



Java Programozás

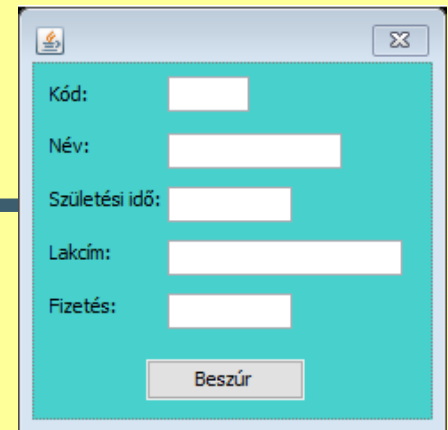
3. óra



BITMAN

v: 2020.03.21

Új adatsor beszúrása a fájlba



1. Panel létrehozása:

New\Swing\JDialog – Neve: NewEmp

– A *Generate JDialog with OK and Cancel buttons* opciót kapcsoljuk ki!

2. Panelen az elrendezés menedzser kikapcsolása:

Jobb klikk a panel közepére\Set layout\Absolute layout

3. Feliratok (JLabel) és szövegmezők (JTextField) felrakása

A feliratok változóneve lényegtelen

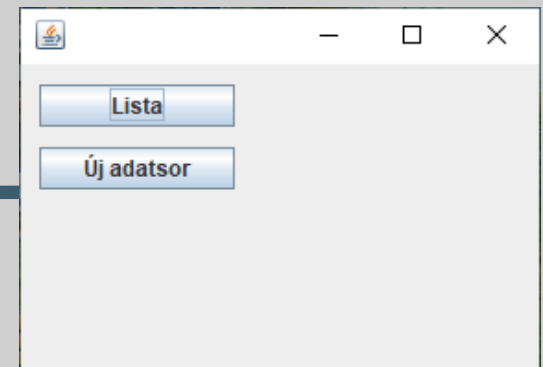
A szövegmezők változóneve rendre: kod, nev, szid, lak, fiz

4. Készítsük el e Beszúr (btnBeszur) gombot, aztán adjunk hozzá eseménykezelőt.

A Program módosítása

- Szereljük egy új nyomógombot a panelre, a változó neve legyen: `btnUjAdat`
- Adjunk hozzá `eseménykezelőt`
- Írjuk meg a kódját:

```
JButton btnUjAdat = new JButton("\u00DAj adatsor");  
btnUjAdat.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent e) {  
        NewEmp ne = new NewEmp();  
        ne.setVisible(true);  
    }  
});
```



A beszúráshoz szükséges metódus

A beszúrás metódusa a **FileManager** osztályba kerül:

```
public static void Insert(String kod, String nev, String szid, String lak, String fiz) {  
    String x="";  
    try {  
        PrintStream out = new PrintStream(new FileOutputStream("adatok.csv", true));  
        out.println(kod+x+nev+x+szid+x+lak+x+fiz);  
        out.close();  
        SM("Adatok kiírva!", 1);  
    } catch (IOException ioe) {  
        SM("CsvWriter: "+ioe.getMessage(), 0);  
    }  
}
```

```
public static void SM(String msg, int tipus) {  
    JOptionPane.showMessageDialog(null, msg, "Program üzenet", tipus);  
}
```

A NewEmp osztály kódjának módosítása

A kódból töröljük a **main** függvényt!

```
public class NewEmp extends JDialog {
    private JTextField kod;
    private JTextField nev;
    private JTextField szid;
    private JTextField lak;
    private JTextField fiz;

    /**
     * Launch the application.
     */
    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    NewEmp dialog = new NewEmp();
                    dialog.setDefaultCloseOperation(JDialog.DISPOSE_ON_CLOSE);
                    dialog.setVisible(true);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }

    /**
     * Create the dialog.
     */
    public NewEmp() {
        getContentPane().setBackground(new Color(72, 209, 204));
        setBounds(100, 100, 241, 240);
    }
}
```

A NewEmp osztály kódjának módosítása

A **Beszúr** gomb kódja, és egy segédmetódus (**RTF**), mely kiolvassa egy szövegmező tartalmát:

```

    JButton btnBeszr = new JButton("Besz\u00FAr");
    btnBeszr.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            ➡    FileManager.Insert(RTF(kod), RTF(nev), RTF(szip), RTF(lak), RTF(fiz));
        }
    });
    btnBeszr.setBounds(64, 167, 89, 23);
    getContentPane().add(btnBeszr);
}

➡ public String RTF(JTextField jtf) {
    return jtf.getText();
}
}
```

Nincsenek a mezőkbe beírt adatok ellenőrizve. Módosítsuk a kódot!

