

Metodický list 2 – Základné pojmy OOP v jazyku JAVA



Príloha B

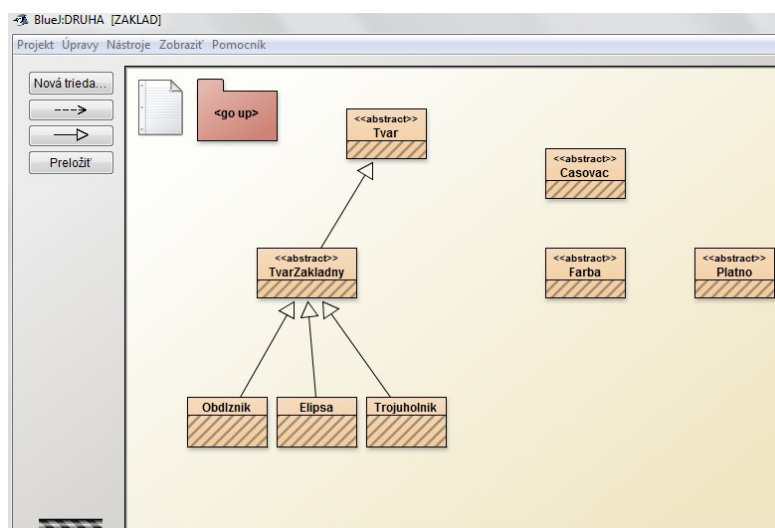
<p>Názov témy: Kód jazyka JAVA - Editor zdrojového kódu. Vytvorenie tried, vzťahy medzi triedami.</p>	
Cieľová skupina:	- 2.-3 ročník gymnázií a stredných škôl
Predmet:	- informatika
Ciele:	- vzdelávací cieľ - Obsah objektu, vytvorenie novej triedy, zdrojový kód, pojem dedičnosť – vzťah medzi triedami.(oboznámiť sa iba s pojmom, viac nevysvetľovať) - výchovný cieľ – tvorivosť a použitie vedomostí v praktických príkladoch
Organizačné formy:	- hodina osvojovania nových vedomostí, využívania vedomostí a zručností v praxi, - Práca žiakov – individuálna, frontálna
Organizačné metódy:	<ul style="list-style-type: none"> ▪ Vysvetľovanie s názornými ukážkami s prácou v prostredí BLUEJ s použitím hotového príkladu. ▪ Práca žiakov – individuálna, frontálna (s učiteľom) a skupinová (vo dvojiciach) – je vhodné ak si študenti navzájom skúšajú jednotlivé príklady.
Popis:	<ul style="list-style-type: none"> ▪ Úvodná motivácia – rozhovor o objektoch, uložení údajov, programovaní. ▪ Oboznámiť sa uložením dát v objekte v prostredí BLUEJ ▪ Základy práce so zdrojovým kódom, editor kódu, ako ho nájdeme. ▪ Vytvorenie nových tried vo vzorovom príklade ▪ Pojem dedičnosť – iba ako vzťah medzi triedami (nevysvetľovať dopodrobna) – jej znázornenie. <p>Inovačné formy výučby – pracovný list v e-learningovom prostredí Moodle, práca v prostredí BLUEJ na vývoj programov v objektovo orientovanom jazyku JAVA.</p>
Zadanie pre žiakov:	<p>Podľa pokynov učiteľa a pracovného listu pre žiaka žiaci vykonávajú nasledujúce činnosti (učiteľ môže postup vykonávať priamo na príklade, alebo prostredníctvom prezentácie):</p> <ol style="list-style-type: none"> 1. Študenti otvoria na svojich počítačoch pripravený príklad s názvom ZAKLAD (<i>package.bluej</i>) a skompilujú ho - tlačidlo <i>Preložiť</i>. 2. Vytvoria nový objekt napríklad <i>Obdlznik 1</i> a prostredníctvom kliknutia pravého tlačidla na objekte (červený obdĺžnik) po zobrazení menu vyberú možnosť <i>Prehliadať</i> 3. Študenti môžu vytvoriť ďalšie objekty aj z inej triedy a môžu ich prezrieť 4. Študenti podľa pokynov a naznačeného postupu vytvárajú novú triedu <i>Cinka1</i> (prostredie Bluej, stlačenie pravého tlačidla myši a výber menu <i>Nová trieda</i> a vyberú <i>Štandardná trieda</i>) 5. Zobrazia zdrojový kód triedy <i>Cinka1</i>, zatiaľ prázdny (Pravé tlačidlo myši na triede <i>Cinka1</i> a výber <i>Editor kódu</i>) 6. Úloha 1 – Študenti prezerajú kód ostatných vytvorených tried v príklade ZAKLAD 7. Úloha 2 – Po vysvetlení kódu študenti sami doplnia pripravený kód jazyka JAVA – <i>Cinka1.txt</i> 8. Úloha 3 – Študenti po vytvorení triedy <i>Cinka 1</i> vytvoria objekty tejto triedy a vyskúšajú ju zobrazíť – metóda <i>zobraz()</i>. Vyskúšajú aj ďalšie metódy.
Príprava, učebné pomôcky:	Nainštalované vývojové prostredie BLUEJ. Hotový príklad pre vysvetľovanie základných pojmov OOP – triedy s metódami a s možnosťou ich zobrazenia, Pracovný list 1,2.
Metodický postup:	<p>3. Téma (2 vyučovacie hodiny): Kód jazyka JAVA – editor zdrojového kódu v prostredí BLUEJ</p> <p>Úvod: Učiteľ uvedie na začiatku: Naučili sme sa čo je to objekt, vytvoriť objekt z existujúcej triedy, ale predsa ako to vyzerá vo vnútri objektu keď zadáme atribúty? A kedy sa bude programovať? Je tu zdrojový kód?</p> <p>Dnes si vysvetlíme, že bez programovania to nejde a triedy, ktoré sme doteraz využívali musel niekto naprogramovať.</p> <p>Teoretický základ s výkladom a názornými ukážkami:</p>

