


C++/CLI LAB8 – Alakzatok

Új projekt:  WindowsForm
Windows Form MOGI GP tárgyhoz -> LAB8

Felület:

3 db Label

3 db NumericUpDown

1 db CheckBox

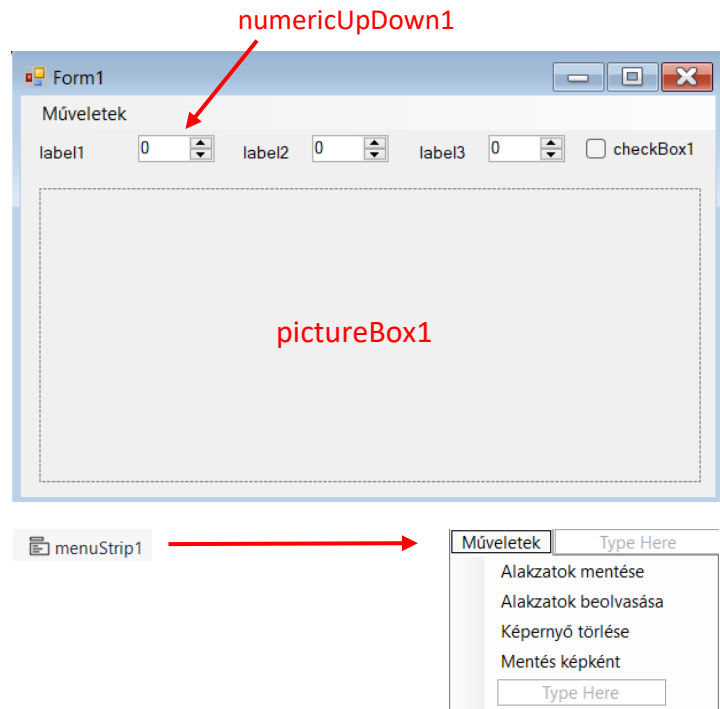
1 db PictureBox (Anchor: Top, Bottom, Left, Right)

1 db Timer

1 db MenuStrip

Ismeretek:

- Síkidom
- Egér klikkelés helye
- Véletlen szín előállítás



 Form1.h

```
using namespace System::Collections::Generic;           // List
using namespace System::IO;                             // StreamReader/StreamWriter
```

```
ref class Alakzat {
private:
    int x, y, oldal, meret, szog;
    Color szin;
public:
    Alakzat(int x, int y, int oldal, int meret, int szog, Color szin)
        : x(x), y(y), oldal(oldal), meret(meret), szin(szin) {};
    // Alakzat rajzolása átadott Graphics felületre
    void DrawAlakzat(Graphics^ gr) {
        array<PointF>^ sikidom = gcnew array<PointF>(oldal);
        for (int j = 0; j < oldal; j++) {
            sikidom[j].X = meret * Math::Sin(j * 2 * Math::PI / oldal);
            sikidom[j].Y = meret * Math::Cos(j * 2 * Math::PI / oldal);
        }
        gr->ResetTransform();
        gr->TranslateTransform(x, y);
        gr->RotateTransform(szog);
        gr->FillPolygon(gcnew SolidBrush(szin), sikidom);
    }
}
```

```

    // az origó helye
    property Point Origo {
        Point get() { return Point(x, y); }
    }
    // fájlba mentéshez
    String^ ToString() override {
        return String::Format("{1}{0}{2}{0}{3}{0}{4}{0}{5}{0}{6}",
            "\t", x, y, oldal, meret, szog, szin.ToArgb());
    }
};

// Globális változók
Graphics^ gr;
Bitmap^ bm;
Random^ randi = gcnew Random();

// ebben a listában kerülnek tárolásra az Alakzat objektumok
List<Alakzat^> alakzatok = gcnew List<Alakzat^>();

// ablak indul
private: System::Void Form1_Load(System::Object^ sender, System::EventArgs^ e) {
    Text = "Alakzatok";
    label1->Text = "Oldalszám";
    label2->Text = "Méret";
    label3->Text = "Írány";
    checkBox1->Text = "Véletlen adatok";
    numericUpDown1->Minimum = 3;
    numericUpDown2->Minimum = 10;
    numericUpDown3->Maximum = 360;
    bm = gcnew Bitmap(pictureBox1->Width, pictureBox1->Height);
    gr = Graphics::FromImage(bm);
}

// egér klikkelés eseménye
// e->Button visszaadja, hogy melyik gommal történt a klikkelés
// e->X és e->Y megadja a klikkelés helyét
private: System::Void pictureBox1_MouseClick(System::Object^ sender,
System::Windows::Forms::MouseEventArgs^ e) {
    if (e->Button == System::Windows::Forms::MouseButtons::Left) //bal gomb
    {
        // alakzat méretei a vezérlőkről
        int n = (int)numericUpDown1->Value;
        int l = (int)numericUpDown2->Value;
        int f = (int)numericUpDown3->Value;
        if (checkBox1->Checked) // ha be van kapcsolva a Random
        { // véletlen méretek
            n = randi->Next(3, 10);
            l = randi->Next(10, 100);
            f = randi->Next(361);
        }
        // mindig változó színek
        Color szin = Color::FromArgb(randi->Next(256), randi->Next(256),
            randi->Next(256));
        // alakzat hozzáadása a listához
        alakzatok->Add(gcnew Alakzat(e->X, e->Y, n, l, f, szin));
    }
    else if (e->Button == System::Windows::Forms::MouseButtons::Right) //jobb gomb
    { // utolsó elem törlése
        if (alakzatok->Count > 0)
            alakzatok->RemoveAt(alakzatok->Count - 1);
    }
}