A NewEmp osztály kódjának módosítása

- Két dolgot kell ellenőrizni:
 - Van-e adat (beírt szöveg) a mezőben
 - A numerikus adatok numerikusak-e
- Hol legyenek a metódusok?
 - Mivel több osztályban (pl. módosítás) is szükségesek lehetnek, tegyük őket egy új osztályba
- Hozzunk létre egy Checker nevű osztályt

A Checker osztály kódja

```
public class Checker {  
  
    public boolean filled(JTextField a, String an) {  
        String s = RTF(a);  
        if (s.length() > 0) return true;  
        ① else {  
            SM("Hiba: a(z) "+an+" mező üres!", 0);  
            return false;  
        }  
    }  
  
    public boolean goodInt(JTextField a, String an) {  
        String s = RTF(a);  
        boolean b = filled(a, an);  
        ② if (b) try {  
            Integer.parseInt(s);  
        } catch (NumberFormatException e){  
            SM("A(z) "+an+" mezőben hibás a számadat!", 0);  
            b = false;  
        }  
        return b;  
    }  
}
```

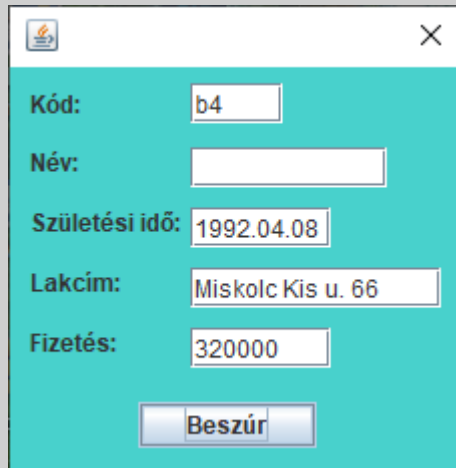

A Checker osztály kódja

```
3 public String RTF(JTextField jtf) {  
    return jtf.getText();  
}  
  
4 public void SM(String msg, int tipus) {  
    JOptionPane.showMessageDialog(null, msg, "Program üzenet", tipus);  
}  
}
```

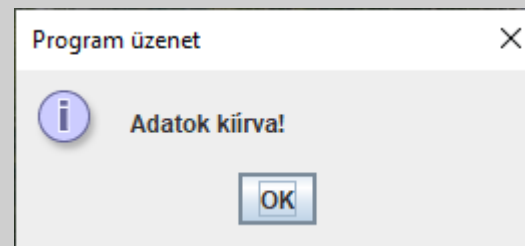
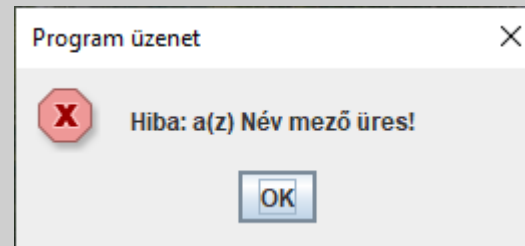
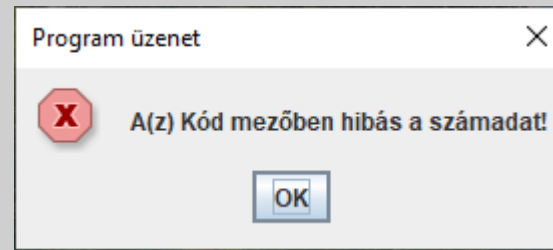
A NewEmp osztály kódjának módosítása

```
JButton btnBeszr = new JButton("Besz\u00FAr");
btnBeszr.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        if (c.goodInt(kod, "Kód"))
            if (c.filled(nev, "Név"))
                if (c.filled(szid, "Születési idő"))
                    if (c.filled(lak, "Lakcím"))
                        if (c.goodInt(fiz, "Fizetés"))
                            FileManager.Insert(RTF(kod), RTF(nev),
                                RTF(szid), RTF(lak), RTF(fiz));
    }
});
```

A NewEmp osztály kipróbálása



A screenshot of a Windows application window titled 'NewEmp'. The window has a teal background and contains several text input fields with labels: 'Kód:' (containing 'b4'), 'Név:' (empty), 'Születési idő:' (containing '1992.04.08'), 'Lakcím:' (containing 'Miskolc Kis u. 66'), and 'Fizetés:' (containing '320000'). At the bottom right is a button labeled 'Beszúr'.



