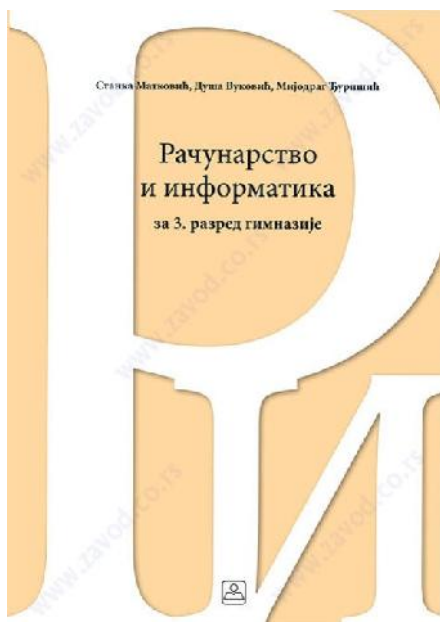


Настава објектно оријентисаног програмирања – креирање класа, добри примери за средњу школу Душа Вуковић (dusavukovic@hotmail.com)



Ученици у основној и средњој школи могу да се једноставно на савремен начин уведу у свет програмирања употребом графичког окружења и модерног објектно оријентисаног језика C#. Кроз креативне примере креирања једноставних апликација које личе на игре, ученици уче да програмирају, али се и упознају са начином рада рачунара. Овакав приступ је одличан увод за озбиљнију тему – самостално креирање класа и употреба њихових објеката у апликацијама.

Данашњи почетници у програмирању не треба да пролазе исти пут којим су програмирање учили старије генерације. То би било исто као када би се у географији прво детаљно изучавала теорија да је земља равна, па тек кроз неколико година се деци презентовало да уствари није. Настава са ученицима који су потпуни почетници у програмирању и са којима треба да се обраде основне теме из програмирања (променљиве, наредбе, гранање, циклуси) може да се организује у модерном окружењу савремених рачунара тако да буде забавна и интересантна ученицима.

У књижарама се ове године појавио једини одобрени уџбеник на тему програмирања који је објавила издавачка кућа **Завод за уџбенике**:

http://zavod.co.rs/index.php?route=product%2Fproduct&filter_name=23171&product_id=4543

Уџбеник прате и видео лекције на Јутјубу:

<http://youtu.be/UbORVl7SbTg>

<http://youtu.be/5CGwGegoT-g>

<http://youtu.be/mQg5wgTbsxs>

Данашње апликације су креиране употребом објеката. Чак је и сама апликација објекат класе *Form*. Објекат обједиљује у једну целину **податке** (атрибуте, својства) и **функционалности** (методе, функције, догађаје, операције). Дефиницијом класе су описани сви објекти те класе. Неко опште правило је да су сви подаци приватни и на тај начин заштићени, док су све функције, тј. методе, јавне и помоћу њих се управља објектом.

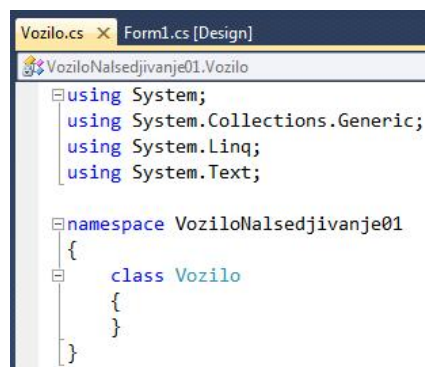
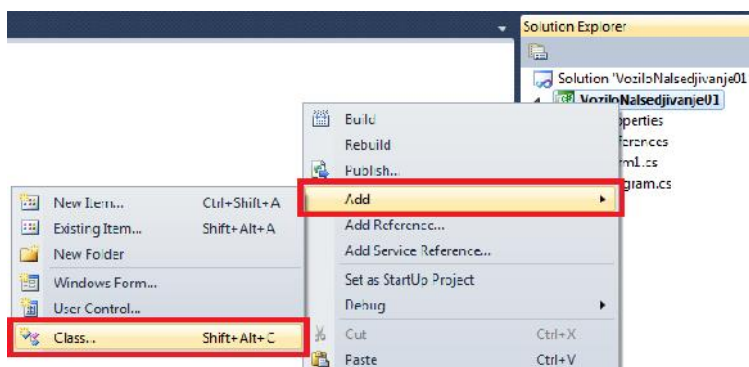
Посматрајмо објекат *timer1* класе *Timer* (часовник).

