

```
<connectionStrings>
    <add name="ConStr" connectionString="Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=|DataDirectory|\12011c2gyak.mdf;Integ
rated Security=True" />
</connectionStrings>
```

// AB kezelo osztalyokhoz

```
using System.Configuration;
using System.Data.SqlClient;
using System.Windows.Forms;
using System.Drawing;
```

```
//Ref->System.Configuration
```

```
// .mdf -> Copy= Cpy if newer
```

```
#region Adattagok
    static SqlConnection kapcsolat;
    static SqlCommand parancs;
#endregion
```

```
public static void Csatlakozas()
{
    try
    {
        kapcsolat = new SqlConnection();
        kapcsolat.ConnectionString =
ConfigurationManager.ConnectionStrings["ConStr"].ConnectionString;
        kapcsolat.Open();

        parancs = new SqlCommand();
        parancs.Connection = kapcsolat;
    }
    catch (Exception ex)
    {
        throw new ABKivetel("Sikertelen csatlakozas!", ex);
    }
}
```

```
public static void Lecsolakozas()
{
    try
    {
        kapcsolat.Close();
        parancs.Dispose();
    }
    catch (Exception ex)
    {
        throw new ABKivetel("Sikertelen lecsolakozas!", ex);
    }
}
```

// Kilistázás készítése

```
public static List<Televizio> TeleviziokListazasa()
{
    try
    {
        parancs.Parameters.Clear();
        parancs.CommandText = "SELECT * FROM [ElektronikaiCikk] INNER JOIN
[Televizio] ON [ElektronikaiCikk].[Id] = [Televizio].[Id]";
        List<Televizio> televiziok = new List<Televizio>();

        using (SqlDataReader olvaso = parancs.ExecuteReader())
        {
            while (olvaso.Read())
            {
                televiziok.Add(new
Televizio((KijelzoTechnologia)(int)olvaso["KijelzoTechnologia"],
(int)olvaso["Kepatlo"], (int)olvaso["Id"], olvaso["Gyarto"].ToString(),
olvaso["Tipus"].ToString(), (int)olvaso["Fogyasztas"], (int)olvaso["Ar"],
(int)olvaso["KiadasEve"]));
            }

            olvaso.Close();
        }

        return televiziok;
    }
    catch (Exception ex)
    {
        throw new ABKivetel("Sikertelen televizio kilistazas!", ex);
    }
}
```

```
public static List<Auto> AutokLekerdezese()
{
    try
    {
        parancs.Parameters.Clear();
        parancs.CommandText = "SELECT * FROM [Jarmu] INNER JOIN [Auto] ON
[Jarmu].[Id] = [Auto].[Id]";
        List<Auto> autok = new List<Auto>();

        using (SqlDataReader olvaso = parancs.ExecuteReader())
        {
            while (olvaso.Read())
            {
                autok.Add(new Auto((int)olvaso["Id"],
(Uzemanyag)(int)olvaso["Uzemanyag"], olvaso["Gyarto"].ToString()));
            }
            olvaso.Close();
        }
        return autok;
    }
    catch (Exception ex)
    {
        throw new ABKivetel("Sikertelen auto lekerdezese!", ex);
    }
}
```

// Felvitel

```
public static void TelevizioFelvitele(Televizio televizio)
{
    try
    {
        parancs.Parameters.Clear();
        parancs.CommandText = "INSERT INTO [ElektronikaiCikk]([Gyarto],
[Tipus], [Fogyasztas], [Ar], [KiadasEve]) OUTPUT INSERTED.Id VALUES (@gyarto,
@tipus, @fogyasztas, @ar, @kiadasEve)";
        parancs.Parameters.AddWithValue("@gyarto", televizio.Gyarto);
        parancs.Parameters.AddWithValue("@tipus", televizio.Tipus);
        parancs.Parameters.AddWithValue("@fogyasztas",
televizio.Fogyasztas);
        parancs.Parameters.AddWithValue("@ar", televizio.Ar);
        parancs.Parameters.AddWithValue("@kiadasEve", televizio.KiadasEve);

        televizio.Id = (int)parancs.ExecuteScalar();

        parancs.Parameters.Clear();
        parancs.CommandText = "INSERT INTO [Televizio] VALUES (@id,
@kijelzoTechnologia, @kepatlo)";
        parancs.Parameters.AddWithValue("@id", televizio.Id);
        parancs.Parameters.AddWithValue("@kijelzoTechnologia",
televizio.KijelzoTechnologia);
        parancs.Parameters.AddWithValue("@kepatlo", televizio.Kepatlo);

        parancs.ExecuteNonQuery();
    }
    catch (Exception ex)
    {
        throw new ABKivitel("Sikertelen televizio felvitel!", ex);
    }
}
```