A NewEmp – egy kis extra

- Nem ártana leellenőrizni azt, hogy a dátum mezőbe beírt szöveg valóban dátum-e! Oldjuk meg!
- Írjunk metódust a **Checker** osztályba a **dátum** ellenőrzésére:

```
public boolean DateFormatChecker(String SDate) {  
    try {  
        Date date = RDF.parse(SDate);  
        return true;  
    } catch (ParseException ef) {return false;}  
}  
  
public boolean goodDate(JTextField a, String an) {  
    String s = RTF(a);  
    boolean b = filled(a, an);  
    if (b && DateFormatChecker(s)) return true;  
    else {  
        SM("A(z) "+an+" mezőben hibás a dátum!", 0);  
        return false;  
    }  
}
```

A NewEmp – egy kis extra

- A DateFormatChecker metódus működéséhez szükséges, hogy a **Checker** osztályban létrehozzuk az **RDF** adattagot:

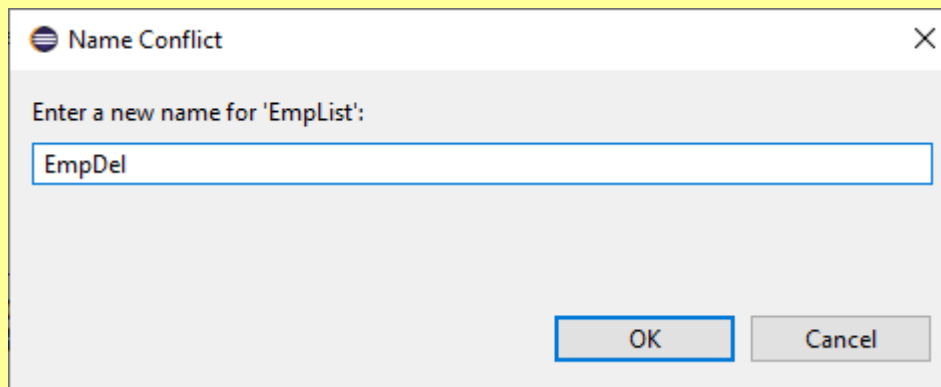
```
private SimpleDateFormat RDF = new SimpleDateFormat("yyyy.MM.dd");
```

- A beszúrás gomb kódja a NewEmp osztályban a dátum ellenőrzésével:

```
JButton btnBeszr = new JButton("Besz\u00FAr");  
btnBeszr.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent e) {  
        if (c.goodInt(kod, "Kód"))  
            if (c.filled(nev, "Név"))  
                if (c.goodDate(szip, "Születési idő"))  
                    if (c.filled(lak, "Lakcím"))  
                        if (c.goodInt(fiz, "Fizetés"))  
                            FileManager.Insert(RTF(kod), RTF(nev),  
                                RTF(szip), RTF(lak), RTF(fiz));  
    }  
});
```

A törlés

- A legegyszerűbb módszerelni egy **Törlés** gombot a **Lista** panelre. Ezért másoljuk le az **EmpList** panelt, legyen az új neve: **EmpDel**.
- Jelöljük ki a projektünkben az **EmpList.java** osztályt (egyszerűen kattintsunk rá)
- Adjuk ki a **CTRL** + **C**, majd a **CTRL** + **V** parancsokat
- A megjelenő panelen adjuk meg az osztály új nevét: **EmpDel**



Az EmpDel panel életre keltése

- Szereljük egy nyomógombot a Program panelre, legyen a neve: **Törlés**
- Adjunk hozzá eseménykezelőt (dupla katt)
- Másoljuk át a Lista panel megjelenítésének a kódját, és javítsuk ki:



