

1. Gyümölcsáruló automata

Készítsünk egy gyümölcsáruló automatát szimuláló alkalmazást. A program négy gyümölcsöt tud kezelni. A bal oldali oszlopban csak olvasható szövegmezőkben jelennek meg az egységárak. A jobb oldali nyomógombokon történő kattintás idézi elő az ár kiszámítását. A „Tömeg” felirat melletti szövegmezőbe kell beírja a felhasználó a mennyiséget. Az „Ár” felirat melletti szövegmezőben jelenik meg a fizetendő összeg.

A Beállít gombon történő kattintás hatására egy jelszóbekérő ablak jelenik meg. A „blabla” jelszó megadása után az egységárak szerkesztőmezői írhatóvá válnak. Az új értékek megadása után az Alkalmaz gombon történő kattintással rögzíthetjük az adatokat. Ennek hatására az egységár mezők újra csak olvashatóak lesznek.

Ár	Egység	Gyümölcs
150	Ft/kg	Álma
190	Ft/kg	Körte
256	Ft/kg	Banán
199	Ft/kg	Narancs

Tömeg kg

Ár Ft

Beállít Alkalmaz Kilép

Jelszó

OK

1.1. A feladat megoldása

A feladat megoldásának fontosabb lépései a következők.

- A főablak grafikus felületének létrehozása.
- Eseménykezelő a Kilép nyomógombhoz.
- Asszociatív tömb a szerkesztőmező nyomógomb párosokhoz.
- Közös eseménykezelő a négy gyümölcs nyomógombhoz.
- Egységárak megváltoztatási lehetőségének megvalósítása.

A teljes osztálydiagram a megoldás végén található.

1.1.1. A főablak grafikus felületének létrehozása

Hozunk létre egy új projektet Gyumolcsarazo néven. Az ablak objektumot nevezzük át **frmGyümölcsÁruló**-nak, és helyezzük el fejlécében a fenti képen látható szöveget. Ehhez tervezési nézetben (Design) válasszuk ki az ablakot (Form), majd nyissuk meg a Properties ablakot, ott válasszuk ki a (Name) tulajdonságot, és az eredeti Form1 értéket írjuk át frmGyümölcsÁruló-ra. A feliratot az előzőhöz hasonló módon a Text tulajdonság segítségével módosíthatjuk.

