

```

//Névterek feloldása
using namespace Globalization; // területi beállítások
using namespace System::Threading; // aktuális szál
using namespace System::IO; // fájlkezelés
using valasz = System::Windows::Forms::DialogResult; //állnév

-----

//Globális változók
String ^dec, ^sep, ^filename, ^sor;
System::Globalization::CultureInfo ^kult; // kultúra információk
StreamReader ^sr; // fájl olvasó
StreamWriter ^sw; // fájl író

// 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;

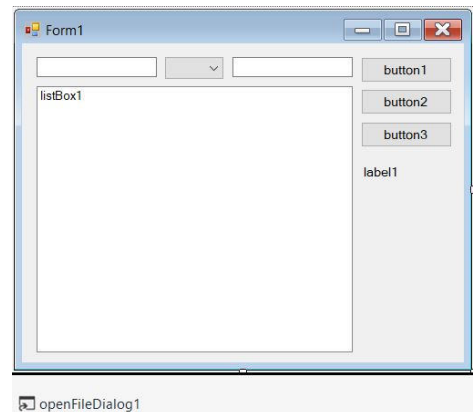
    // területi beállítások lekérdezése
    kult = Thread::CurrentThread->CurrentCulture;
    // tizedes vessző vagy pont
    dec = kult->NumberFormat->NumberDecimalSeparator;
    // lista szeparáló elem
    sep = kult->TextInfo->ListSeparator;

    //beállítások kiírása
    label1->Text = String::Format("nyelv:{0}\ndec:{1}\nsep:{2}"
        , kult->TextInfo->CultureName, dec, sep);
}

// Szöveg -> ellenállás érték konvertáló
double ConvertR(String ^s)
{
    double pref = 1;
    if (s->Contains("R"))
    {
        s = s->Replace("R", dec);
    }
    if (s->Contains("k"))
    {
        pref = 1E3;
        s = s->Replace("k", dec);
    }
    if (s->Contains("M"))
    {
        pref = 1E6;
        s = s->Replace("M", dec);
    }
    return Convert::ToDouble(s)*pref;
}

// Számolás
private: System::Void button1_Click(System::Object^ sender, System::EventArgs^ e)
{
    double a, b, c;
    int ok = 0; // hiba helyének detektálása
    try

```



```

{
    // hiba keletkezhet a konvertáláskor
    a = ConvertR(textBox1->Text);
    ok++; // ok = 1
    b = ConvertR(textBox2->Text);
    ok++; // ok = 2
}
catch (Exception ^e)
{
    if (ok == 0)
        MessageBox::Show("Az első érték nem megfelelő!");
    else
        MessageBox::Show("A második érték nem megfelelő!");
}

if (ok == 2)
{
    // számítás elvégzése a művelet alapján
    if (comboBox1->Text == "+") c = a + b;
    else c = replus(a, b);

    //számítási eredmény hozzáadása a lista elejéhez
    listBox1->Items->Insert(0, String::Format("{0} {1} {2} = {3:F2}",
        textBox1->Text, comboBox1->Text, textBox2->Text, c));
}
}

// ellenállások replussza
double replus(double a, double b)
{
    return (a*b) / (a + b);
}

// 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
    System::Windows::Forms::DialogResult dr = openFileDialog1->ShowDialog();
    if (dr == Valasz::OK) // ha a megnyitás gombra klikkeltem
    {
        filename = openFileDialog1->FileName; // kijelölt fájl név
        sr = gcnew StreamReader(filename); // fájl megnyitásra olvasásra
        sw = gcnew StreamWriter(filename+".bak"); // átmeneti fájl megnyitásra írásra

        while (!sr->EndOfStream) // olvasás a fájl végjelig
        {
            sor = sr->ReadLine(); // egy sor beolvasása, sorvégjelig

            if (sor->Length > 0) // megvizsgáljuk, hogy történt-e beolvasás
            {
                // elemek szétszedése a szeparáló karakter alapján Split metódussal
                // a szétszedett elemek egy array<String ^> ^tömbbe kerülnek
                array<String ^> ^ oszlopok = sor->Split(Convert::ToChar(sep));
            }
        }
    }
}

```

```

        //számítás végrehajtása, hibás szöveg esetén a sor kimarad
        try
        {
            double a = ConvertR(oszlopok[0]); // szöveg konvertálása ellenállás
            értékre
            double b = ConvertR(oszlopok[2]); // szöveg konvertálása ellenállás
            értékre
            double c;

            // számítás a művelet alapján
            if (oszlopok[1] == "+") c = a + b;
            else c = replus(a, b);

            // eredmény kiírása a sor végére
            listBox1->Items->Add(String::Format("{0} {1} {2} = {3:F2}",
                oszlopok[0], oszlopok[1], oszlopok[2], c));

            // 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}",
                oszlopok[0], oszlopok[1], oszlopok[2], c, sep));
        }
        catch (Exception ^e)
        {
        }
    }
}
// olvasás bezárása
sr->Close();
// írás kényszerítése
sw->Flush();
// írás bezárása
sw->Close();
}

File::Delete(filename); //eredeti fájlt törlése
File::Move(filename + ".bak", filename); //átmeneti fájlt átnevezése az eredetiére
}

// billentyű leütések figyelése
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;
}
}

```