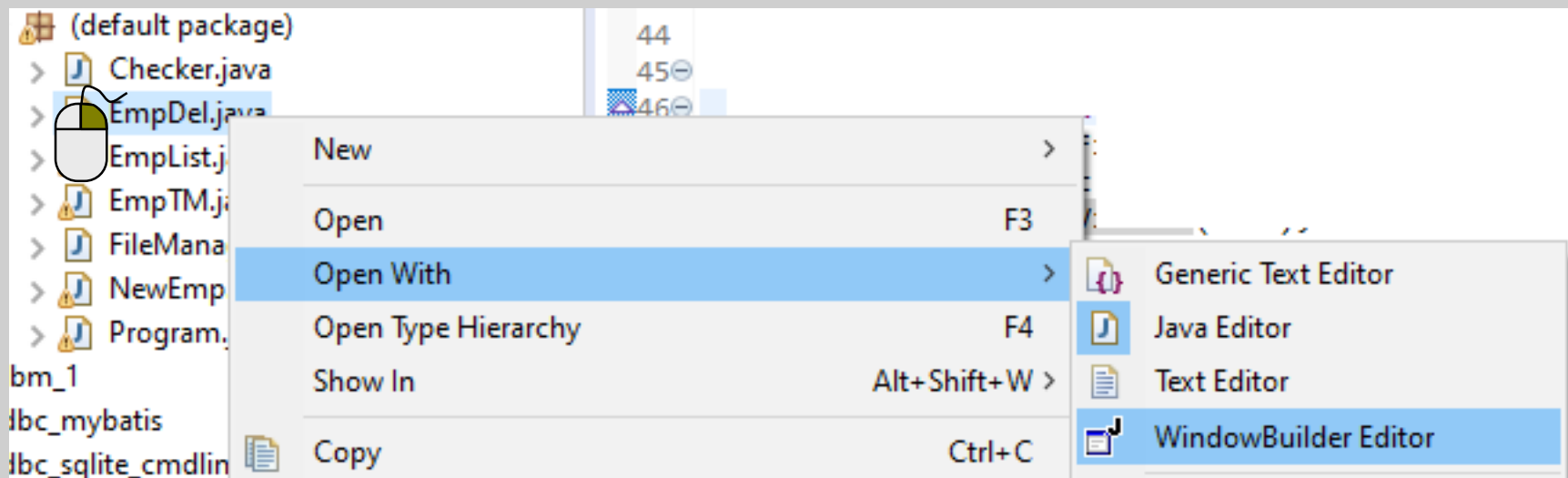
```
JBUTTON btnTrls = new JButton("T\u00F6rl\u00E9s");  
btnTrls.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent e) {  
        etm = FileManager.CsvReader();  
        EmpDel ed = new EmpDel(Program.this, etm);  
        ed.setVisible(true);  
    }  
});
```

- Indítsuk el a programot, próbáljuk ki.



Az EmpDel panel életre keltése

- Nyissuk meg az EmpDel osztályt a WindowBuilder-ben

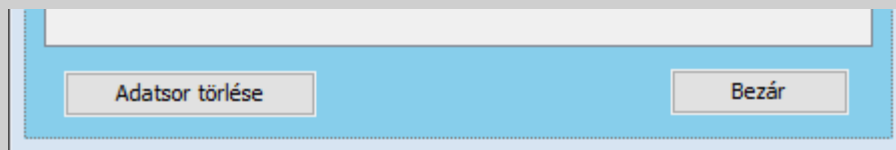


- Felül a Dolgozók listája szöveget javítsuk ki Dolgozók törlése szövegre

```
public EmpDel(JFrame f, EmpTM betm) {  
    super(f, "Dolgozók törlése", true);  
    etm = betm;  
}
```


Az EmpDel panel életre keltése

- Készítsünk egy Adatsor törlése feliratú gombot a panelre



- Találjuk ki a törlés logikáját, pl:
 - Ki kell jelölni 1 adatsort
 - Csak 1 adatsor törölhető egyszerre, ezt ellenőrizni kell
 - A törlés gombra kattintva:
 - Törlődjön az adatsor,
 - Töröljük fájlból az adatokat,
 - Ez csak úgy lehetséges, hogy felülírjuk a fájl teljes tartalmát, természetesen a törölt adatsor nélkül
 - Zárjuk be a Törlés panelt.

Az EmpDel panel életre keltése

- A panel életre keltéséhez szükség lesz egy **Checker** és egy **DbMethods** példányra, hozzuk ezeket létre:

```
public class EmpDel extends JDialog {  
  
    private final JPanel contentPanel = new JPanel();  
    private JTable table;  
    private EmpTM etm;  
    private Checker c = new Checker();  
    private DbMethods dbm = new DbMethods();  
}
```

Az EmpDel panel életre keltése

- Először készítsük el azt a kódot, ami ellenőrzi, hogy hány sor van a táblázatban kijelölve, és ha pontosan 1, akkor kitörli a táblázatból a sort:

```
JButton btnAdatsorTrlse = new JButton("Adatsor t\u00F6rl\u00E9se");
btnAdatsorTrlse.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        int db=0, jel=0, x=0;
        for(x = 0; x < etm.getRowCount(); x++)
            if ((Boolean)etm.getValueAt(x,0)) {db++; jel=x;}
        if (db==0) c.SM("Nincs kijelölve a törlendő rekord!",0);


        if (db>1) c.SM("Több rekord van kijelölve!\nEgyszerre csak egy"
            + " rekord törölhető!",0);

        if (db==1) {
            etm.removeRow(jel);
            c.SM("A rekord törölve!",1);
        }
    }
});
```