Properties

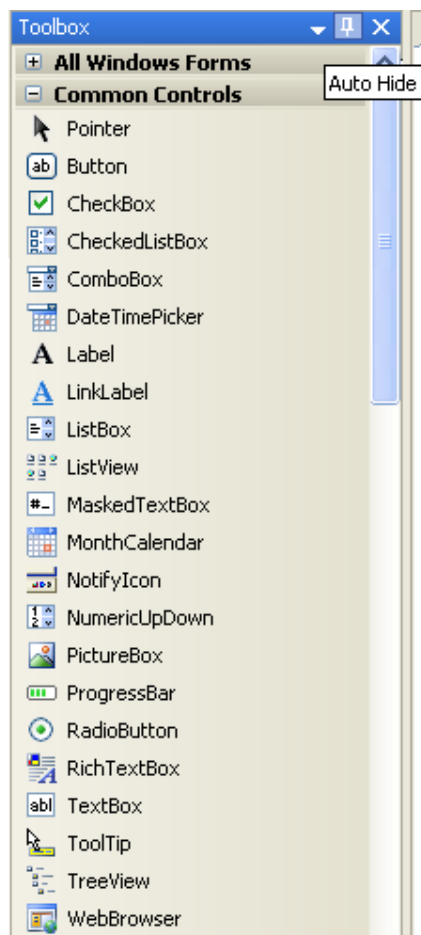
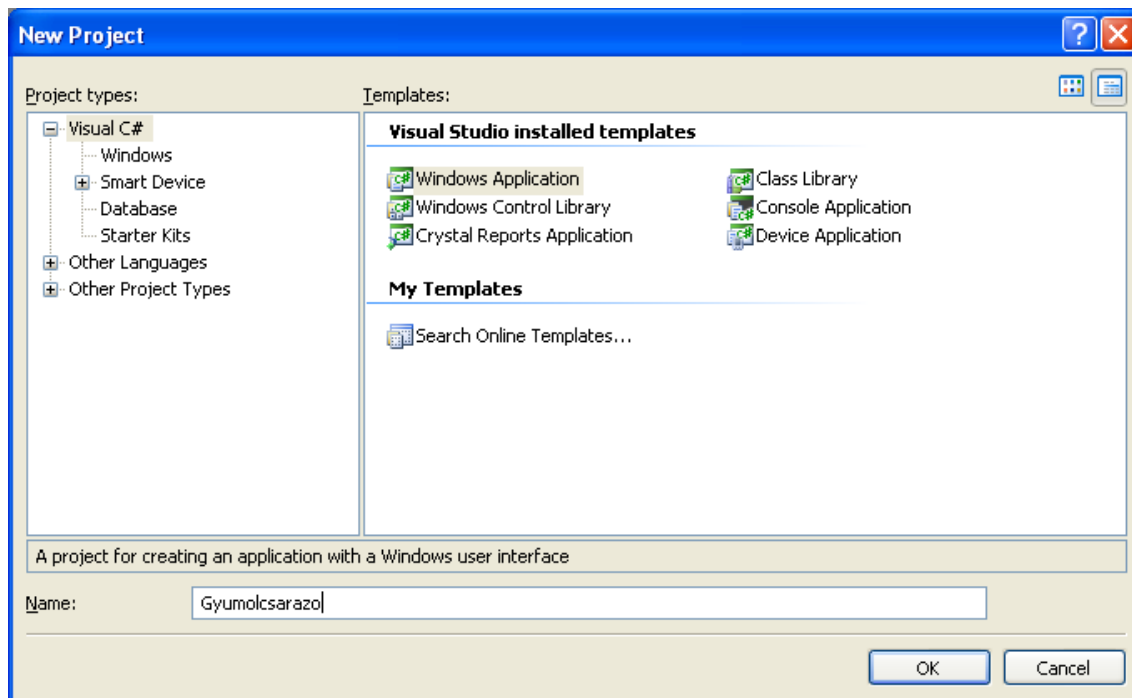
Form1 System.Windows.Forms.Form

(ApplicationSettings)

(DataBindings)

(Name) frmGyümölcsÁruló

A Solution Explorerben az ablak állományát Form1.cs-ről nevezzük át frmGyumolcsArazo.cs-rek (jobb egérgomb, Rename)



Következő lépésként helyezük el a komponenseket a formon (ablakon). Ehhez nyissuk meg a ToolBox palettát (a főablak keretében bal oldalon jelenik meg), majd kattintsunk a rajzszög gombon. Ebben a gyakorlatban olyan komponenseket használunk, amelyek a Common Controls csoportban vannak. Más esetekben egyszerűbb, ha az All Windows Forms csoportot nyitjuk meg, mert itt minden komponenst megtalálunk neveik szerint sorba rendezve. A három használt komponens a nyomógomb (Button), szövegmező (TextBox) és a címke (Label). Ezeket az egér segítségével foghatjuk meg és húzhatjuk az ablak területére. Általános elnevezési konvenció lesz a továbbiakban az, hogy egy ablakra helyezett komponens nevét két részből állítjuk össze: az első rész kisbetűs és a típust tükrözi, a második rész nagy betűvel kezdődik és a típuson belüli egyedi azonosításra szolgál. Tekintsük át az elnevezést típusonként.

A nyomógombok neveit úgy alakítsuk ki, hogy bt-vel kezdődjenek, majd a név további része legyen azonos a felirattal (pl. btAlma).

A szerkesztőmezők neve kezdődjön tb-vel, majd ezt követően legyen azonos a jobb oldalukon levő nyomógomb feliratával az első négy esetben (pl. tbAlma), illetve a bal oldalukon látható felirattal az utolsó két esetben (pl. tbTömeg).

Az ötödik kivétellel mindegyik szövegmező legyen

csak olvasható (ReadOnly=True). Az Alkalmaz nyomógomb legyen letiltva (Enabled=False).

