

Cukrászda ¹

Minden városban megtalálható legalább egy cukrászda, amelyet életre szólóan megőröznk emlékezetünkben. Megmarad a fagyalt íze, a sütemények illata. Ebben a feladatban egy cukrászda süteménykínálatával foglalkozunk. A feliratok felejenek meg a mintának, de a programot tetszés szerint dekorálhatja.

1. Készítsen grafikus applikációt *cukraszda* néven! A mellékelt *cuki.txt* pontosvesszővel tagolt, UTF-8 kódolású szöveges állományt olvassa be egy megfelelő adatszerkezetbe! (Osztály és lista)
Pl.: Sajttorta (málnás);torta;false;9000;12 szeletes

Az állomány tartalma:

- A sütemény neve (szöveg)
 - A sütemény típusa (szöveg)
 - A sütemény díjazott volt-e a Magyarország Tortája versenyen (logikai) true/false
 - A sütemény ára (egész szám)
 - Annak az egységnek a neve, amelyben a süteményt értékesítik (szöveg)
2. Válasszon ki véletlen szám generátor segítségével egy sütit és jelenítse meg nevét a képernyőn a minta szerint!
 3. Jelenítse meg a képernyőn a minta szerint a legdrágább és legolcsóbb süti nevét és árát!
 4. Hány féle Magyarország Tortája versenyen díjazott finomság közül választhatnak a vásárlók? (Ha az egység eltérő, akkor a sütemény különbözőnek számít.)
 5. Készítsen listát a cukrászdában kapható süteményekről! A süti nevét és típusát szóközzel elválasztva írja ki a lista.txt állományba! Ügyeljen rá, hogy minden süti neve csak egyszer szerepeljen!

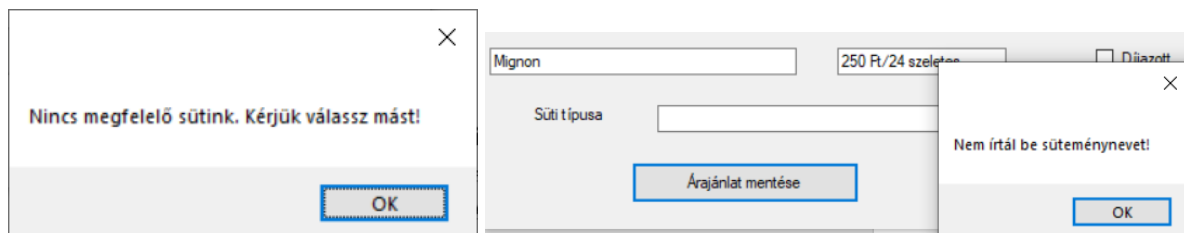
```
Süni vegyes  
Gesztenyealagút vegyes  
Sajtos pogácsa sós teasütemény  
Diós-mákos bejgli  
Sajttorta (málnás) torta  
Citrom torta  
Eszterházy tortaszelet  
Rákóczi-túrós pite  
Meggyes kocka tejszínes sütemény  
Legényfogó torta  
Alpesi karamell tortaszelet  
Kókuszcsók édes teasütemény  
Habos mákos pite  
Szilvás pite  
Juhtúrós párna sós teasütemény  
Mákos guba tortaszelet  
Néró édes teasütemény  
Sacher tortaszelet  
Ribizlihabos-almás réteges különleges torta  
Három kívánság torta  
Dobos torta
```