Új Insert metódus a FileManager-ben

- Ez a metódus nem hozzáfűz, hanem felülírja a fájl teljes tartalmát

```
public static void Insert(EmpTM etm) {  
    String x=";";  
    try {  
        PrintStream out = new PrintStream(new FileOutputStream("adatok.csv"));  
        out.println("Kód;Név;Születési_idő;Lakcím;Fizetés");  
        for(int i = 0; i < etm.getRowCount(); i++) {  
            String kod=etm.getValueAt(i,1).toString();  
            String nev=etm.getValueAt(i,2).toString();  
            String szid=etm.getValueAt(i,3).toString();  
            String lak=etm.getValueAt(i,4).toString();  
            String fiz=etm.getValueAt(i,5).toString();  
            out.println(kod+x+nev+x+szid+x+lak+x+fiz);  
        }  
        out.close();  
    } catch (IOException ioe) {  
        SM("FM.Insert: "+ioe.getMessage(), 0);  
    }  
}
```



Az EmpDel panel életre keltése

- Fejezzük be a törlés kódját:

```
JButton btnAdatsorTrlse = new JButton("Adatsor t\u00F6rl\u00E9se");
btnAdatsorTrlse.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        int db=0, jel=0, x=0;
        for(x = 0; x < etm.getRowCount(); x++)
            if ((Boolean)etm.getValueAt(x,0)) {db++; jel=x;}
        if (db==0) c.SM("Nincs kijelölve a törlendő rekord!",0);

        if (db>1) c.SM("Több rekord van kijelölve!\nEgyszerre csak egy"
            + " rekord törölhető!",0);

        if (db==1) {
            etm.removeRow(jel);
            FileManager.Insert(etm);
            dispose();
            c.SM("A rekord törölve!",1);
        }
    }
});
```

Program


■ Próbáljuk ki a törlés funkciót!

Dolgozók törlése

| Jel | Kód | Név | Szüldő | Lakóhely | Fizetés |
|-------------------------------------|-----|------------|------------|------------------|---------|
| <input type="checkbox"/> | 31 | Jég Elek | 1985.05.06 | Ózd | 99999 |
| <input type="checkbox"/> | 32 | Rossz Géza | 1981.11.09 | Miskolc | 180000 |
| <input type="checkbox"/> | 33 | Kecs Zsóé | 1993.02.13 | Eger | 411000 |
| <input checked="" type="checkbox"/> | 40 | Sokk Ernő | 1992.04.08 | Miskolc Kő u. 66 | 320000 |

Adatsor törlése

Program üzenet

 A rekord törölve!

OK

Dolgozók törlése

| Jel | Kód | Név | Szüldő | Lakóhely | Fizetés |
|--------------------------|-----|------------|------------|----------|---------|
| <input type="checkbox"/> | 31 | Jég Elek | 1985.05.06 | Ózd | 99999 |
| <input type="checkbox"/> | 32 | Rossz Géza | 1981.11.09 | Miskolc | 180000 |
| <input type="checkbox"/> | 33 | Kecs Zsóé | 1993.02.13 | Eger | 411000 |

Adatsor törlése

Bezár

Dolgozók listája

| Jel | Kód | Név | Szüldő | Lakóhely | Fizetés |
|--------------------------|-----|------------|------------|----------|---------|
| <input type="checkbox"/> | 31 | Jég Elek | 1985.05.06 | Ózd | 99999 |
| <input type="checkbox"/> | 32 | Rossz Géza | 1981.11.09 | Miskolc | 180000 |
| <input type="checkbox"/> | 33 | Kecs Zsóé | 1993.02.13 | Eger | 411000 |