//Felvitel 2

```
public static void AutoFelvitel(Auto auto)
{
    try
    {
        parancs.Parameters.Clear();
        parancs.CommandText = "INSERT INTO [Jarmu]([Gyarto], [Tipus],
[Szin], [Loketterfogat], [GyartasiEv]) OUTPUT INSERTED.Id VALUES (@gyarto, @tipus,
@szin, @loketterfogat, @gyartasiEv)";
        /*parancs.Parameters.AddWithValue("@gyarto", auto.Gyarto);*/
        /*parancs.Parameters.AddWithValue("@tipus", auto.Tipus);*/
        /*parancs.Parameters.AddWithValue("@szin", auto.Szin);*/
        /*parancs.Parameters.AddWithValue("@loketterfogat",
auto.Loketterfogat);*/
        /*parancs.Parameters.AddWithValue("@gyartasiEv", auto.GyartasiEv);*/

        /*auto.Id = (int)parancs.ExecuteScalar();*/

        parancs.Parameters.Clear();
        parancs.CommandText = "INSERT INTO [Auto] VALUES (@id, @uzemanyag,
@ferohelyekSzama)";
        /*parancs.Parameters.AddWithValue("@id", auto.Id);*/
        /*parancs.Parameters.AddWithValue("@uzemanyag",
(int)auto.Uzemanyag);*/
        /*parancs.Parameters.AddWithValue("@ferohelyekSzama",
auto.FerohelyekSzama);*/
        parancs.ExecuteNonQuery();
    }
    catch (Exception ex)
    {
        throw new ABKivetel("Sikertelen auto felvitel!", ex);
    }
}
```

// Módosítás

```
public static void TelevizioModositasa(Televizio televizio)
{
    try
    {
        parancs.Parameters.Clear();
        parancs.CommandText = "UPDATE [ElektronikaiCikk] SET [Ar] = @ar
WHERE [Id] = @id";
        parancs.Parameters.AddWithValue("@ar", televizio.Ar);
        parancs.Parameters.AddWithValue("@id", televizio.Id);
        parancs.ExecuteNonQuery();

        /*parancs.Parameters.Clear();
        parancs.CommandText = "UPDATE [Televizio] SET [Kepatlo] = @kepatlo
WHERE [Id] = @id";
        parancs.Parameters.AddWithValue("@ar", televizio.Kepatlo);
        parancs.Parameters.AddWithValue("@id", televizio.Id);
        parancs.ExecuteNonQuery();*/
    }
    catch (Exception ex)
    {
        throw new ABKivétel("Sikertelen televizio modositás!", ex);
    }
}
```

```
public static void AutoModositas(Auto auto)
{
    try
    {
        parancs.Parameters.Clear();
        parancs.CommandText = "UPDATE [Jarmu] SET [Szin] = @szin WHERE [Id]
= @id";
        /*parancs.Parameters.AddWithValue("@id", auto.Id);*/
        /*parancs.Parameters.AddWithValue("@szin", auto.Szin);*/
        parancs.ExecuteNonQuery();
    }
    catch (Exception ex)
    {
        throw new ABKivétel("Sikertelen auto modositás!", ex);
    }
}
```

// Törlés

```
public static void TelevizioTorlese(Televizio televizio)
{
    try
    {
        //Eloszor onnan torlok ahol az FK van
        parancs.Parameters.Clear();
        parancs.CommandText = "DELETE FROM [Televizio] WHERE [Id] = @id";
        parancs.Parameters.AddWithValue("@id", televizio.Id);
        parancs.ExecuteNonQuery();

        //Utana onnan torlok ahova az FK mutatott
        parancs.Parameters.Clear();
        parancs.CommandText = "DELETE FROM [ElektronikaiCikk] WHERE [Id] =
@id";

        parancs.Parameters.AddWithValue("@id", televizio.Id);
        parancs.ExecuteNonQuery();
    }
    catch (Exception ex)
    {
        throw new ABKivétel("Sikertelen televizio torles!", ex);
    }
}
```

```
public static void TelevizioTorlese(int id)
{
    try
    {
        //Eloszor onnan torlok ahol az FK van
        parancs.Parameters.Clear();
        parancs.CommandText = "DELETE FROM [Televizio] WHERE [Id] = @id";
        parancs.Parameters.AddWithValue("@id", id);
        parancs.ExecuteNonQuery();

        //Utana onnan torlok ahova az FK mutatott
        parancs.Parameters.Clear();
        parancs.CommandText = "DELETE FROM [ElektronikaiCikk] WHERE [Id] =
@id";

        parancs.Parameters.AddWithValue("@id", id);
        parancs.ExecuteNonQuery();
    }
    catch (Exception ex)
    {
        throw new ABKivétel("Sikertelen televizio torles!", ex);
    }
}
```

