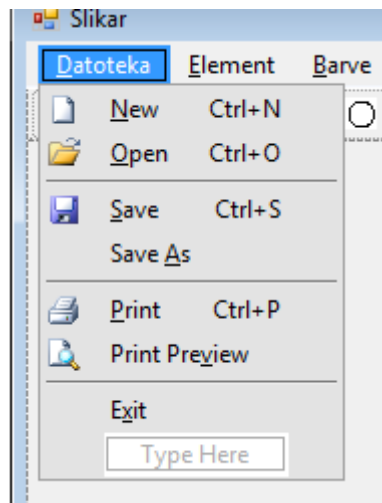


DELAVNICA ZA PROGRAMIRANJE VIZUALNIH APLIKACIJ V C++ 2

Projekt Slikar -1. del

V drugem tednu delavnice za programiranje vizualnih aplikacij v C++ smo začeli s projektom slikarja.

1. Ustvarimo projekt slikar (podobno kot smo ustvarili prvo aplikacijo)
2. V form1 dodamo menu strip, ter dodamo vse standardne gumba, nato pa odstranimo, vse razen gumba File.



3. V menu strip dodamo še gumba Element in Barve (razvidno zgoraj).
4. V začetku form1.h datoteke, ustvarimo dve spremenljivki, v kateri bomo hranili trenutno stanje izbire elementa in barve. Za barve uporabimo kar tip Color, ki je v System::Drawing, za elemente pa ustvarimo enum class.

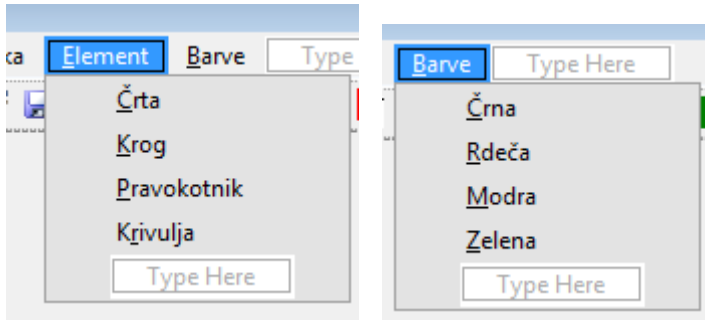
```
enum class TipElementa {CRTA,KROG,KRIVULJA,PRAVOKOTNIK};  
...  
public ref class Form1 : public System::Windows::Forms::Form  
{  
public:  
    Form1(void){  
        InitializeComponent();  
        trenutniTip=TipElementa::CRTA;  
        trenutnaBarva=Color::Black;  
        posodobiMenu();  
        risanje=false;  
        zacetnaTocka=Point(0);  
    }  
protected:  
    TipElementa trenutniTip;  
    Color trenutnaBarva;  
    ...  
};
```

5. V istem konstruktorju, določimo začetni element in barvo (razvidno zgoraj).

6. Določimo kaj se zgodi ob kliku gumba exit v Meni->Datoteka.

```
private: System::Void exitToolStripMenuItem_Click(System::Object^ sender, System::EventArgs^ e){
    Close();
}
```