1.1.2. Eseménykezelő a Kilép nyomógombhoz

Készítsünk egy eseménykezelőt a Kilép nyomógombhoz. Pl. tervezési nézetben duplán kattintunk a nyomógombon. Alternatív lehetőség az, hogy tervezési nézetben kijelöljük a nyomógombot, majd a Properties ablakban kiválasztjuk a villámra emlékeztető nyomógombot, kattintunk rajta, és a megjelenő eseménylistában kiválasztjuk a Click eseményt. Ezután duplán kattintunk a mellette levő üres mezőben.



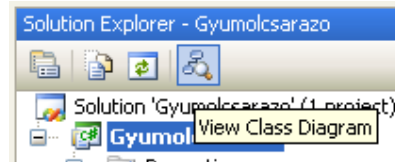
```
/// <summary>
/// Kilépés a programból.
/// </summary>
/// <param name="sender">Az eseményt előidéző
/// nyomógomb</param>
/// <param name="e">Esemény paraméter.</param>
private void btKilép_Click(object sender, EventArgs e)
{ Application.Exit();
}
```

1.1.3. Asszociatív tömb a szerkesztőmező nyomógomb párosokhoz

A későbbi kényelmesebb programozás érdekében az összetartozó szerkesztőmező és nyomógomb referenciákat egy asszociatív tömbben szeretnénk tárolni.

Hozzunk létre egy olyan osztályt, ami asszociatív tömböt valósít meg. Az osztály neve legyen Párosok és származzon a DictionaryBase-ből. A Project menüben válasszuk ki az Add Class pontot, a sablonok közül az elsőt (Class) jelöljük be, majd állománynévként adjuk meg a Párosok.cs-t. A kódszerkesztőben az alábbi mintának megfelelően egészítsük ki az osztály kódját.

Az osztály létrehozásának alternatív módja az, hogy a Solution Explorerben kiválasztjuk a projektet, majd kattintunk a View Class Diagram gombon. Az ekkor megjelenő osztálydiagramban egy üres területen kattintva jobb egérgombbal a gyorsmenüben Add/Add class-t választunk.



```
/// <summary>
/// Asszociatív tömb megvalósítását támogató osztály.
/// </summary>
public class Párosok : DictionaryBase
{ public Párosok()
{
}
/// <summary>
/// Indexelő: feladata megadható/lekérdezhető, hogy
/// egy adott nyomógombhoz melyik szerkesztőmező tartozik.
/// </summary>
```

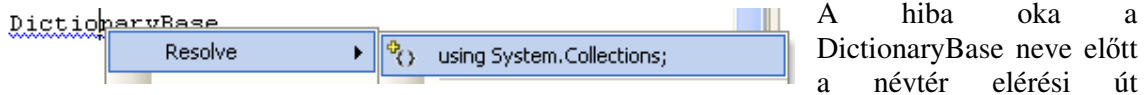
```

/// <param name="kulcs">A nyomógomb neve</param>
/// <returns>A szerkesztőmező neve</returns>
public TextBox this[Button kulcs]
{
    get
    {
        return ((TextBox)Dictionary[kulcs]);
    }
    set
    {
        Dictionary[kulcs] = value;
    }
}
}

```

Fordítsuk le az alkalmazást (Ctrl+Shift+B). Három hibüzenetet kapunk. Az első az alábbi:

The type or namespace name 'DictionaryBase' could not be found (are you missing a using directive or an assembly reference?)



A hiba oka a DictionaryBase neve előtt a névtér elérési út megadásának elmaradása. A kódszerkesztőben a DictionaryBase kék hullámos vonallal van aláhúzva. Kattintsunk jobb egérgombbal rajta, majd a gyorsmenüben válasszuk a Resolve pontot, majd a using ...-t. Ennek eredményeképp az állomány elejére bekerül a megfelelő using direktíva. A másik két hibát is hasonló módon kell megoldani. Mindkettő a

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

direktívát hiányolja.