// Ősosztály

// : IComparable – kell , hogy összehasonlítható legyen

//Tablából EC kiolvasása

```
protected ElektronikaiCikk(int id, string gyarto, string tipus, int
fogyasztas, int ar, int kiadasEve)
```

```
{
    this.Id = id;
    this.Gyarto = gyarto;
    this.Tipus = tipus;
    this.Fogyasztas = fogyasztas;
    this.Ar = ar;
    this.KiadasEve = kiadasEve;
}
```

//Új EC létrehozása

```
protected ElektronikaiCikk(string gyarto, string tipus, int fogyasztas, int
ar, int kiadasEve)
```

```
{
    this.Gyarto = gyarto;
    this.Tipus = tipus;
    this.Fogyasztas = fogyasztas;
    this.Ar = ar;
    this.KiadasEve = kiadasEve;
}
```

#endregion

```
public override string ToString()
```

```
{
    return $"({id}) {Gyarto} {Tipus}: {Ar} Ft ({Fogyasztas}W, {KiadasEve}";
}
```

```
public virtual string CSV()
```

```
{
    return $" {Id};{Gyarto};{Tipus};{Fogyasztas};{Ar};{KiadasEve}";
}
```

//GET/SET beállítás

```
public int? Id
```

```
{
    get => id;
    set
    {
```

```
        if (id == null)
        {
            id = value;
        }
    }
```

```
    else
    {
```

```
        throw new InvalidOperationException("Az ID csak beallithato, de
nem modosithato!");
    }
```

```
    }
}
```

```
public string Tipus
{
    get => tipus;
    private set
    {
        if (!string.IsNullOrEmpty(value))
        {
            tipus = value;
        }
        else
        {
            throw new ArgumentException("A tipus megadása kötelező!");
        }
    }
}
```

```
public int Fogyasztas
{
    get => fogyasztas;
    private set
    {
        if (value >= 5 && value <= 1000)
        {
            fogyasztas = value;
        }
        else
        {
            throw new ArgumentException("A fogyasztas 5W és 1000W között..");
        }
    }
}
```

```
public int KiadasEve
{
    get => kiadasEve;
    private set
    {
        if (value <= DateTime.Now.Year)
        {
            kiadasEve = value;
        }
        else
        {
            throw new ArgumentException("A kiadas éve maximum a jelenlegi év lehet!");
        }
    }
}
```


// Alosztály

```
public enum KijelzoTechnologia
{
    TN, VA, IPS, PLS, egyeb
}
```

```
public Televizio(KijelzoTechnologia kijelzoTechnologia, int kepatlo, int id, string
gyarto, string tipus, int fogyasztas, int ar, int kiadasEve)
    : base(id, gyarto, tipus, fogyasztas, ar, kiadasEve)
{
    this.KijelzoTechnologia = kijelzoTechnologia;
    this.Kepatlo = kepatlo;
}

//Uj T létrehozasa
public Televizio(KijelzoTechnologia kijelzoTechnologia, int kepatlo, string
gyarto, string tipus, int fogyasztas, int ar, int kiadasEve)
    : base(gyarto, tipus, fogyasztas, ar, kiadasEve)
{
    this.KijelzoTechnologia = kijelzoTechnologia;
    this.Kepatlo = kepatlo;
}
```

```
public override string ToString()
{
    return base.ToString() + $", {KijelzoTechnologia}, {Kepatlo} inch) = br.
{BruttoTul} Ft";
}

public override string CSV()
{
    return base.CSV() + $";{(int)kijelzoTechnologia};{kepatlo}";
}
```

```
private Size felbontas;

public Size Felbontas
{
    get => felbontas;
    private set
    {
        if ((value.Width >= 640 && value.Height >= 480) && (value.Width <=
4096 && value.Height <= 2160))
        {
            felbontas = value;
        }
        else
        {
            throw new ArgumentException("A felbontasnak 640x480 es 4096x2160
koze kell esnie!");
        }
    }
}
```