¹ **Forrás:** <https://nandori.hu/arlista> Utolsó megtekintés: 2018.12.25

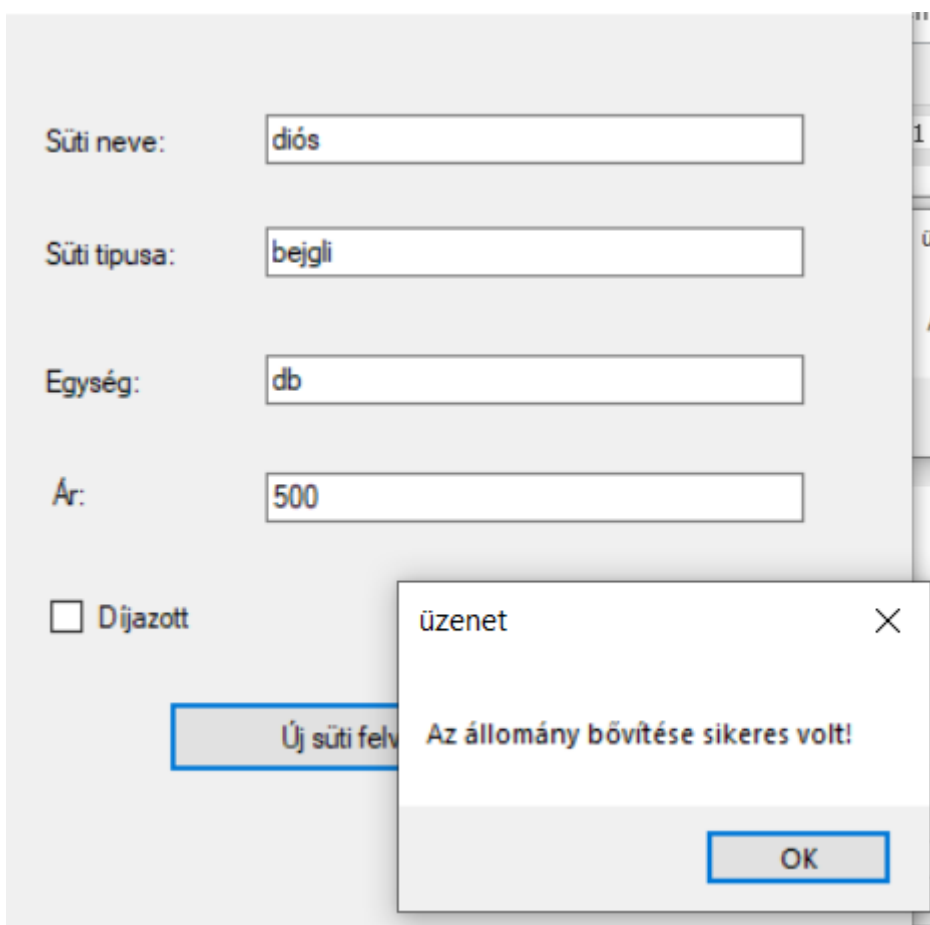
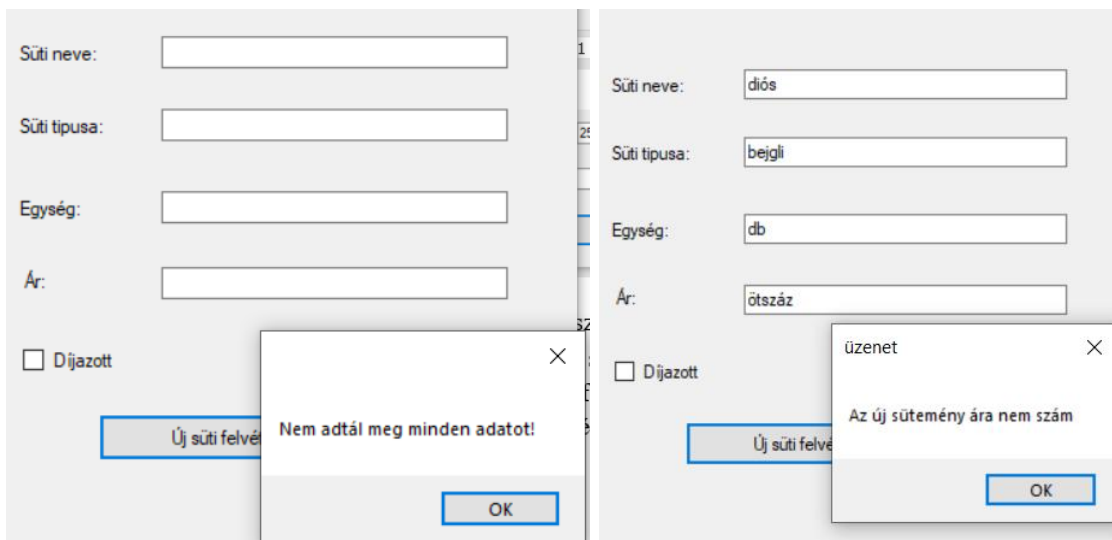
6. Hány fajta sütit árul a cukrászda az egyes édességtípusokból? Hozza létre a stat.csv állományt! Írja ki a süti típusát és a sütifajták számát pontosvesszővel elválasztva az állományba!

```
stat.csv – Jegyzetömb
Fájl Szerkesztés Formátum Néze
vegyes;14
sós teasütemény;13
bejgli;7
torta;107
tortaszelet;27
pite;10
tejszínes sütemény;7
édes teasütemény;17
különleges torta;4
```