Bezár

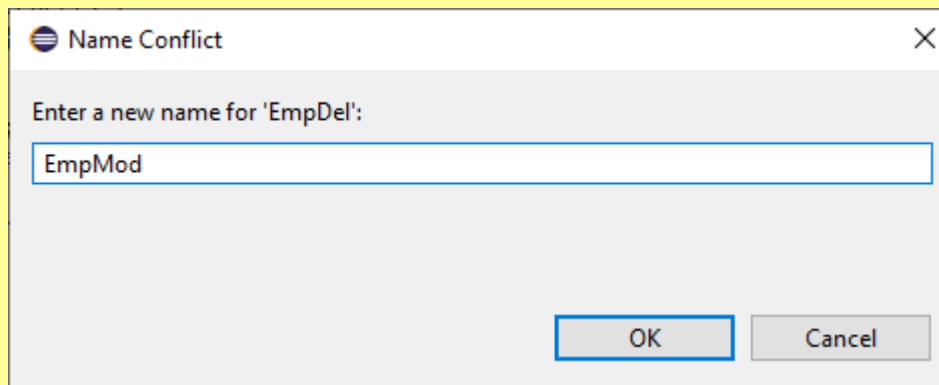
Ha elakadtál
küldj ímélt a
help@help.com
címmre!

Vagy tedd fel
a kezed!



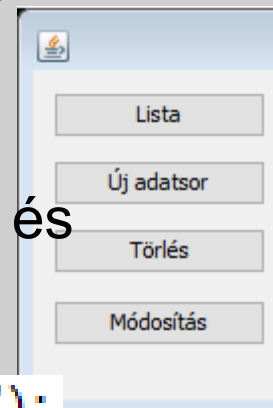
A módosítás

- A legegyszerűbb, ha a már módosított **Törlés** panelt használjuk alapanyagként. Ezért másoljuk le az EmpDel osztályt, legyen az új neve: **EmpMod**.
- Jelöljük ki a projektünkben az **EmpDel.java** osztályt (egyszerűen kattintsunk rá)
- Adjuk ki a **CTRL + C**, majd a **CTRL + V** parancsokat
- A megjelenő panelen adjuk meg az osztály új nevét: **EmpMod**



Az EmpMod panel életre keltése

- Szereljük egy nyomógombot a Program panelre, legyen a neve: **Módosítás**
- Adjunk hozzá eseménykezelőt (dupla katt)
- Másoljuk át a **Törlés** panel megjelenítésének a kódját, és javítsuk ki:



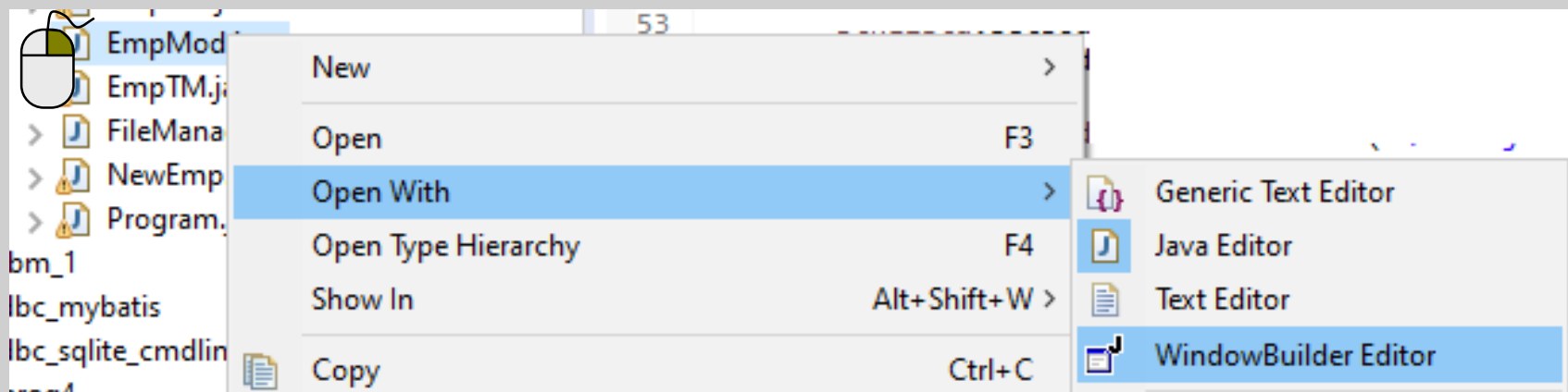
```
	JButton btnMdosts = new JButton("M\u00F3dos\u00EDt\u00E1s");  
	btnMdosts.addActionListener(new ActionListener() {  
		public void actionPerformed(ActionEvent e) {  
			etm = FileManager.CsvReader();  
			➡ EmpMod em = new EmpMod(Program.this, etm);  
			➡ em.setVisible(true);  
		}  
	});
```

- Indítsuk el a programot, próbáljuk ki.