Prezeranie obsahu objektu

Predtým ako začneme pracovať otvoríme si náš základný hotový príklad v priečinku ZAKLAD a spustíme *package.bluej*. (príklad ZAKLAD je pripravený na túto dvojhodinovku)

package	bluej	3 694	23.01.2015	14:29	-a-
bluej	pkg	3 694	23.01.2015	14:29	-a-
README	TXT	15	23.01.2015	14:27	-a-
Trojuholnik	java	1 827	06.05.2009	11:54	-a-
Platno	java	4 443	23.04.2009	12:06	-a-
Casovac	java	241	08.04.2009	16:23	-a-
Tvar	java	1 471	08.04.2009	11:33	-a-
Elipsa	java	1 483	08.04.2009	10:36	-a-
Obdlznik	java	1 487	08.04.2009	10:35	-a-
TvarZakladny	java	1 348	08.04.2009	10:35	-a-
Farba	java	875	08.04.2009	10:35	-a-

Otvoríme si pôvodný príklad s názvom ZAKLAD :



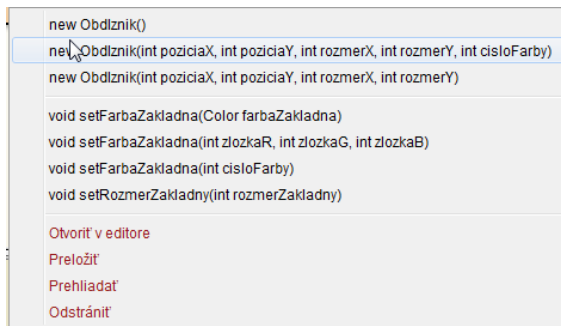
Obr. Základný príklad

Po otvorení ho skompilujeme prostredníctvom tlačidla resp. *Compile*.

Vytvoríme nové objekty (názvy objektov si môžu študenti ľubovoľne zvoliť) :

- cez konštruktor `new Obdlznik()` bez parametrov
- cez konštruktor `new Obdlznik(int poziciaX, int poziciaY, int rozmerX, int rozmerY, int cisloFarby)` je to konštruktor s parametrami a je potrebné zadať aj hodnoty týchto parametrov – rozmerY a farba.