Hozzunk létre egy Párosok típusú adattagot Szótár néven az ablak osztályában (frmGyümölcsÁrazó) az osztálydiagram és a Class Details segítségével.

```
protected Párosok Szótár;
```

Az ablakosztály konstruktorában adjunk kezdőértéket az egységáraknak, azaz határozzuk meg a négy szövegmező kezdeti tartalmát. Hozzunk létre egy objektumot a Szótár adattaghoz, majd helyezzük el benne az összetartozó nyomógomb és szövegmező párosokat.

```

public frmGyümölcsÁrazó()
{
    InitializeComponent();
    tbAlma.Text = "150";
    tbBanán.Text = "256";
    tbKörte.Text = "190";
    tbNarancs.Text = "199";
    Szótár = new Párosok();
    Szótár[btAlma] = tbAlma;
    Szótár[btBanán] = tbBanán;
    Szótár[btKörte] = tbKörte;
    Szótár[btNarancs] = tbNarancs;
}

```

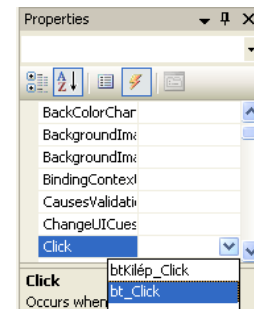
1.1.4. Közös eseménykezelő a négy gyümölcs nyomógombhoz

Készítsünk egy közös eseménykezelőt a négy gyümölcs nyomógombhoz. Az alábbi kódrészletet közvetlenül be kell gépelni az ablak osztályába.

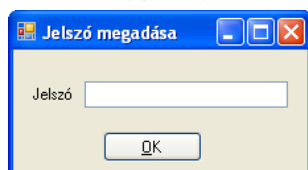
```
/// <summary>
```

```
/// A négy gyümölcsnyomógomb közös eseménykezelője.  
/// Kiszámolja a fizetendő árat, és elhelyezi azt a  
/// megfelelő szövegmezőben.  
/// </summary>  
/// <param name="sender">Az eseményt előidéző  
/// nyomógomb</param>  
/// <param name="e">Esemény paraméter.</param>  
private void bt_Click(object sender, System.EventArgs e)  
{ // Meghatározzuk a gyümölcsnek megfelelő szövegmezőt  
  TextBox tbMunka = Szótár[(Button)sender];  
  try  
  { // Az aktuális egységár  
    double EgységÁr = double.Parse(tbMunka.Text);  
    // Az aktuális tömeg  
    double Tömeg = double.Parse(tbTömeg.Text);  
    if (Tömeg <= 0)  
      throw new ApplicationException("A tömeg értéke "+  
        "negatív vagy nulla!");  
    // A fizetendő ár  
    double Ár = EgységÁr * Tömeg;  
    // Ár kiírása  
    tbÁr.Text = Ár.ToString();  
  }  
  catch (Exception exc)  
  { MessageBox.Show("Hibás adatok! " + exc.Message,  
    "Hiba", MessageBoxButtons.OK, MessageBoxIcon.Error);  
  }  
}
```

Állítsuk be a négy nyomógombhoz a `bt_Click` függvényt eseménykezelőként. Ehhez tervezési nézetben jelöljük ki a négy nyomógombot egyszerre a Shift billentyű segítségével, majd a Properties ablakban kattintsunk a narancssárga villámra emlékeztető ikonra. A listában válasszuk ki a Click eseményt, és a mellette levő üres mezőben kattintsunk. A legördülő listából válasszuk aki a `bt_Click` elemet. Fordítsuk le, és próbáljuk ki a programot.



1.1.5. Egységárok megváltoztatása



Az egységárok megváltoztatását jelszóhoz kívánjuk kötni, ezért hozzunk létre egy új ablakot (Form) a jelszó bevitelhez. Válasszuk a Project menü Add Windows Form... menüpontját, majd a sablonok közül a Windows Form-ot. Az állomány neve legyen `frmJelszoBekero`. Kezdetben az új ablak osztálya is ugyanezt a nevet fogja viselni. A Properties ablak segítségével nevezzük át ékezetesre. Az ablakot a mellékelt ábrának megfelelően alakítsuk ki. A nyomógombnál és a szövegmezőnél a már megismert elnevezési konvenciót alkalmazzuk. A szövegmező esetében a `PasswordChar=*` beállítás szükséges annak érdekében, hogy a jelszó begépelésekor csak csillagok jelenjenek meg. A nyomógomb esetében a `DialogResult=OK` beállítás hatására az OK gombon történő

kattintásra bezárul az ablak. Hozzunk létre egy csak olvasható tulajdonságot az frmJelszóBekérő osztályban Jelszó néven, aminek célja szövegmezőben megadott adatok lekérdezése.