- Временски интервал је **податак** који нам говори о времену које протекне између два откуцаја часовника (изражено у милисекундама): *timer1.Interval = 1000*;
- Часовник може да се покрене или заустави позивом **метода** (функција): *timer1.Start()*; *timer1.Stop()*;

Права предност објектно оријентисаног програмирања (ООП) се види тек када исту класу употребимо у неколико апликација, зато што се онда далеко смањује количина програмирања у свим следећим апликацијама где користимо објекте једном креиране класе. Други механизам који олакшава програмирање и доводи до мањег простора за грешке је **наслеђивање**. Све оно што је заједничко за неколико класа се пише само једном у надкласи и тиме се избегава непотребно понављање. На пример, свако возило може да се покрене, креће, заустави, па ће се те функционалности издвојити у надкласу, док ће се изведене класе разликовати по приказу: ауто, аутобус и камион не изгледају исто.

Креираћемо две апликације. У првој апликацији ћемо креирати надкласу *Vozilo* и изведене класе *Auto*, *Kamion* и *Autobus*. Корисник кликом на тастер миша креира једно возило које наставља да се креће надесно. Када стигне до краја форме, корисник може да креира друго возило на исти начин. У другој апликацији ћемо имати колону возила која се креће.

ПРВА АПЛИКАЦИЈА: Покренимо нови пројекат са називом *VoziloNasledjivanje01* у окружењу "Visual C# 2010 Express". Након притиска на десни тастер миша над називом пројекта у прозору *Solution Explorer* изабрати *Add.../Class...*



Унети назив класе *Vozilo.cs* уместо *Class1.cs*. Појављује се прозор у којем можемо да пишемо дефиницију класе. Вратимо се на дизајн апликације. Додајмо часовник *Timer* из прозора *Toolbox*. У прозору *Properties* на списку догађаја (испод муњице) за часовник *timer1* наћи догађај *Tick*, а на списку догађаја за форму *Form1* наћи догађај *MouseDown*, и два пута кликнути на бела поља поред назива догађаја. У сваком тренутку можемо да прелазимо са прозора са дефиницијом класе (**Vozilo.cs**), на прозор са наредбама апликације (**Form1.cs**) или на дизајн (**Form1.cs[Design]**).

У фајлу *Vozilo.cs* дописати *using System.Drawing;* на врх документа (одмах у реду испод *using System.Text;*) због класа које су нам потребне за цртање, као што су *Graphics*, *Pen* и *SolidBrush*. Написати дефиницију класе *Vozilo*, а затим у истом фајлу и изведене класе *Auto*, *Kamion* и *Autobus*.