7. Egy megfelelő objektum segítségével olvasson be egy sütitípust! Árajánlat mentése nyomógomra kattintás hatására az ajanlat.txt állományba írja ki az összes felhasználó által megadott típusnak megfelelő sütemény nevét, árát és egységét! Amennyiben sikeres volt a file-ba írás, előugró ablakon jelenítse meg, hogy hány db sütit írt az állományba és mennyi ezek átlagára! Ha nincs olyan süti, ami megfelel a keresésnek, akkor az előugró ablakban a „Nincs megfelelő sütink. Kérjük, válasszon mást!” felirat jelenjen meg!



8. Az „Új süti felvétele” nyomógomb hatására adja hozzá adatszerkezetéhez és a *cuki.txt* állományhoz a felhasználó által a bementi vezérlő elemek segítségével megadott sütit! Amennyiben valamelyik adat hiányzik vagy nem megfelelő formátumú, felugró ablakban küldjön hibaüzenetet, sikeres mentés esetén is ilyen módon tájékoztassa a felhasználót!



Minta a feladathoz:

Megoldás:

```
using System;
using System.IO;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
/*1. Készítsen grafikus applikációt cukraszda néven!
* A mellékelt cuki.txt pontos vesszővel tagolt, UTF-8 kódolású szöveges állományt olvassa
be
* egy megfelelő adatszerkezetbe! (Osztály és lista)
```

Pl.: Sajttorta (málnás);torta;false;9000;12 szeletes

Az állomány tartalma:

- A sütemény neve (szöveg)
- A sütemény típusa (szöveg)
- A sütemény díjazott volt-e a Magyarország Tortája versenyen (logikai) true/false
- A sütemény ára (egész szám)
- Annak az egységnek a neve, amelyben a süteményt értékesítik (szöveg)

*/

```
namespace cukraszdagui
```

```
{
```

```
    public partial class Form1 : Form
```

```
    {
```

```
        struct sutik
```

```
        {
```

```
            public string nev;
```

```
            public string tipus;
```

```

    public bool dij;
    public int ar;
    public string egyseg;
}
static sutik[] adatok = new sutik[300];
string[] fajlbol = File.ReadAllLines("cuki.txt");

int sorokszama = 0;//sorok száma a fájlban
int i,j;//ciklusváltozó
Random n = new Random();

public Form1()
{
    InitializeComponent();
    adatokbetoltese();
}
private void adatokbetoltese()
{
    for (int k = 0; k < fajlbol.Count(); k++)
    {
        string[] egysordarabolva = fajlbol[k].Split(';');
        adatok[sorokszama].nev = egysordarabolva[0];
        adatok[sorokszama].tipus = egysordarabolva[1];
        if (egysordarabolva[2] == "false") adatok[sorokszama].dij = false;
        if (egysordarabolva[2] == "true") adatok[sorokszama].dij = true;
        adatok[sorokszama].ar = Convert.ToInt32(egysordarabolva[3]);
        adatok[sorokszama].egyseg = egysordarabolva[4];
        sorokszama++;
    }
    int sutikszama = sorokszama;
    int kivalaszt = n.Next(0, sutikszama);
    /*2.   Válasszon ki véletlen szám generátor segítségével egy sütit
    * és jelenítse meg nevét a képernyőn a minta szerint!*/
    kivalasztottsuti.Text = "Mai ajánlatunk: " + adatok[kivalaszt].nev;
    /*3.   Jelenítse meg a képernyőn a minta szerint a legdrágább
    * és legolcsóbb süti nevét és árát!*/
    int max = adatok[0].ar;
    int min = adatok[0].ar;
    int maxi = 0, mini = 0;
    for (i = 0; i < sutikszama; i++)
    {
        if (adatok[i].ar > max) { max = adatok[i].ar;maxi = i; }
        if (adatok[i].ar < min) { min = adatok[i].ar;mini = i; }
    }
    legdragabbsutinev.Text = adatok[maxi].nev;
    legolcsobbsutinev.Text = adatok[mini].nev;
    legdragabbsutiar.Text = Convert.ToString(adatok[maxi].ar) + " Ft/" +
    adatok[maxi].egyseg;
    legolcsobbsutiar.Text = Convert.ToString(adatok[mini].ar) + " Ft/" +
    adatok[maxi].egyseg;
    /*4.   Hány féle Magyarország Tortája versenyen díjazott finomság közül
    választhatnak a vásárlók?
    * (Ha az egység eltérő, akkor a sütemény különbözőnek számít.)*/