// Fő Form

```
using System.Windows.Forms;
using System.IO;
```

```
//lista dekralálása
```

```
private List<Televizio> televiziok; // és ha kell a többit
```

```
public frm_ElektronikaiCikkek()
{
    InitializeComponent();
    ABKezelo.Csatlakozas();

    try
    {
        TeleviziokFrissitese();
        ProjektorokFrissitese();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "Hiba", MessageBoxButtons.OK,
        MessageBoxIcon.Error);
    }
}
```

//lb frissítése

```
public void TeleviziokFrissitese()
{
    televiziok = null;
    televiziok = ABKezelo.TeleviziokListazasa();

    lb_Televiziok.DataSource = null;
    lb_Televiziok.DataSource = televiziok;
}
```

```
private void frm_ElektronikaiCikkek_FormClosing(object sender, FormClosingEventArgs e)
{
    ABKezelo.Lecsatlakozas();
}
```

// új elem hozzá adása

```
private void bt_UjTv_Click(object sender, EventArgs e)
{
    frm_TelevizioUrlap frm = new frm_TelevizioUrlap();
    if (frm.ShowDialog() == DialogResult.OK)
    {
        try
        {
            ABKezelo.TelevizioFelvitele(frm.Televizio);
            TeleviziokFrissitese();
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message, "Hiba", MessageBoxButtons.OK,
            MessageBoxIcon.Error);
        }
    }
}
```

```

    }
//törlés gomb

private void bt_Torol_Click(object sender, EventArgs e)
{
    if (lb_Televizioik.SelectedIndex != -1)
    {
        if (MessageBox.Show("Biztos torli a televiziot?", "Torles",
        MessageBoxButtons.OKCancel, MessageBoxIcon.Question) == DialogResult.OK)
        {
            try
            {
                Televizio televizio = (Televizio)lb_Televizioik.SelectedItem;
                ABKezelo.TelevizioTorlese(televizio);
                TelevizioikFrissitese();
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message, "Hiba", MessageBoxButtons.OK,
        MessageBoxIcon.Error);
            }
        }
        else
        {
            MessageBox.Show("Jeloljon ki egy televiziot!", "Hiba",
        MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }
}

```

//módosítás gomb

```

private void bt_TvModosit_Click(object sender, EventArgs e)
{
    if (lb_Televizioik.SelectedIndex != -1)
    {
        Televizio televizio = (Televizio)lb_Televizioik.SelectedItem;
        frm_TelevizioUrlap frm = new frm_TelevizioUrlap(televizio);
        if (frm.ShowDialog() == DialogResult.OK)
        {
            try
            {
                ABKezelo.TelevizioModositasa(frm.Televizio);
                TelevizioikFrissitese();
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message, "Hiba", MessageBoxButtons.OK,
        MessageBoxIcon.Error);
            }
        }
        else
        {
            MessageBox.Show("Jeloljon ki egy televiziot!", "Hiba",
        MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }
}

```

//CSV mentés gomb

```
private void bt_CSV_Click(object sender, EventArgs e)
{
    try
    {
        string fnev = "elektronikaiCikkek.csv";
        using (StreamWriter file = new StreamWriter(fnev, false,
Encoding.UTF8))
        {
            file.WriteLine("Televizio");
            foreach (Televizio televizio in televizio)
            {
                file.WriteLine(televizio.CSV());
            }

            file.WriteLine();
            file.WriteLine("Projektor");
            foreach (Projektor projektor in projektor)
            {
                file.WriteLine(projektor.CSV());
            }

            file.Close();
        }

        MessageBox.Show("Sikeres exportálás!", "Exportálás",
MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
    catch (Exception ex)
    {
        MessageBox.Show("Sikertelen exportálás!", "Hiba",
MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}
```