7. Naredimo gumbe pod elementi za črto, krog, pravokotnik in krivuljo, ter barve črna, rdeča, modra in zelena. Tem gumbom tudi določimo dejanja, ki se izvedejo, ko kliknemo na kateri gumb. Za dejanja še ustvarimo dodatno funkcijo posodobiMenu, ki bo spreminjala trenutno stanje elementov in barv v meniju, ter orodni vrstici, ki jo bomo kasneje dodali.



```
private: System::Void črtaToolStripMenuItem_Click(System::Object^ sender, System::EventArgs^ e) {
    trenutniTip=TipElementa::CRTA;
    posodobiMenu();
}
private: System::Void krogToolStripMenuItem_Click(System::Object^ sender, System::EventArgs^ e) {
    trenutniTip=TipElementa::KROG;
    posodobiMenu();
}
private: System::Void pravokotnikToolStripMenuItem_Click(System::Object^ sender, System::EventArgs^ e) {
    trenutniTip=TipElementa::PRAVOKOTNIK;
    posodobiMenu();
}
private: System::Void krivuljaToolStripMenuItem_Click(System::Object^ sender, System::EventArgs^ e) {
    trenutniTip=TipElementa::KRIVULJA;
    posodobiMenu();
}
private: System::Void črnaToolStripMenuItem1_Click(System::Object^ sender, System::EventArgs^ e) {
    trenutnaBarva=Color::Black;
    posodobiMenu();
}
private: System::Void rdečaToolStripMenuItem_Click(System::Object^ sender, System::EventArgs^ e) {
    trenutnaBarva=Color::Red;
    posodobiMenu();
}
private: System::Void modraToolStripMenuItem_Click(System::Object^ sender, System::EventArgs^ e){
    trenutnaBarva=Color::Blue;
    posodobiMenu();
}
private: System::Void zelenaToolStripMenuItem_Click(System::Object^ sender, System::EventArgs^ e) {
    trenutnaBarva=Color::Green;
    posodobiMenu();
}
void posodobiMenu(){
    črtaToolStripMenuItem->Checked=(trenutniTip==TipElementa::CRTA);
    krogToolStripMenuItem->Checked=(trenutniTip==TipElementa::KROG);
    pravokotnikToolStripMenuItem->Checked=(trenutniTip==TipElementa::PRAVOKOTNIK);
}
```

```

krivuljaToolStripMenuItem->Checked=(trenutniTip==TipElementa::KRIVULJA);
črnaToolStripMenuItem->Checked=(trenutnaBarva==Color::Black);
rdečaToolStripMenuItem->Checked=(trenutnaBarva==Color::Red);
modraToolStripMenuItem->Checked=(trenutnaBarva==Color::Blue);
zelenaToolStripMenuItem->Checked=(trenutnaBarva==Color::Green);
gCrta->Checked=(trenutniTip==TipElementa::CRTA);
gKrog->Checked=(trenutniTip==TipElementa::KROG);
gPravokotnik->Checked=(trenutniTip==TipElementa::PRAVOKOTNIK);
gKrivulja->Checked=(trenutniTip==TipElementa::KRIVULJA);
gCrna->Checked=(trenutnaBarva==Color::Black);
gRdeca->Checked=(trenutnaBarva==Color::Red);
gModra->Checked=(trenutnaBarva==Color::Blue);
gZelena->Checked=(trenutnaBarva==Color::Green);
}

```

8. Formi dodamo orodno vrstico, kamor dodamo nekatere standardne gumbe, ter dodamo še 8 gumbov, za izbiro elementa in barve. Tem 8 gumbom izdelamo tudi pripadajoče sličice. Podobno kot smo naredili za gumbe v meniju, kjer izbiramo element in barvo, tudi tukaj določimo dejanja, ki se zgodijo, kadar kliknemo na gumb, vendar teh dejanj ne rabimo ponovno napisati, saj samo uporabimo dejanja, ki smo jih določili, za gumbe v meniju. Moramo biti samo pozorni, da določenemu gumbu določimo ustrezno dejanje.



9. Ustvarimo še dve spremenljivki, ki ju bomo potrebovali kasneje v projektu.

```

bool risanje;
Point zacetnaTocka;

```

10. Ustvarimo funkcijo mouse down, ki nam bo služila, za določitev stanja risanja, ter zajem začetne točke.

```

private: System::Void Form1_MouseDown(System::Object^ sender,
System::Windows::Forms::MouseEventArgs^ e) {
    if(e->Button==System::Windows::Forms::MouseButtons::Left){
        risanje=true;
        zacetnaTocka=e->Location;
    }
}

