


C++/CLI LAB3 – Fájelkezelés

Új projekt:  WindowsForm
Windows Form MOGI GP tárgyhoz -> LAB3

Felület:

2 db TextBox

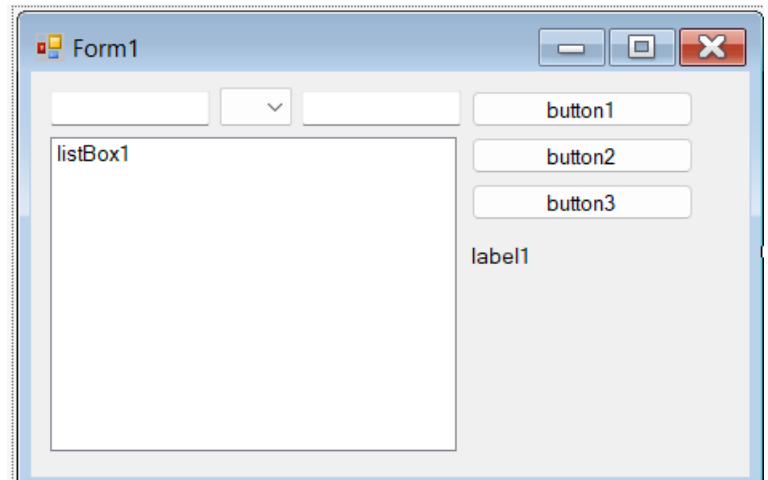
1 db ComboBox (DropDownStyle: **DropDownList**)

1 db ListBox

3 db Button

1 db Label


1 db OpenFileDialog



Ismeretek:

- fájl műveletek

- fájl olvasása, írása

 openFileDialog1

 Form1.h

```
using namespace System::IO; // fájlkezeléshez
using DialogResult = System::Windows::Forms::DialogResult; // álnév

// Ellenállás osztály
ref class Resistance {
private:
    double value;
public:
    Resistance():value(0) { }
    Resistance(double value):value(value) { }
    Resistance(String^ text) { value = ToResistance(text); }
    property double Value {
        double get() { return value; }
        void set(double value) { this->value = value; }
    }
    Resistance^ operator +(Resistance^ R) {
        return gcnew Resistance(Value + R->Value);
    }
    Resistance^ operator *(Resistance^ R) {
        return gcnew Resistance((Value * R->Value) / (Value + R->Value));
    }
}
```

```

// statikus adattagok, metódusok
static property String^ DecimalSeparator; // tizedesjegy triviális property
static double ToResistance(String^ text) // statikus konvertáló fgv
{
    double pref = 1;
    if (text->Contains("R")) {
        text = text->Replace("R", DecimalSeparator);
    }
    if (text->Contains("k")) {
        pref = 1E3;
        text = text->Replace("k", DecimalSeparator);
    }
    if (text->Contains("M")) {
        pref = 1E6;
        text = text->Replace("M", DecimalSeparator);
    }
    return Convert::ToDouble(text) * pref;
}
};

// ablak betöltése
private: System::Void Form1_Load(System::Object^ sender, System::EventArgs^ e) {
    // vizuális elemek beállítása
    Text = "Szöveg és fájl-kezelés";
    button1->Text = "=";
    button2->Text = "Betölt";
    button3->Text = "Lista ürítése";
    comboBox1->Items->Add("+");
    comboBox1->Items->Add("x");
    comboBox1->SelectedIndex = 0;

    // területi beállítások lekérdezése
    System::Globalization::CultureInfo^ culture =
        System::Threading::Thread::CurrentThread->CurrentCulture;
    // kultúra neve
    String^ cultureName = culture->TextInfo->CultureName;
    // tizedesjegy: tizedes vessző , (hu) vagy pont . (en)
    String^ decimalSeparator = culture->NumberFormat->NumberDecimalSeparator;
    // lista szeparáló elem: ; (hu) vagy , (en)
    String^ listSeparator = culture->TextInfo->ListSeparator;

    // beállítások kiírása
    label1->Text = String::Format("nyelv:{0}\ndec:{1}\nsep:{2}"
        , cultureName, decimalSeparator, listSeparator);

    // Resistance osztály is tudja, hogy mi a tizedesjegy karakter
    Resistance::DecimalSeparator = decimalSeparator;
}

// Számolás
private: System::Void button1_Click(System::Object^ sender, System::EventArgs^ e) {
    Resistance^ R1, ^ R2, ^ Re;
    int ok = 0; // hiba helyének detektálása
    try {
        // hiba keletkezhet a konvertáláskor
        R1 = gcnew Resistance(textBox1->Text);
        ok++; // ok = 1
        R2 = gcnew Resistance(textBox2->Text);
        ok++; // ok = 2
    }
    catch (Exception^ ex) {