Класе – фајл **Vozilo.cs**:

```
abstract public class Vozilo
{
    protected int x, y; //polozaj
    protected int a, b; //dimenzije
    protected int dx; //brzina
    protected Color boja;

    public int X
    {
        get { return x; } set { x = value; }
    }
    public Vozilo()
    {
        this.x = 0; this.y = 0; this.a = 0;
        this.b = 0; this.boja = Color.White;
    }
    public Vozilo(int x, int y, int a, int b, Color boja)
    {
        this.x = x; this.y = y; this.a = a; this.b = b; this.boja = boja;
    }
    public void Pokreni(int dx) { this.dx = dx; }
    public void Zaustavi() { this.dx = 0; }
    public void Pomeri() { x += dx; }
    abstract public void Crtaj(Graphics g);
}

public class Auto:Vozilo
{
    public Auto(int x, int y, int a, int b, Color boja):base(x, y, a, b, boja) {}

    public override void Crtaj(Graphics g)
    {
        SolidBrush cetka = new SolidBrush(boja);
        Pen olovka = new Pen(Color.Black);
        g.FillRectangle(cetka, x - a / 4, y - b / 2, a / 2, b / 2);
        g.FillRectangle(cetka, x - a / 2, y, a, b / 2);
        g.DrawLine(olovka, x - a / 4, y - b / 2, x + a / 4, y - b / 2);
        g.DrawLine(olovka, x - a / 2, y + b / 2, x + a / 2, y + b / 2);
        g.DrawLine(olovka, x - a / 2, y, x - a / 4, y);
        g.DrawLine(olovka, x + a / 2, y, x + a / 4, y);
        g.DrawLine(olovka, x - a / 2, y, x - a / 2, y + b / 2);
        g.DrawLine(olovka, x + a / 2, y, x + a / 2, y + b / 2);
        g.DrawLine(olovka, x - a / 4, y, x - a / 4, y - b / 2);
        g.DrawLine(olovka, x + a / 4, y, x + a / 4, y - b / 2);
        cetka.Color = Color.Black;
        g.FillEllipse(cetka, x - a / 4 - a / 6, y + b / 2 - a / 6, a / 3, a / 3);
        g.FillEllipse(cetka, x + a / 4 - a / 6, y + b / 2 - a / 6, a / 3, a / 3);
    }
}
```

```

public class Kamion : Vozilo
{
    public Kamion(int x, int y, int a, int b, Color boja) : base(x, y, a, b, boja) {}

    public override void Crtaj(Graphics g)
    {
        SolidBrush cetka = new SolidBrush(boja);
        Pen olovka = new Pen(Color.Black);
        g.FillRectangle(cetka, x - a / 2, y - b / 2, a, b);
        g.DrawRectangle(olovka, x - a / 2, y - b / 2, a, b);
        g.DrawLine(olovka, x + a / 2, y - b / 4, x + 3 * a / 4, y - b / 4);
        g.DrawLine(olovka, x + 3 * a / 4, y - b / 4, x + a, y);
        g.DrawLine(olovka, x + a, y, x + a, y + b / 2);
        g.DrawLine(olovka, x + a, y + b / 2, x + a / 2, y + b / 2);
        cetka.Color = Color.Black;
        g.FillEllipse(cetka, x - a / 2 + a / 6, y + b / 2 - a / 6, a / 3, a / 3);
        g.FillEllipse(cetka, x + a / 4 + a / 6, y + b / 2 - a / 6, a / 3, a / 3);
    }
}

public class Autobus : Vozilo
{
    public Autobus(int x, int y, int a, int b, Color boja) : base(x, y, a, b, boja) {}

    public override void Crtaj(Graphics g)
    {
        SolidBrush cetka = new SolidBrush(boja);
        Pen olovka = new Pen(Color.Black);
        g.FillRectangle(cetka, x - 2 * a / 3, y - b / 2, 4 * a / 3, b);
        g.DrawRectangle(olovka, x - 2 * a / 3, y - b / 2, 4 * a / 3, b);
        cetka.Color = Color.LightBlue;
        g.FillRectangle(cetka, x - 2 * a / 3 + a / 10, y - b / 2 + b / 10, 4 * a / 3 - a / 5, b / 2);
        g.DrawRectangle(olovka, x - 2 * a / 3 + a / 10, y - b / 2 + b / 10, 4 * a / 3 - a / 5, b / 2);
        cetka.Color = Color.Black;
        g.FillEllipse(cetka, x - a / 4 - a / 6, y + b / 2 - a / 6, a / 3, a / 3);
        g.FillEllipse(cetka, x + a / 4 - a / 6, y + b / 2 - a / 6, a / 3, a / 3);
    }
}

```

Апликација – фајл **Form1.cs**:

```

bool imaVozilo = false;
Vozilo vozilo;
Random r = new Random();

private void Form1_MouseDown(object sender, MouseEventArgs e)
{
    if (!imaVozilo)
    {
        Color boja = Color.FromArgb(r.Next(256), r.Next(256), r.Next(256));
        int tip = r.Next(3);
        if (tip == 0) vozilo = new Auto(e.X, e.Y, 70, 50, boja);
        else if (tip == 1) vozilo = new Kamion(e.X, e.Y, 70, 50, boja);
        else vozilo = new Autobus(e.X, e.Y, 70, 50, boja);
        vozilo.Pokreni(r.Next(3, 10)); timer1.Start(); imaVozilo = true;
    }
}

private void timer1_Tick(object sender, EventArgs e)
{
    vozilo.Pomeri(); Graphics g = CreateGraphics(); g.Clear(BackColor); vozilo.Crtaj(g);
    if (vozilo.X > ClientRectangle.Width + 70) imaVozilo = false;
}