// Űrlap Formok

```
#region Adattagok
    private Projektor projektor;
#endregion
```

```
public frm_ProjektorUrlap()
{
    InitializeComponent();
    nud_KiadasEve.Maximum = DateTime.Now.Year;
}

public frm_ProjektorUrlap(Projektor projektor) : this()
{
    this.projektor = projektor;

    tb_Gyarto.Text = projektor.Gyarto;
    tb_Tipus.Text = projektor.Tipus;
    nud_Fogyasztas.Value = projektor.Fogyasztas;
    nud_Ar.Value = projektor.Ar;
    nud_KiadasEve.Value = projektor.KiadasEve;
    nud_KontrasztArany.Value = projektor.KontrasztArany;
    nud_X.Value = projektor.Felbontas.Width;
    nud_Y.Value = projektor.Felbontas.Height;

    tb_Gyarto.Enabled = false;
    tb_Tipus.Enabled = false;
    nud_Fogyasztas.Enabled = false;
    nud_KiadasEve.Enabled = false;
    nud_KontrasztArany.Enabled = false;
    nud_X.Enabled = false;
    nud_Y.Enabled = false;
}
```

//Ok gomb

```
private void bt_Ok_Click(object sender, EventArgs e)
{
    try
    {
        if (projektor == null)
        {
            projektor = new Projektor((int)nud_KontrasztArany.Value, new
Size((int)nud_X.Value, (int)nud_Y.Value), tb_Gyarto.Text, tb_Tipus.Text,
(int)nud_Fogyasztas.Value, (int)nud_Ar.Value, (int)nud_KiadasEve.Value);
        }
        else
        {
            projektor.Ar = (int)nud_Ar.Value;
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "Hiba", MessageBoxButtons.OK,
MessageBoxIcon.Error);
        DialogResult = DialogResult.None;
    }
}

public Projektor Projektor { get => projektor; } // ez kell a végére
```

//Konstruktor, ha enum is van benne

```

public frm_TelevizioUrlap()
{
    InitializeComponent();
    cb_Kijelzo.DataSource = Enum.GetValues(typeof(KijelzoTechnologia));
    nud_KiadasEve.Maximum = DateTime.Now.Year;
}

public frm_TelevizioUrlap(Televizio televizio) : this()
{
    this.televizio = televizio;

    tb_Gyarto.Text = televizio.Gyarto;
    tb_Tipus.Text = televizio.Tipus;
    nud_Fogyasztas.Value = televizio.Fogyasztas;
    nud_Ar.Value = televizio.Ar;
    nud_KiadasEve.Value = televizio.KiadasEve;
    cb_Kijelzo.SelectedItem = televizio.KijelzoTechnologia;
    nud_Kepatlo.Value = televizio.Kepatlo;

    tb_Gyarto.Enabled = false;
    tb_Tipus.Enabled = false;
    nud_Fogyasztas.Enabled = false;
    nud_KiadasEve.Enabled = false;
    cb_Kijelzo.Enabled = false;
    nud_Kepatlo.Enabled = false;
}

```

//Ok gomb, ha enum is van benne

```

private void bt_Ok_Click(object sender, EventArgs e)
{
    try
    {
        if (televizio == null)
        {
            televizio = new
Televizio((KijelzoTechnologia)cb_Kijelzo.SelectedItem, (int)nud_Kepatlo.Value,
tb_Gyarto.Text, tb_Tipus.Text, (int)nud_Fogyasztas.Value, (int)nud_Ar.Value,
(int)nud_KiadasEve.Value);
        }
        else
        {
            televizio.Ar = (int)nud_Ar.Value;
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "Hiba", MessageBoxButtons.OK,
MessageBoxIcon.Error);
        DialogResult = DialogResult.None;
    }
}