```

```

        if (ok == 0) MessageBox::Show(L"Az első érték nem megfelelő!");
        else MessageBox::Show(L"A második érték nem megfelelő!");
    }

    if (ok == 2) {
        // számítás elvégzése a művelet alapján
        if (comboBox1->Text == "+") Re = R1 + R2;
        else Re = R1 * R2;
        //számítási eredmény hozzáadása a lista elejéhez
        listBox1->Items->Insert(0, String::Format("{0} {1} {2} = {3:F2}",
            textBox1->Text, comboBox1->Text, textBox2->Text, Re->Value));
    }
}

// billentyű leütések figyelése, mindkét textBox-hoz hozzá kell rendelni
private: System::Void textBox1_KeyPress(System::Object^ sender,
    System::Windows::Forms::KeyPressEventArgs^ e) {
    // karakterek kicserélése
    if (e->KeyChar == 'r') e->KeyChar = 'R';
    if (e->KeyChar == 'm') e->KeyChar = 'M';
    if (e->KeyChar == 'k') e->KeyChar = 'k';
    // csak a számokat, R, k, M karaktereket, vagy Backspace-t engedjük használni
    if (e->KeyChar >= '0' && e->KeyChar <= '9' || e->KeyChar == 'R'
        || e->KeyChar == 'k' || e->KeyChar == 'M' || e->KeyChar == 8)
        e->Handled = false;
    // a többi billentyű nem kerül kiküldésre
    else e->Handled = true;
}

// lista ürítése
private: System::Void button3_Click(System::Object^ sender, System::EventArgs^ e) {
    listBox1->Items->Clear();
}

// adatok fájlból történő betöltése
// Excelből -> csv fájl készítése
// oszlopok: | R1 | művelet(+ vagy x) | R2 | =
// az elválasztó karakter területi beállítás függő, magyarnál (;) angolnál (,)
private: System::Void button2_Click(System::Object^ sender, System::EventArgs^ e) {
    // fájl kijelölése openFileDialog ablak segítségével
    // FileName tartalmazza majd a kijelölt fájl teljes elérési útját
    openFileDialog1->FileName = ""; // ne jelenjen meg név induláskor
    // Filter határozza meg, hogy milyen fájl típusok közül választhatunk
    openFileDialog1->Filter = "Ellenállások (*.csv)|*.csv";
    // az ablak a ShowDialog metódussal hívható meg, Megnyitás és Mégsem gombokkal
    DialogResult answer = openFileDialog1->ShowDialog();
    if (answer == DialogResult::OK) { // ha a megnyitás gombra klikkeltem
        String^ row = "";
        Char listSeparator = Convert::ToChar(System::Threading::Thread::CurrentThread
            ->CurrentCulture->TextInfo->ListSeparator);
        String^ fileName = openFileDialog1->FileName; // kijelölt fájl név
        StreamReader^ sr = gcnew StreamReader(fileName); // fájl megnyitásra olvasásra
        StreamWriter^ sw = gcnew StreamWriter(fileName + ".bak"); // átmeneti fájl írásra
        while (!sr->EndOfStream) { // olvasás a fájl végéig
            row = sr->ReadLine(); // egy sor beolvasása, sorvéggel
            if (row->Length > 0) { // megvizsgáljuk, hogy történt-e beolvasás
                // elemek szétszedése a szeparáló karakter alapján alapján Split metódussal
                // a szétszedett elemek egy array<String ^> ^tömbbe kerülnek
                array<String^>^ cols = row->Split(listSeparator);
                // számítás végrehajtása, hibás szöveg esetén a sor kimarad
                try {

```

```
        // szövegek konvertálása ellenállásra
        Resistance^ R1 = gcnw Resistance(cols[0]);
        Resistance^ R2 = gcnw Resistance(cols[2]);
        Resistance^ Re;
        // számítás a művelet alapján
        if (cols[1] == "+") Re = R1 + R2;
        else Re = R1 * R2;
        // eredmény kiírása a sor végére
        listBox1->Items->Add(String::Format("{0} {1} {2} = {3:F2}",
            cols[0], cols[1], cols[2], Re->Value));
        // az átmeneti fájlba történő kiírás
        // a formátum azonos a bemenettel, csak lesz benne eredmény is
        sw->WriteLine(String::Format("{0}{4}{1}{4}{2}{4}={4}{3:F2}",
            cols[0], cols[1], cols[2], Re->Value, listSeparator));
    }
    catch (Exception^ ex) {
        // nem kell csinálni semmit hiba esetén a sor kimarad
    }
}
} // while vége

sr->Close(); // olvasás bezárása
sw->Flush(); // írás kényszerítése
sw->Close(); // írás bezárása

File::Delete(fileName); // eredeti fájlt törlése, majd
// az átmeneti fájlt átnevezése az eredeti fájl nevére
File::Move(fileName + ".bak", fileName);
}
}
```