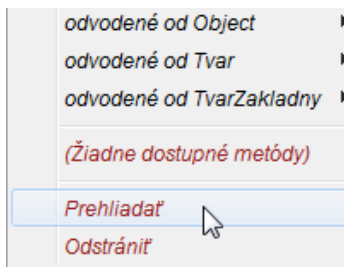




Obr. Voľba nového objektu

Nastavené rozmery objektu „Obdlznik1“ môžeme **Prehliadať**, resp. uloženie v pamäti aj prostredníctvom prostredia BLUEJ:

1. Klikneme na objekt pravým tlačidlom myši
2. Z menu vyberieme voľbu Prehliadať



Obr. Výber prehliadania objektu

3. Zobrazia sa jednotlivé hodnoty atribútov objektu – inštancie



Obr. Prehliadanie atribútov objektov *Obdlznik1* a *Obdlznik 2*

Na obrázku vidíme nastavené atribúty objektu *Obdlznik1* a atribúty, ktoré sme si nastavili sami v objekte *Obdlznik2*

Ďalej sa budeme venovať vytváraniu novej triedy odvodenej z existujúcich tried

Vytvorenie novej triedy napríklad „Činka1“

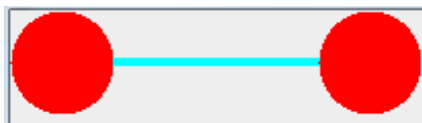
Popis:

Našou ďalšou úlohou bude vytvorenie novej triedy vlastnej triedy na príklade.

Chceme vytvoriť a graficky znázorniť objekt, ktorý vznikne zložením objektov tried, ktoré už máme v základnom príklade. Vybrali sme si „činku“.

Ak si predstavíme činku mohla by vyzeráť tak ako ju vidíme na obrázku 14 (prípadne s inou veľkosťou a farbou).

Činka sa skladá z dvoch kruhov, resp. elips a z jednej tzv. žrdky – t. z. úchytu činky, ktorá môže byť nahradená obdĺžnikom.

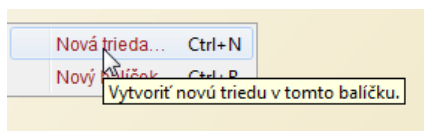


Obr. Príklad znázornenej činky

Ak chceme vytvárať objekty „činka“ tak potrebujeme vytvoriť triedu *Cinka*. Z predchádzajúceho popisu je zrejmé, že na budúce objekty triedy *Cinka* potrebujeme dve elipsy – kruhy – **dva objekty triedy *Elipsa* a jeden obdĺžnik - jeden objekt triedy *Obdlznik*.**

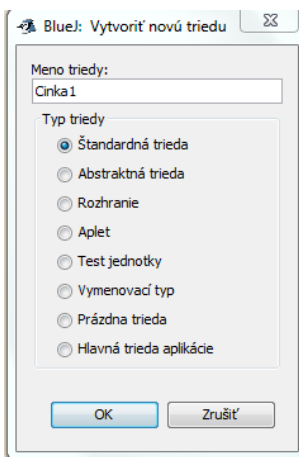
Vytvorenie novej triedy *Cinka1* v prostredí BLUEJ:

Stlačíme tlačidlo **Nová trieda...** v menu prostredia BLUEJ, alebo klikneme pravým tlačidlom na voľnej ploche prostredia BLUEJ a zobrazí sa menu, ktoré má v ponuke vytvorenie novej triedy. (Nasledujúci Obr.).



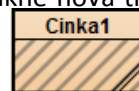
Obr. Vytvorenie novej triedy

1. Zobrazí sa ďalšie menu na vytvorenie triedy. Vyplníme *Meno triedy* - ***Cinka1***. Názov triedy je vždy s prvým veľkým písmenom. Vo voľbe typ triedy pre tento prípad vyberieme *Štandardná trieda*.