```

// SQL

```
CREATE TABLE [ElektronikaiCikk] (
    [Id] INT NOT NULL PRIMARY KEY IDENTITY(1, 1),
    [Gyarto] VARCHAR(128) NOT NULL,
    [Tipus] VARCHAR(128) NOT NULL,
    [Fogyasztas] INT NOT NULL,
    [Ar] INT NOT NULL,
    [KiadasEve] INT NOT NULL
);

CREATE TABLE [Televizio] (
    [Id] INT NOT NULL PRIMARY KEY,
    [KijelzoTechnologia] INT NOT NULL,
    [Kepatlo] INT NOT NULL,

    CONSTRAINT [FK_TId_ECId] FOREIGN KEY ([Id]) REFERENCES
[ElektronikaiCikk]([Id])
);

CREATE TABLE [Projektor] (
    [Id] INT NOT NULL PRIMARY KEY,
    [KontrasztArany] INT NOT NULL,
    [FelbontasSzelesseg] INT NOT NULL,
    [FelbontasMagassag] INT NOT NULL,

    CONSTRAINT [FK_PId_ECId] FOREIGN KEY ([Id]) REFERENCES
[ElektronikaiCikk]([Id])
);



---



CREATE TABLE [Jarmu] (
    [Id] INT NOT NULL PRIMARY KEY IDENTITY(1, 1),
    [Gyarto] VARCHAR(128) NOT NULL,
    [Tipus] VARCHAR(128) NOT NULL,
    [Szin] VARCHAR(128) NOT NULL,
    [Loketterfogat] INT NOT NULL,
    [GyartasiEv] INT NOT NULL
);

CREATE TABLE [Auto] (
    [Id] INT NOT NULL PRIMARY KEY,
    [Uzemanyag] INT NOT NULL,
    [FerohelyekSzama] INT NOT NULL,

    CONSTRAINT [FK_AutoID_JarmuID] FOREIGN KEY ([Id]) REFERENCES [Jarmu]([Id])
);

CREATE TABLE [Motor] (
    [Id] INT NOT NULL PRIMARY KEY,
    [Kivitel] VARCHAR(128) NOT NULL,
    [JogositvanyKategoria] INT NOT NULL,

    CONSTRAINT [FK_MotorID_JarmuID] FOREIGN KEY ([Id]) REFERENCES [Jarmu]([Id])
);
```

// Összehasonlítás

//ez az őosztályba kell

```
public int CompareTo(object obj)
{
    if (obj is ElektronikaiCikk ec)
    {
        if (fogyasztas < ec.Fogyasztas)
        {
            return -1;
        }
        else if (fogyasztas > ec.Fogyasztas)
        {
            return 1;
        }
        else
        {
            return 0;
        }
    }
    else
    {
        throw new InvalidOperationException("Elektronikai cikket csak
elektronikai cikkekkel lehet összehasonlítani!");
    }
}
```

// Összehasonlító form

```
private Dictionary<int, List<ElektronikaiCikk>> fogyasztasAlapuCsoporthozas;

public frm_AzonosFogyasztasuak()
{
    InitializeComponent();
    fogyasztasAlapuCsoporthozas = AzonosFogyasztasuakKilistazasa();
    Kiiras(0, 10000000);
}

private Dictionary<int, List<ElektronikaiCikk>> AzonosFogyasztasuakKilistazasa()
{
    Dictionary<int, List<ElektronikaiCikk>> fogyasztasAlapuCsoporthozas =
new Dictionary<int, List<ElektronikaiCikk>>();

    foreach (Televizio televizio in ABKezelo.TelevizioKilistazasa())
    {
        if (!fogyasztasAlapuCsoporthozas.ContainsKey(televizio.Fogyasztas))
        {
            fogyasztasAlapuCsoporthozas.Add(televizio.Fogyasztas, new
List<ElektronikaiCikk>());
        }

        fogyasztasAlapuCsoporthozas[televizio.Fogyasztas].Add(televizio);
    }
}
```



```

public void Kiiras(int min, int max)
{
    lb_AzonosFogyaszatasuak.Items.Clear();

    foreach (KeyValuePair<int, List<ElektronikaiCikk>> kvp in
fogyasztasAlapuCsoportositas)
    {
        if (kvp.Value.Count > 1)
        {
            foreach (ElektronikaiCikk ec in kvp.Value)
            {
                if (ec.BruttoTul >= min && ec.BruttoTul <= max)
                {
                    lb_AzonosFogyaszatasuak.Items.Add(ec);
                }
            }
        }
    }
}

```

```

private void bt_Szur_Click(object sender, EventArgs e)
{
    if (nud_MinBr.Value < nud_MaxBr.Value)
    {
        Kiiras((int)nud_MinBr.Value, (int)nud_MaxBr.Value);
        bt_Szur.Enabled = false;
        bt_SzuroTorlese.Enabled = true;
    }
    else
    {
        MessageBox.Show("Az also hatarnak kisebbnek kell lennie a felso
hatarnal!", "Hiba", MessageBoxButtons.OK, MessageBoxIcon.Error);
        DialogResult = DialogResult.None;
    }
}

```

```

private void bt_SzuroTorlese_Click(object sender, EventArgs e)
{
    bt_Szur.Enabled = true;
    bt_SzuroTorlese.Enabled = false;
    Kiiras(0, 10000000);
}

```