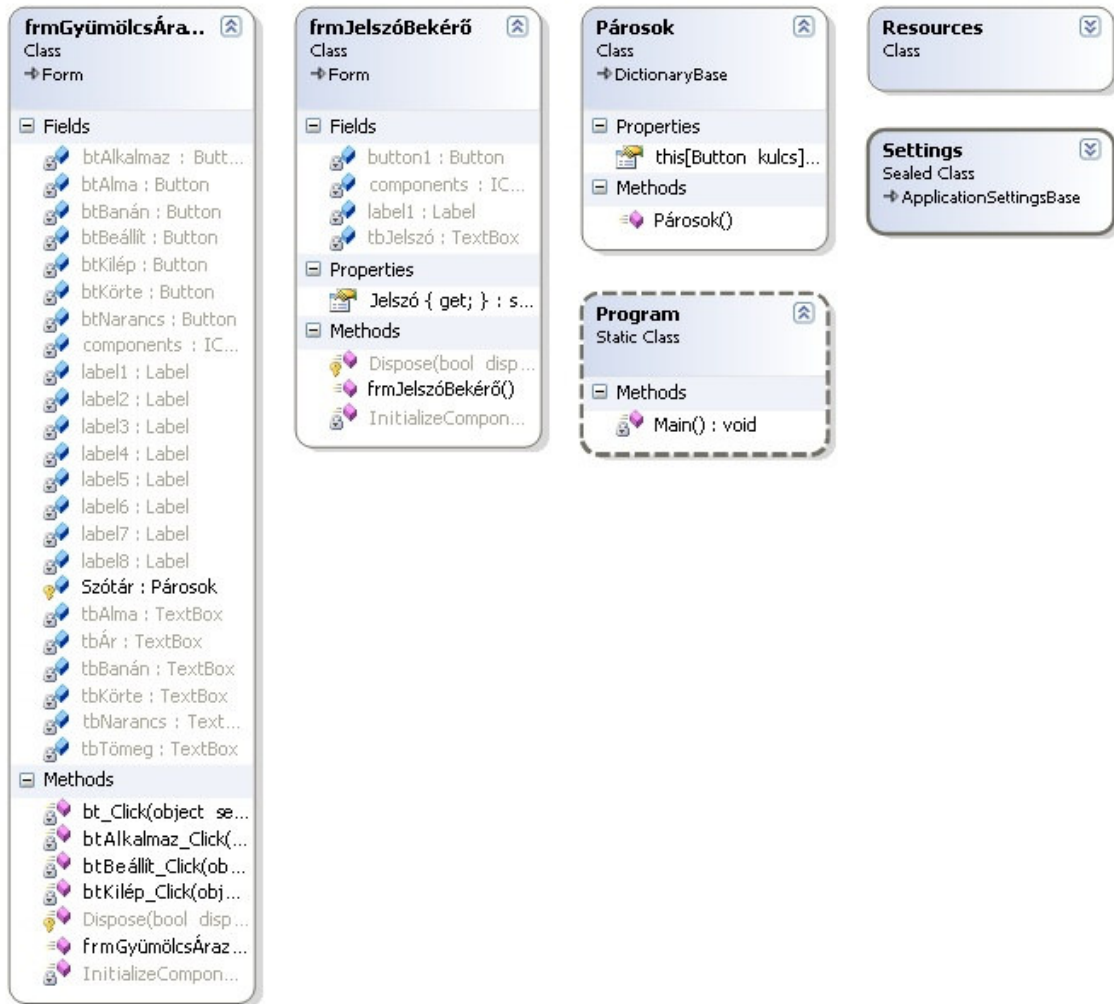
```
/// <summary>
/// Csak olvasható tulajdonság a jelszóbekérő szövegmező
/// tartalmának lekérdezésére.
/// </summary>
public string Jelszó
{
    get
    {
        return tbJelszó.Text;
    }
}
```

