciaX + 150, poziciaY + 0, 50, 50, 1
);
    jeZobrazeny = false;
}

// Konštruktor bez parametrov
public Cinka1() // --> KONŠTRUKTOR bez parametrov
{
    /* Konštruktor je taký istý ako s parametrami - to sa vyjadrilo
príkazom this, ale má pozíciu 0,0, teda činka bude v ľavom hornom rohu.
*/
    this( 0, 0 );
}

// Metoda na zmenu pozície cinky, posuva sa po osi X a Y
public void zmenaPozicie( int zmenaX, int zmenaY )// --> METÓDA 1

{
    /* je to podobne ako pri konštruktore pozícia X a Y sa posunie
o zmenaX a zmenaY */

    poziciaX += zmenaX;  poziciaY += zmenaY;
    zrdka.zmenaPozicie( zmenaX, zmenaY );
    gulaL.zmenaPozicie( zmenaX, zmenaY );
    gulaP.zmenaPozicie( zmenaX, zmenaY );
}

// Metoda na zobrazenie cinky

public void zobraz()//--> METÓDA 2
{
    /* atribút jeZobrazeny je nastavený na true - pravda
zobrazí sa žrdka a pravá a ľavá guľa cez metódy ktoré má trieda
ELIPSA a trieda OBDLZNIK */
    jeZobrazeny = true;
    zrdka.zobraz();
    gulaL.zobraz();
    gulaP.zobraz();
}

// Metoda na ukrytie cinky
public void skry() //--> METÓDA 3
{
    /* atribút jeZobrazeny je nastavený na false - nepravda
// skryje sa žrdka a pravá a ľavá guľa cez metódy ktoré má trieda
ELIPSA a trieda OBDLZNIK
```

Príloha E

```
    jeZobrazeny = false;  
    zrdka.skry();  
    gulaL.skry();  
    gulaP.skry();  
}
```

```
} //koniec triedy
```

Úloha 3 - Vytvorenie objektov z triedy *Cinka 1*

Vytvorte samostatne dva objekty novej triedy *Cinka1*. Jeden objekt je bez parametrov, vytvorí ho cez konštruktor `new Cinka1()` a druhý je s parametrami `new Cinka1(int poziciaX, int poziciaY)`. Zobrazte „činku“ prostredníctvom metódy `zobraz()`. Vyskúšaj aj jej ďalšie aj zdedené metódy.