```

```

int db = 0;
for (i = 0; i < sutikszama; i++)
{
    if (adatok[i].dij == true)
        db++;
}
dijnyertessutik.Text = Convert.ToString(db) + " féle díjnyertes édességből
választhat.";
/*5. Készítsen listát a cukrászdában kapható süteményekről!
* A süti nevét és típusát szóközzel elválasztva írja ki a lista.txt állományba!
* Ügyeljen rá, hogy minden süti neve csak egyszer szerepeljen!*/
int kulonbozoelemekszama = 0;
string[] sutemenyeknevei = new string[250];
string[] tipusoknevei = new string[250];

for (i = 0; i < sutikszama; i++)
{
    j = 0;
    while ((j <= kulonbozoelemekszama) && (adatok[i].nev != sutemenyeknevei[j]))
    {
        j++;
    }
    if (j > kulonbozoelemekszama)
    {
        kulonbozoelemekszama++;
        sutemenyeknevei[kulonbozoelemekszama] = adatok[i].nev;
        tipusoknevei[kulonbozoelemekszama] = adatok[i].tipus;
    }
}
FileStream fnev = new FileStream("lista.txt", FileMode.Create);
StreamWriter fajlbairo = new StreamWriter(fnev);

for (i = 0; i < kulonbozoelemekszama; i++)
{
    fajlbairo.Write("{0} ", sutemenyeknevei[i]);
    fajlbairo.WriteLine("{0}", tipusoknevei[i]);
}
fajlbairo.Close();
fnev.Close();

```

```

/*6. Hány fajta sütit árul a cukrászda az egyes édességtípusokból?
* Hozza létre a stat.csv állományt! Írja ki a süti típusát
* és a sütifajták számát pontosvesszővel elválasztva az állományba!*/

```

```

kulonbozoelemekszama = 0;
string[] sutitipusok = new string[200];
int[] tipusokszama = new int[100];
for (i = 0; i < sutikszama; i++)
{
    j = 0;
    while ((j <= kulonbozoelemekszama) && (adatok[i].tipus != sutitipusok[j]))
    {
        j++;
    }
}