Obr. Menu na vytvorenie triedy

2. Potvrdíme vytvorenie triedy tlačidlom *OK*. Vznikne nová trieda ***Cinka1*** a graficky sa znázorní na ploche prostredia



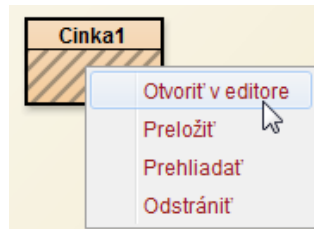
3. Trieda obsahuje „vzorový kód“ v jazyku JAVA, ale na vytvorenie novej triedy je potrebné doplniť zdrojový kód triedy v jazyku JAVA.

Ako zobrazíme zdrojový kód?

Zdrojový kód tried, triedy - zobrazenie

Zdrojový kód môžeme zobraziť pri každej triede prostredníctvom *Editora zdrojového kódu*. *Editor zdrojového kódu* v prostredí BLUEJ zobrazíme nasledujúcim spôsobom :

1. Klikneme pravým tlačidlom na triedu
2. Vyberieme voľbu *Otvoriť v editore* a otvorí sa *Editor kódu* (Obr.)



Obr. Otvorenie zdrojového kódu triedy

Pre novú triedu **Cinka1** sa otvorí zdrojový kód novej triedy. (Obr. 19)

```

/*****
 * Trieda (@code Cinka1) slúži ... - doplňte opis účelu triedy
 *
 * @author      {meno autora}
 * @version     {verzia, napr.: 0.00.000, prípadne dátum}
 */
public class Cinka1
{
    // inštančné premenné - nahraďte nižšie uvedený príklad svojimi
    // vlastnými premennými
    private int x;

    /*****
     * Konštruktory objektov triedy Cinka1
     */
    public Cinka1()
    {
        // inicializujte inštančné premenné
        x = 0;
    }

    /*****
     * Prázdna metóda - tento komentár nahraďte svojim opisom
     *
     * @param    y    opis významu parametra
     * @return   opis významu návratovej hodnoty - napr.: súčet x a y
     */
    public int príkladMetódy(int y)
    {
        // Sem vložte kód
        return x + y;
    }
}

```

Obr. 19 Zdrojový kód novej triedy **Cinka1**

Prostredie BLUEJ vytvorí novú triedu a hneď vytvorí aj v editore kódu štruktúru kódu, kde budeme môcť dopísať konkrétny zdrojový kód v jazyku JAVA. Tento kód novej triedy je prázdny, obsahuje iba príklad atribútov triedy, konštruktora triedy a jednej metódy triedy.

Ak chceme novú triedu naprogramovať, tak aby nám zobrazila na zobrazovacom plátne

činku musíme zapísať rozumný zdrojový kód do vytvorenej kódovej štruktúry. Zatiaľ nepoznáme programovací jazyk JAVA, ani jeho príkazy a syntax, pomôžeme si hotovým zdrojovým kódom a naučíme sa v kóde rozlišovať jednotlivé časti triedy: atribúty, konštruktor, metódy. Pre triedu *Cinka1* sa nám vygeneroval takýto prázdny zdrojový kód :

```

/*****
 * Trieda {@code Cinka1} slúži ... - doplňte opis účelu triedy -->
POZNÁMKY
 * @author      {meno autora}
 * @version     {verzia, napr.: 0.00.000, prípadne dátum}
 */
public class Cinka1 --> ZAČIATOK kódu triedy
{
    // inštančné premenné - nahraďte nižšie uvedený príklad svojími
    // vlastnými premennými
    private int x; --> ATRIBÚTY triedy
/*****
 * Konštruktory objektov triedy Cinka1 --> POZNÁMKY
 */
    public Cinka1()--> KONŠTRUKTOR triedy (je ich viac)
    {
        // inicializujte inštančné premenné
        x = 0;
    }

/*****
 * Prázdna metóda - tento komentár nahraďte svojím opisom
 *
 * @param      y      opis významu parametra --> POZNÁMKY
 * @return     opis významu návratovej hodnoty - napr.: súčet x a y
 */
    public int príkladMetody(int y) --> METÓDA triedy (je ich viac)
    {
        // Sem vložte kód
        return x + y;
    }
} //--> KONIEC kódu triedy

```

Vysvetlíme si jednotlivé jeho časti – v kóde sú označené šípkou a za ňou je pomenovanie, napr. ZAČIATOK KÓDU TRIEDY, POZNÁMKY a pod.