Hozzunk létre egy eseménykezelőt a Beállít nyomógombhoz, ami létrehozza és megjeleníti a jelszóbekérő ablakot, és érvényes jelszó (bla-bla) megadása esetén írhatóvá teszi az egységár szövegmezőket, majd engedélyezi az Alkalmaz nyomógombot és letiltja a Beállít nyomógombot.

```
/// <summary>
/// Bekéri a jelszót, és egyezés esetén írhatóvá teszi
/// az egységár szövegmezőket.
/// </summary>
/// <param name="sender">Az eseményt előidéző
/// nyomógomb</param>
/// <param name="e">Esemény paraméter.</param>
private void btBeállít_Click(object sender, EventArgs e)
{
    // Jelszóbekérő ablak objektum létrehozása
    frmJelszóBekérő JSz = new frmJelszóBekérő();
    // Jelszóbekérő ablak megjelenítése
    DialogResult Dr = JSz.ShowDialog();
    // Ha az OK nyomógombbal zárta be a felhasználó az ablakot
    if (Dr == DialogResult.OK)
    {
        // Jelszó ellenőrzése
        if (JSz.Jelszó == "bla-bla")
        {
            // A szövegmezők írhatóvá tétele
            tbAlma.ReadOnly = false;
            tbKörte.ReadOnly = false;
            tbNarancs.ReadOnly = false;
            tbBanán.ReadOnly = false;
            // Beállít nyomógomb letiltása
            btBeállít.Enabled = false;
            // Alkalmaz nyomógomb engedélyezése
            btAlkalmaz.Enabled = true;
        }
        else
        {
            MessageBox.Show("Hibás jelszó! ",
                "Hiba", MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }
}
```

Készítsünk egy eseménykezelőt az Alkalmaz nyomógombhoz, ami ellenőrzi a megadott egységárakat, és ha minden rendben, akkor újból csak olvashatóvá teszi a megfelelő szövegmezőket, engedélyezi a Beállít nyomógombot, majd letiltja az Alkalmaz nyomógombot.

```
/// <summary>
/// Leellenőrzi a megadott egységárakat, és ha rendben
/// vannak, akkor újból csak olvashatóvá teszi az
/// egységár szövegmezőket.
/// </summary>
/// <param name="sender">Az eseményt előidéző
/// nyomógomb</param>
/// <param name="e">Esemény paraméter.</param>
private void btAlkalmaz_Click(object sender, EventArgs e)
{ try
  { double Egységár=double.Parse(tbAlma.Text);
    if (Egységár <= 0)
      throw new Exception("Az alma ára negatív");
    tbAlma.ReadOnly=true;
    Egységár=double.Parse(tbKörte.Text);
    if (Egységár <= 0)
      throw new Exception("A körte ára negatív");
    tbKörte.ReadOnly=true;
    Egységár=double.Parse(tbBanán.Text);
    if (Egységár <= 0)
      throw new Exception("Az banán ára negatív");
    tbBanán.ReadOnly=true;
    Egységár=double.Parse(tbNarancs.Text);
    if (Egységár <= 0)
      throw new Exception("A narancs ára negatív");
    tbNarancs.ReadOnly=true;
    // Beállít nyomógomb engedélyezése
    btBeállít.Enabled=true;
    // Alkalmaz nyomógomb letiltása
    btAlkalmaz.Enabled=false;
  }
  catch(Exception exc)
  { MessageBox.Show("Hibás adatok! "+exc.Message,
    "Hiba", MessageBoxButtons.OK, MessageBoxIcon.Error);
  }
}
```



2. Házi feladat

Készítsünk egy grafikus felületű alkalmazást a másodfokú egyenlet gyökeinek kiszámítására. Ha csak egy gyök van, akkor az $x_2=$ és a mellette levő szövegmező ne legyen látható. Ha a gyökök nem valósak, akkor $a+b*i$ formában jelenjenek meg.