Az EmpMod panel életre keltése

- Nyissuk meg az EmpMod osztályt a WindowBuilder-ben

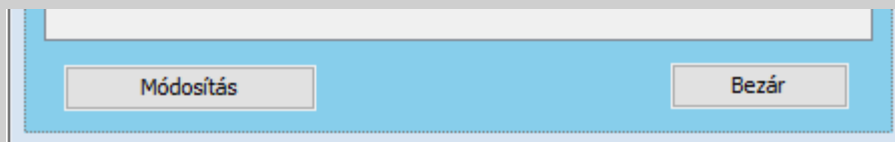


- Felül a Dolgozók törlése szöveget javítsuk ki Dolgozók módosítása szövegre

```
public EmpMod(JFrame f, EmpTM betm) {  
    super(f, "Dolgozók módosítása", true);  
    etm = betm;  
}
```

Az EmpMod panel életre keltése

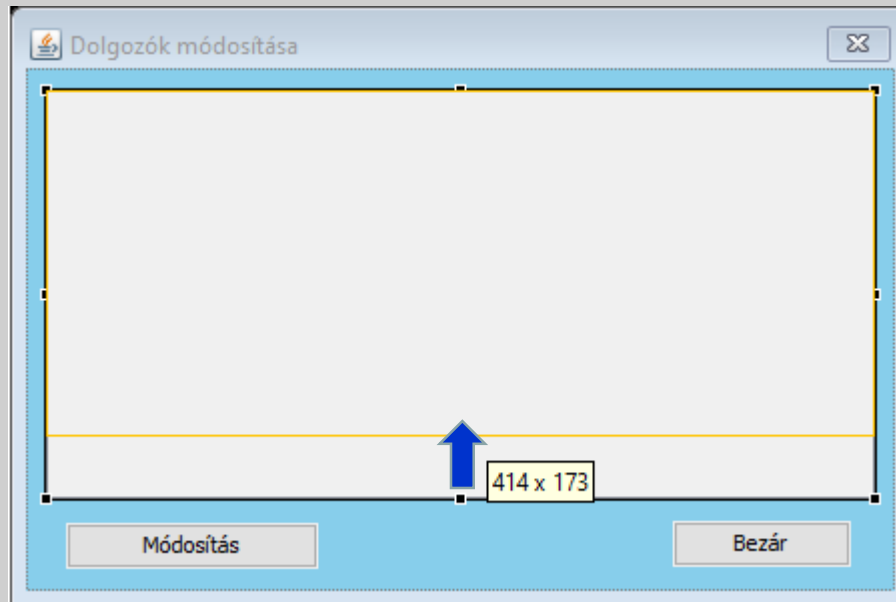
- Az Adatsor törlése feliratú gomb nevét írjuk át **Módosítás-ra**



- Találjuk ki a módosítás logikáját, pl:
 - Helyezzünk mezőket a táblázat alá, ide lehet beírni a módosítandó adatokat
 - Ki kell jelölni 1 adatsort
 - Csak 1 adatsor módosítható egyszerre, ezt ellenőrizni kell
 - A módosítás gombra kattintva:
 - Ellenőrizzük a mezők kitöltöttségét, ebből tudjuk megállapítani, hogy mely adatokat kell kijavítani
 - Javítsuk ki a táblázat kijelölt sorában az adatokat
 - Írjuk felül a fájl teljes tartalmát a táblázat tartalmával
 - Küldjünk üzenetet a módosítás megtörténtéről