Trochu teórie do kódu v jazyku JAVA – vysvetlenie:

Kód triedy začína slovom *public class* za ktorým je názov triedy ***Cinka1*** – toto nazývame **hlavička triedy**. (je označená v predchádzajúcom zdrojovom kóde)

Potom nasleduje zložená zátvorka – ľavá { a pravá zátvorka - } je na konci definície kódu triedy, po všetkých atribútoch, konštruktoroch a metódach, teda na konci. Je označená slovami KONIEC KÓDU TRIEDY.

Po hlavičke triedy sa zapisujú **do kódu atribúty triedy** (sú označené v predchádzajúcom zdrojovom kóde **ATRIBÚTY TRIEDY**) a ich typy, čo vyjadruje aké údaje – dáta budeme ukladať v triede, napríklad sú **textové** – **string**, alebo **číselné** – **celé čísla** sú napr. **int**, **reálne čísla s desatinnou čiarkou** sú **float** a **pod**. (typom premenných a sa budeme venovať

neskôr). Atribúty triedy majú pred svojim názvom väčšinou aj slovo **private** (súkromný) to znamená, že nie sú prístupné iným objektom, alebo triedam.

Konštruktor tried – ako sme si už vysvetlili - každá trieda musí mať aspoň jeden konštruktor aby mohli z nej vznikáť objekty. Slúži na vytvorenie objektov z triedy. Názov konšuktora je rovnaký ako názov triedy, teda pre **našu triedu Cinka1** bude mať konštruktor názov **public Cinka1()** a kód, ktorý potom nasleduje je rovnako v zložených zátvorkách { }.

Slovo public vyjadruje, že konštruktor je verejne prístupný – t. z. že každý používateľ môže z tejto triedy vytvoriť objekt, môže ho použiť. **Toto slovo public musí mať každý konštruktor.**

V triede definujeme väčšinou viac konštruktorov – jeden je bez parametrov (je iba jeden základný, ktorý sa vytvorí sa hneď pri vytvorení triedy, podobne ako v našom prípade), ďalšie konštruktorov sú s parametrami t. z. že za v zátvorkách za názvom konšuktora sú parametre – premenné. Napríklad vytvoríme ďalší **konštruktor v triede Cinka1** s názvom **public Cinka1 (int poziciaX, int poziciaY) // konštruktor s parametrami. Parametrami sú int poziciaX, int poziciaY.** Podobne môžeme vytvoriť ďalší konštruktor, ktorý bude mať parametre, ale s inými názvami, alebo počtom.

Metóda triedy, resp. metódy triedy sa zapisujú za konšuktormi (V kóde je zatiaľ iba jedna prázdna metóda). Podobne metódy majú väčšinou **slovo public** ako konštruktor, tak aby všetci užívatelia mohli používať metódy. Príkladom prázdnej metódy, ktorá sa nám vytvorila je metóda **public int príkladMetody(int y)** (hlavička metódy). Kód metódy zapisujeme tiež do zložených zátvoriek { } a nasleduje hneď po hlavičke metódy.

V kóde, ktorý sa vygeneroval sú časti ohraničené časti s lomítkom a hviezdičkami, sú to **poznámky**, ktoré sa neskôr zobrazia aj pri dokumentácii programu. Editor ich zapisuje modrou farbou. **Poznámky do kódu zapisujeme na začiatku /* a konci */** - hviezdičky musia byť iba na začiatku za lomítkom a na konci pred lomítkom (*hviezdičky medzitým použité iba na zvýraznenie poznámok*). Poznámky iba do jedného riadku môžeme zapisovať aj dvojitém lomítkom - //

Precvičenie:

Úloha 1 - pre študentov:

Študentom ukážeme vytvorenie triedy a zobrazujú si kód triedy aj oni na svojich počítačoch. Môžu zobraziť aj kód už existujúcich tried.

Aký kód doplníme do vytvorenej triedy **Cinka1**?