```

```

    }
    if (j > kulonbozoelemekszama)
    {
        kulonbozoelemekszama++;
        sutitipusok[kulonbozoelemekszama] = adatok[i].tipus;
    }
}
//megszámlálás tétele
for (i = 0; i < sutikszama; i++)
{
    for (j = 1; j <= kulonbozoelemekszama; j++)

        {
            if (sutitipusok[j] == adatok[i].tipus) tipusokszama[j]++;
        }

}
FileStream fnev2 = new FileStream("stat.csv", FileMode.Create);
StreamWriter fajlbairo2 = new StreamWriter(fnev2);

for (i = 1; i < kulonbozoelemekszama; i++)
{
    fajlbairo2.Write("{0};", sutitipusok[i]);
    fajlbairo2.WriteLine("{0}", tipusokszama[i]);
}
fajlbairo2.Close();
fnev2.Close();
}

private void button2_Click(object sender, EventArgs e)
{
    /*8. Az „Új süti felvétele” nyomógomb hatására adja hozzá adatszerkezetéhez és a
cuki.txt
* állományhoz a felhasználó által a bementi vezérlő elemek segítségével megadott
sütit!
* Amennyiben valamelyik adat hiányzik vagy nem megfelelő formátumú, felugró
ablakban küldjön hibaüzenetet,
* sikeres mentés esetén is ilyen módon tájékoztassa a felhasználót!*/
    if (ujsutinev.Text != "" && ujsutitipus.Text != "" && ujsutiegység.Text != "" &&
    ujsutiar.Text != "")
    {
        try
        {
            string ujnev = ujsutinev.Text;
            string ujtipus = ujsutitipus.Text;
            string ujegység = ujsutiegység.Text;
            int ujar = int.Parse(ujsutiar.Text);
            bool ujdij = false;
            if (checkBox1.Checked) ujdij = true; else ujdij = false;
            FileStream fnev = new FileStream("cuki.txt", FileMode.Append);
            StreamWriter fajlbairo = new StreamWriter(fnev, System.Text.Encoding.UTF8);
            fajlbairo.WriteLine("{0};{1};{2};{3};{4}", ujnev,ujtipus,ujdij,ujar,ujegység );
            fajlbairo.Close();

```

```

        fnev.Close();
        MessageBox.Show("Az állomány bővítése sikeres volt!", "üzenet");
    }
    catch
    {
        MessageBox.Show("Az új sütemény ára nem szám", "üzenet");
    }

}
else
{
    MessageBox.Show("Nem adtál meg minden adatot!");
}
}

private void button1_Click(object sender, EventArgs e)
{
    /*7. Egy megfelelő objektum segítségével olvasson be egy sütitípust!
    * Árajánlat mentése nyomógomra kattintás hatására az ajanlat.txt állományba írja ki
    * az összes felhasználó által megadott típusnak megfelelő sütemény nevét, árát és
egységét!
    * Amennyiben sikeres volt a file-ba írás, előugró ablakon jelenítse meg,
    * hogy hány db sütit írt az állományba és menni ezek átlagára! Ha nincs olyan süti,
    * ami megfelel a keresésnek, akkor az előugró ablakban a „Nincs megfelelő sütiink.
    * Kérjük, válasszon mást!” felirat jelenjen meg!*/
    if (sutitipusa.Text != "")
    {
        string keresettipus = sutitipusa.Text;
        bool van = false;
        i = 0;
        int sutikszama = fajlbol.Count();
        while (i < sutikszama && adatok[i].tipus != keresettipus)
        {
            i++;
        }
        if (i < sutikszama)
        {
            FileStream fnev3 = new FileStream("ajanlat.txt", FileMode.Create);
            StreamWriter fajlbairo3 = new StreamWriter(fnev3);
            int keresettsutikszama = 0;
            double atlagar = 0;
            for (i = 0; i < sutikszama; i++)
            {
                if (adatok[i].tipus == keresettipus)
                {
                    keresettsutikszama++;
                    atlagar += adatok[i].ar;
                    fajlbairo3.Write("{0};", adatok[i].nev);
                    fajlbairo3.Write("{0};", adatok[i].ar);
                    fajlbairo3.WriteLine("{0}", adatok[i].egyseg);
                }
            }
        }
    }
}

```



```
        atlagar = atlagar / keresettsutikszama;
        fajlbairo3.Close();
        fnev3.Close();
        MessageBox.Show(keresettsutikszama+" db sütit írtam az ajanlat.txt-be\nátlagár:
"+ atlagar+" Ft");
    }
    else
    {
        MessageBox.Show("Nincs megfelelő sütink. Kérjük válassz mást!");
    }

}
else
{
    MessageBox.Show("Nem írtál be süteménynevet!");
}
}
}
```