```

```

    // új lista kirajzoltatása
    Rajz();
}

// a listában található elemek kirajzolása
public: void Rajz() {
    // koordináta rendszer visszaállítása
    gr->ResetTransform();
    gr->Clear(Color::Transparent);
    int db = alakzatok->Count;
    // összekötő vonalak rajzolása
    if (db > 2) //vonalak kirajzolása
    { // a pontok számára létrehozott tömb
        array<Point>^ kezeppontok = gcnew array<Point>(db);
        for (int i = 0; i < db; i++)
            kezeppontok[i] = alakzatok[i]->Origo;
        // tömbben szereplő végpontok vonalakkal történő kirajzolása
        gr->DrawPolygon(Pens::Black, kezeppontok);
    }
    // alakzatok rajzolása
    for (int i = 0; i < db; i++)
    { // az osztályon belül létrehozott rajzoló függvény használatával
        alakzatok[i]->DrawAlakzat(gr);
    }
    // megjelenítés a pictureBox-on
    pictureBox1->Image = bm;
}

// ablak átméretezésekor újra számítás és megjelenítés
private: System::Void Form1_Resize(System::Object^ sender, System::EventArgs^ e) {
    bm = gcnew Bitmap(pictureBox1->Width, pictureBox1->Height);
    gr = Graphics::FromImage(bm);
    Rajz();
}

// képernyő törlése
private: System::Void képernyőTörléseToolStripMenuItem_Click(System::Object^ sender,
System::EventArgs^ e) {
    alakzatok->Clear();
    gr->Clear(Color::Transparent);
    pictureBox1->Image = bm;
}

// alakzatok adatainak elmentése
private: System::Void alakzatokMentéseToolStripMenuItem_Click(System::Object^ sender,
System::EventArgs^ e) {
    SaveFileDialog ^ sfd = gcnew SaveFileDialog(); // mentés dialógus ablak
    sfd->FileName = "";
    sfd->Filter = "Pontok|*.txt";
    if (sfd->ShowDialog() == System::Windows::Forms::DialogResult::OK) // ha megadtunk fájlt
    {
        StreamWriter^ sw = gcnew StreamWriter(sfd->FileName); // fájl megnyitása írásra
        for each (Alakzat ^ alakzat in alakzatok) // elemek lista bejárása
        { // a lista elemeinek konvertálása szöveggé
            // és soronkénti kiírása szövegfájlba
            sw->WriteLine(alakzat->ToString());
        }
        sw->Flush();
        sw->Close(); // bezárás
    }
}
}

```

```
// alakzat adatainak visszatöltése fájlból
private: System::Void alakzatokBeolvasásaToolStripMenuItem_Click(System::Object^ sender,
System::EventArgs^ e) {
    OpenFileDialog^ ofd = gcnew OpenFileDialog(); // megnyitás dialógus ablak
    ofd->FileName = "";
    ofd->Filter = "Pontok|*.txt";
    if (ofd->ShowDialog() == System::Windows::Forms::DialogResult::OK) // ha megadtunk fájlt
    {
        StreamReader^ sr = gcnew StreamReader(ofd->FileName); //fájl megnyitása olvasása
        alakzatok->Clear(); // lista ürítése
        while (!sr->EndOfStream)
        {
            String^ sor = sr->ReadLine(); // soronkénti olvasás
            if (sor->Contains("\t")) // van benne adat
            {
                // elemek kigyűjtése és konvertálása
                array<String^>^ resz = sor->Split('\t');
                int x = Convert::ToInt32(resz[0]);
                int y = Convert::ToInt32(resz[1]);
                int n = Convert::ToInt32(resz[2]);
                int l = Convert::ToInt32(resz[3]);
                int f = Convert::ToInt32(resz[4]);
                Color szin = Color::FromArgb(Convert::ToInt32(resz[5]));
                // alakzat hozzáadása a listához
                alakzatok->Add(gcnew Alakzat(x, y, n, l, f, szin));
            }
        }
        sr->Close(); //bezárás
        // új lista kirajzolása
        Rajz();
    }
}

// kép mentése átlátszó png-ként
private: System::Void mentésKépkéntToolStripMenuItem_Click(System::Object^ sender,
System::EventArgs^ e) {
    bm->Save("background.png", System::Drawing::Imaging::ImageFormat::Png);
}
}
```