Kód triedy Cinka1 – pridanie zdrojového kódu do editora kódu

Do našej novej triedy **Cinka1** je potrebné pridať zdrojový kód, ktorý by vykreslil činku. Činku vykreslí plátno ale našou úlohou je zostaviť časti z ktorých sa bude činka skladať. Použijeme na to objekty triedy **ELIPSA** (dva) a triedy **OBDLZNIK** (jeden).


```

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
    {
        /* Tu definujeme činku z dvoch elíps a pripíšeme vzdialenosť pozíciux
        a pozíciuy a k nim obdĺžnik */
        //Vytvorila 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();
    guľaL.zobraz();
    guľaP.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

    jeZobrazeny = false;
    zrdka.skry();
    guľaL.skry();
    guľaP.skry();
}
} //koniec triedy

```

Kód tried triedy skopírujeme a dáme ho do triedy *Cinka1* v prostredí BLUEJ. Môžeme to názorne ukázať cez dataprojektor, prípadne plniť aj po jednotlivých častiach.

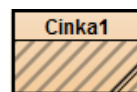
Úloha 2 - pre študentov:

Študenti doplnia – skopírujú kód triedy podľa poznámok a inštrukcií učiteľa. Po doplnení kódu je potrebné triedu skompilovať – preložiť (cez tlačidlo *Preložiť*)

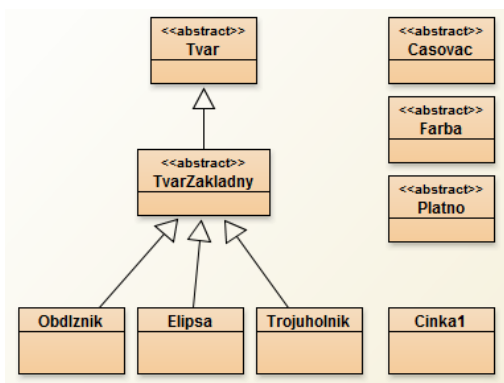
Kódy triedy *Cinka1* sú aj v súbore : ***Cinka1.txt***

Dôležité je študentom vysvetliť a aj individuálne ukazovať, ako si kódom naplnia triedu. Nie všetci študenti sú dostatočne zruční.

Po správnej kompilácii vytvorená trieda nie je už „šráfovaná“



ale má ružovú farbu ako ostatné triedy (Obr.).



Obr. Skompilovaná trieda *Cinka1* a celý projekt

Triedu ***Cinka1*** vyskúšame, tak že vytvoríme z nej objekty.

	<p>Úloha 3 - pre študentov:</p> <p>Študenti vytvoria samostatne dva objekty novej triedy Cinka1. Jeden objekt je bez parametrov, vytvoria ho cez konštruktor <code>new Cinka1()</code> a druhý je s parametrami <code>new Cinka1(int poziciaX, int poziciaY)</code>. Učiteľ im môže pomôcť radou, či ukážkou.</p> <div data-bbox="651 427 1240 862" data-label="Image"> </div> <p>Obr. 21 Zobrazenie objektov triedy Cinka1</p> <p>Záver hodiny: Učiteľ slovne zhodnotí aktivitu žiakov. Zopakuje slovne čo sa naučili.</p>
<p>Hodnotenie: (spätná väzba)</p>	<ul style="list-style-type: none"> - Učiteľ priebežne kontroluje prácu žiakov, pýta sa na výsledky práce, žiaci ich ukazujú po jednotlivých zadaniach. Jednotlivé úlohy je možné aj ohodnotiť – hlavne ak žiaci prichádzajú s novými nápismi, možnosťami, objavmi. Po hodinách je na teoretický základ pripravený aj vedomostný test realizovaný na nasledujúcich hodinách.
<p>Časová dotácia:</p>	<ul style="list-style-type: none"> - 3 vyučovacie hodiny
<p>Prílohy (povinné):</p>	<ul style="list-style-type: none"> - Pracovný list3 pre žiakov v systéme MOODLE - Prezentácia s výkladom učiva, živé ukážky v prostredí BLUEJ - Dataprojektor