```

11. Ustvarimo dodatno Elementi.h datoteko, kjer bomo ustvarili razrede, za objekte črta, pravokotnik, krog, krivulja. (V prvem delu projekta slikar smo naredili samo bazni razred in podrazred črta).

```
#pragma once
```

```
using namespace System;  
using namespace System::Drawing;
```

```
namespace Slikar{  
    public ref class Element abstract {  
    protected:  
        Point polozaj;  
        Color barva;  
        Pen^ pen;  
        Rectangle mejniPravokotnik;  
    public:  
        virtual void Draw(Graphics^ g)abstract;  
    };  
    public ref class Crta:Element{  
    protected:  
        Point koncnaTocka;  
    public:  
        Crta(Color barva,Point zac,Point kon){  
            this->barva=barva;  
            polozaj=zac;  
            koncnaTocka=kon;  
            pen=gcnw Pen(barva);  
            mejniPravokotnik=Rectangle  
(Math::Min(polozaj.X,koncnaTocka.X),Math::Min(polozaj.Y,koncnaTocka.Y),  
Math::Abs(polozaj.X-koncnaTocka.X),Math::Abs(polozaj.Y-koncnaTocka.Y));  
            if(mejniPravokotnik.Width<2)  
                mejniPravokotnik.Width=2;  
            if(mejniPravokotnik.Height<2)  
                mejniPravokotnik.Height=2;  
        }  
        virtual void Draw(Graphics^ g)override  
        {  
            g->DrawLine(pen,polozaj,koncnaTocka);  
        }  
    };  
}
```

Opomba: Za določitev podčrtaja pri imenih gumbov (določimo katero tipko moramo pritisniti na tipkovnici, da izberemo želeno) uporabljamo & pred želeno črko.

Izgled forme po zaključku 1 dela projekta slikar:

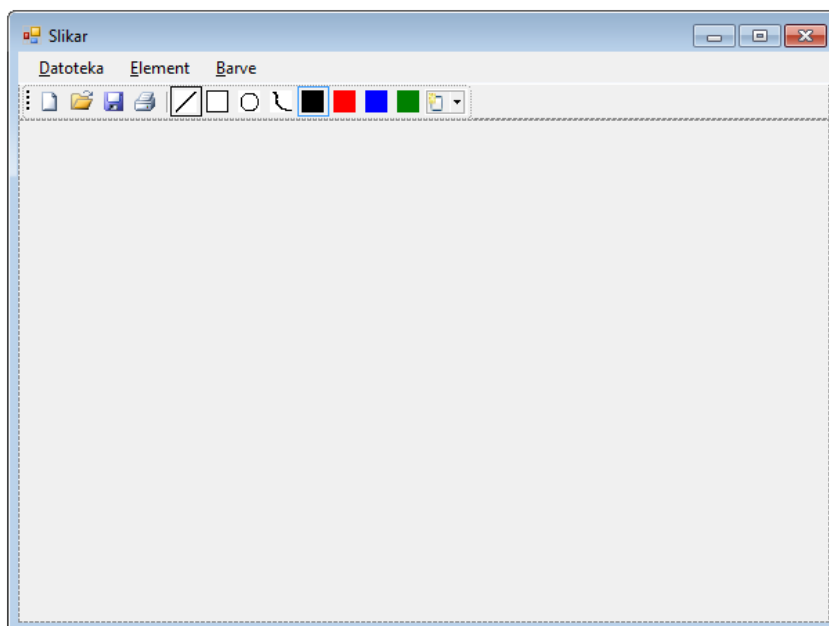


Tabela imen gumbov:

	Menu strip->Elementi	Menu strip->Barve	Tool strip->Elementi	Tool strip->Barve
Črta	črtaToolStripMenuItem		gCrta	
Pravokotnik	pravokotnikToolStripMenuItem		gPravokotnik	
Krog	krogToolStripMenuItem		gKrog	
Krivulja	krivuljaToolStripMenuItem		gKrivulja	
Črna		črnaToolStripMenuItem1		gCrna
Rdeča		rdečaToolStripMenuItem		gRdeca
Modra		modraToolStripMenuItem		gModra
Zelena		zelenaToolStripMenuItem		gZelena