```

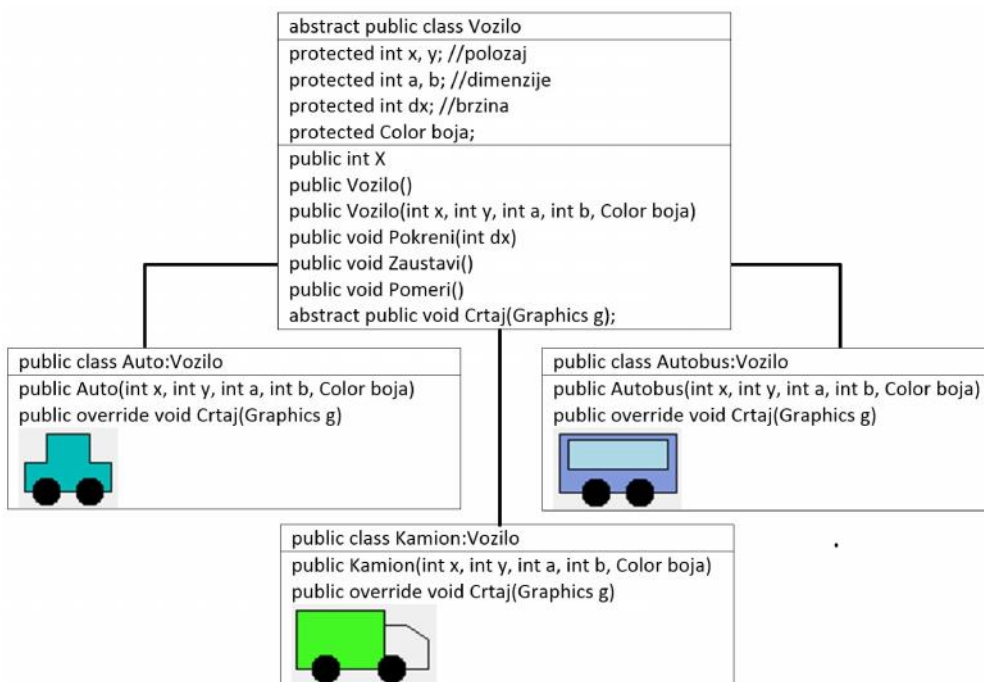
ДРУГА АПЛИКАЦИЈА: Покренимо нови пројекат са називом *VoziloNasledjivanje02*. Након притиска на десни тастер миша над називом пројекта у прозору *Solution Explorer* изабрати *Add.../Existing Item...* У дијалогу који се појави, пронаћи и изабрати фајл са класом *Vozilo.cs* у фолдеру са предходно креираним пројектом. Потребно је затим да се дода *using VoziloNasledjivanje01*; на врх фајла *Form1.cs* новог пројекта и објекти класа *Auto*, *Autobus* и *Kamion* могу да се користе и у овој апликацији без њиховог поновног писања.

```
Random r = new Random(); Vozilo[] vozilo = new Vozilo[50]; int n;

private void Form1_Load(object sender, EventArgs e)
{
    int a = 60; int b = 40;
    for (int x = -5 * ClientRectangle.Width; x < ClientRectangle.Width; x += 2 * a)
    {
        Color boja = Color.FromArgb(r.Next(256), r.Next(256), r.Next(256));
        int tip = r.Next(3);
        if (tip == 0) vozilo[n] = new Auto(x, ClientRectangle.Height / 2, a, b, boja);
        else if (tip == 1) vozilo[n] = new Kamion(x, ClientRectangle.Height / 2, a, b, boja);
        else vozilo[n] = new Autobus(x, ClientRectangle.Height / 2, a, b, boja);
        n++;
    }
    for (int i = 0; i < n; i++) vozilo[i].Pokreni(5);
    timer1.Start();
}

private void Form1_Paint(object sender, PaintEventArgs e)
{ for (int i = 0; i < n; i++) vozilo[i].Crtaj(e.Graphics); }

private void timer1_Tick(object sender, EventArgs e)
{ for (int i = 0; i < n; i++) vozilo[i].Pomeri(); Refresh(); }
```



За програмирање у програмском језику C# може да се користи бесплатно развојно окружење **Visual Studio Express 2013 for Windows Desktop**. Може да се користи и ранија верзија 2010, која може да се прузме са истог сајта.

Процес инсталације смо забележили као видео лекцију: <http://youtu.be/YYkorpJA-0>

Развојно окружење може да се преузме бесплатно са адресе:

<http://www.visualstudio.com/en-us/downloads/download-visual-studio-vs>