

## Bérek 2020 Programozásfeladat

Ebben a feladatban egy cég dolgozóinak 2020-as adatai<sup>1</sup> állnak rendelkezésünkre, melyekkel programozási feladatokat kell megoldania.

A feladat megoldása során vegye figyelembe a következőket:

1. A képernyőre írást igénylő<sup>1</sup> részfeladatok eredményének megjelenítése előtt írja a képernyőre a feladat sorszámát (például:3. feladat:)!
2. Az egyes feladatokban a kiírásokat a minta szerint készítse el!
3. Az ékezetmentes azonosítók és kiírások is elfogadottak.
4. Az azonosítókat kis- és nagybetűkkel is kezdheti.
5. A program megírásakor az állományban lévő adatok helyes szerkezetét nem kell ellenőriznie, feltételezheti, hogy a rendelkezésre álló adatok a leírtaknak megfelelnek.
6. Megoldását úgy készítse el, hogy az azonos szerkezetű, de tetszőleges bemeneti adatok mellett is helyes eredményt adjon!

A berek2020.txt UTF-8 kódolású forrásállomány soraiban egy-egy dolgozó adatait tároltuk a következő sorrendben: •neve, például: Beri Dániel •neme: nő vagy férfi •a részleg, ahol dolgozik, például: beszerzés •a belépés éve, például: 1979 •a dolgozó bére (fizetése), például: 222 943

Az állomány első sora a mezőneveket tartalmazza, az adatokat pontosvesszővel választottuk el:

**Név;Neme;Részleg;Belépés;Bér**

**Beri Dániel;férfi;beszerzés;1979;222943**

**Csavar Pista;férfi;pénzügy;1995;234074**

1. Készítsen grafikus vagy konzolalkalmazást (projektet) a következő feladatok megoldásához, amelynek projektjét **berek2020** néven mentse el!
2. Olvassa be a **berek2020.txt** állomány sorait és tárolja az adatokat egy olyan összetett adatszerkezetben, amely használatával a további feladatok megoldhatók! Ügyeljen arra, hogy az állomány első sora az adatok fejlécét tartalmazza!
3. Határozza meg és írja ki a képernyőre, hogy hány dolgozó adatai találhatóak a forrásállományban!
4. Határozza meg és írja ki a képernyőre a 2020-as átlagbéreket! Az eredményt ezer forintban, egy tizedesjegyre kerekítve jelenítse meg!
5. Kérje be egy részleg nevét a felhasználtól a minta szerint!
6. Az előző feladatban megadott részlegben keresse meg és írja ki a legnagyobb bérrel (fizetéssel) rendelkező dolgozó adatait! Ha a megadott részleg nem létezik a cégnél, akkor a „A megadott részleg nem létezik a cégnél!” feliratot jelenítse meg! Feltételezheti, hogy nem alakult ki „holtverseny” egy-egy részlegben dolgozók fizetése között!
7. Készítsen statisztikát az egyes részlegeken dolgozók számáról! A részlegek kiírásának sorrendje tetszőleges!

Képernyőminta: (Grafikus változat)

---

<sup>1</sup> Adatok forrása: <https://www.komal.hu/feladat?a=honap&h=201901 &t=inf&l=hu>

Form1

Dolgozók száma: 170 fő

Ben Dániel	férfi	beszerzés	1979	222943	
Csavar Pista	férfi	penzügy	1995	234074	
Lakatos Pál	férfi	beszerzés	1986	159538	
Devon Mihály	férfi	asztalosműhely	2007	161533	
El Ilona	nő	beszerzés	1982	299865	
Gábor Gizella	nő	értékesítés	2003	181108	
Kálvin Odómer	férfi	lakatosműhely	1976	321959	
Kolompár Gáspár	férfi	asztalosműhely	2003	253794	
Sörös Sándor	férfi	lakatosműhely	2012	202216	
Léc Elek	férfi	asztalosműhely	2001	220506	

4. Feladat: Bérék átlaga: 250,3 eFt

Kérem a részleg nevét:

6. Feladat: A legtöbbet kereső dolgozó a megadott részlegen  
Név: Czeczeli Zsolt  
Nem: férfi  
Belépés: 1981  
Bér: 452042 forint

7. Feladat: Statistika

beszerzés - 17 fő  
penzügy - 36 fő  
asztalosműhely - 18 fő  
értékesítés - 16 fő  
lakatosműhely - 15 fő  
karbantartás - 26 fő  
szerelőműhely - 30 fő  
személyzeti - 12 fő

Képernyőminta, ha a megadott részleg nem létezik:

Form1

Dolgozók száma: 170 fő

Ben Dániel	férfi	beszerzés	1979	222943	
Csavar Pista	férfi	penzügy	1995	234074	
Lakatos Pál	férfi	beszerzés	1986	159538	
Devon Mihály	férfi	asztalosműhely	2007	161533	
El Ilona	nő	beszerzés	1982	299865	
Gábor Gizella	nő	értékesítés	2003	181108	
Kálvin Odómer	férfi	lakatosműhely	1976	321959	
Kolompár Gáspár	férfi	asztalosműhely	2003	253794	
Sörös Sándor	férfi	lakatosműhely	2012	202216	
Léc Elek	férfi	asztalosműhely	2001	220506	

4. Feladat: Bérék átlaga: 250,3 eFt

Kérem a részleg nevét:

6. Feladat: A megadott részleg nem létezik a cégnél!

7. Feladat: Statistika

beszerzés - 17 fő  
penzügy - 36 fő  
asztalosműhely - 18 fő  
értékesítés - 16 fő  
lakatosműhely - 15 fő  
karbantartás - 26 fő  
szerelőműhely - 30 fő  
személyzeti - 12 fő

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;

namespace berek2020GUI
{
    public partial class Form1 : Form
    {
        class Dolgozok
        {
            public Dolgozok(string sor)
            {

```

```

        string[] sorelemek = sor.Split(';');
        this.Nev = sorelemek[0];
        this.Nem = sorelemek[1];
        this.Reszleg = sorelemek[2];
        this.Belepés = Convert.ToInt32(sorelemek[3]);
        this.Ber = Convert.ToInt32(sorelemek[4]);
    }
    //Név;Neme;Részleg;Belépés;Bér
    public string Nev { get; set; }
    public string Nem { get; set; }
    public string Reszleg { get; set; }
    public int Belepés { get; set; }
    public int Ber { get; set; }
}
List<Dolgozok> dolgozoadatok = new List<Dolgozok>();

string bekertReszleg = "";
public Form1()
{
    InitializeComponent();
}

private void Form1_Load(object sender, EventArgs e)
{
    label4.Text = "";
    /*2. Olvassa be a berek2020.txt állomány sorait és tárolja az adatokat egy olyan
összetett adatszerkezetben,
* amely használatával a további feladatok megoldhatók!
* Ügyeljen arra, hogy az állomány első sora az adatok fejlécét tartalmazza!*/
    StreamReader olvas = new StreamReader("berek2020.txt", Encoding.UTF8);
    string fejléc = olvas.ReadLine();//ha van fejléc
    while (!olvas.EndOfStream)
    {
        dolgozoadatok.Add(new Dolgozok(olvas.ReadLine()));
    }

    string sorok = "";
    for (int i = 0; i < dolgozoadatok.Count; i++)
    {
        sorok = dolgozoadatok[i].Nev + "\t" + dolgozoadatok[i].Nem + "\t" +
dolgozoadatok[i].Reszleg + "\t" + dolgozoadatok[i].Belepés + "\t" + dolgozoadatok[i].Ber;
        listBox1.Items.Add(sorok);
    }
    label1.Text = "Dolgozók száma: " + dolgozoadatok.Count + " fő";
    label2.Text = "4. Feladat: Bérek átlaga: " + Math.Round((double)dolgozoadatok.Average(x =>
x.Ber) / 100,1) + " eFt";

    dolgozoadatok.GroupBy(x => x.Reszleg).ToList().ForEach(x => listBox2.Items.Add(x.Key + " -
"+x.Count() + " fő"));
}

private void button1_Click(object sender, EventArgs e)
{
    if(textBox1.Text!="")
    {
        try
        {
            bekertReszleg = textBox1.Text;
            int index = dolgozoadatok.IndexOf(dolgozoadatok.Where(x => x.Reszleg ==
bekertReszleg).OrderByDescending(x => x.Ber).First());
            label4.Text = "6. Feladat: A legtöbbet kereső dolgozó a megadott részlegen\nNév:
" +
                dolgozoadatok[index].Nev + "\nNem: " + dolgozoadatok[index].Nem+
                "\nBelépés: " + dolgozoadatok[index].Belepés +
                "\nBér: " + dolgozoadatok[index].Ber+ " forint";
        }
        catch (InvalidOperationException)
        {

```

```
        label4.Text = "6. Feladat: A megadott részleg nem létezik a cégnél!";
    }
}
else
{
    MessageBox.Show("Kérem a részleg nevét beírni!");
}
}

private void textBox1_TextChanged(object sender, EventArgs e)
{
    label4.Text = "";
}
}
}
```