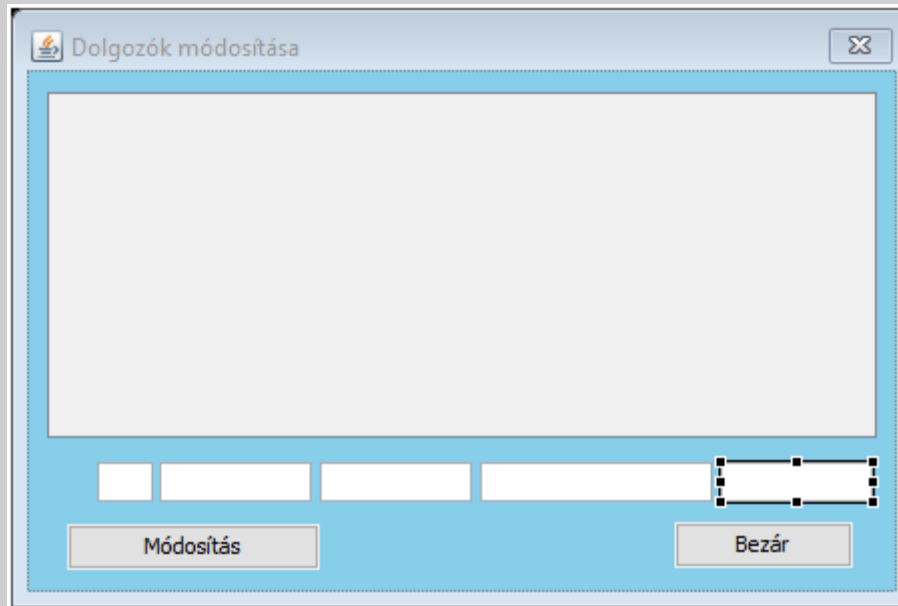
Az EmpMod panel életre keltése

- Vegyük kisebbre a táblázatot a panelen



Az EmpMod panel életre keltése

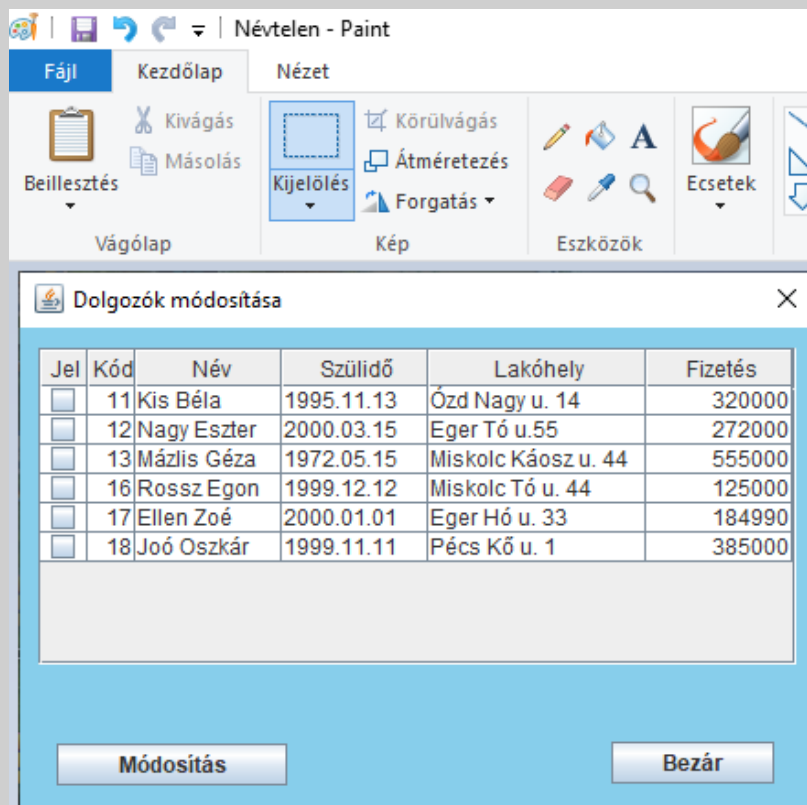
- Helyezzünk el szövegmezőket a panelen, legyen a változónevük rendre: **kod**, **nev**, **szid**, **lak**, **fiz**



- Rakjuk fel találomra a mezőket (mindjárt profin beállítjuk a helyzetüket), és adjuk meg a változónevüket!

Az EmpMod panel életre keltése

- Ötlet: mivel nem látszanak az oszlopok határai a panelen, aprócska trükkhöz folyamodunk:
 - Jelenítsük meg a panelt, és készítsünk róla egy képernyőképet:
ALT + PrintScreen
 - Nyissuk meg a **Paint** alkalmazást, és másoljuk bele a képet:
CTRL + V



Az EmpMod panel életre keltése

- A Paint-ben az egeret az egyes oszlopokra állítva olvassuk le az oszlopok X koordinátáit, és írjuk fel valahová.

 Dolgozók módosítása ✕

| Jel | Kód | Név | Szülidő | Lakóhely | Fizetés |
|--------------------------|-----|-------------|------------|---------------------|---------|
| <input type="checkbox"/> | 11 | Kis Béla | 1995.11.13 | Ózd Nagy u. 14 | 320000 |
| <input type="checkbox"/> | 12 | Nagy Eszter | 2000.03.15 | Eger Tó u.55 | 272000 |
| <input type="checkbox"/> | 13 | Mázlis Géza | 1972.05.15 | Miskolc Káosz u. 44 | 555000 |
| <input type="checkbox"/> | 16 | Rossz Egon | 1999.12.12 | Miskolc Tó u. 44 | 125000 |
| <input type="checkbox"/> | 17 | Ellen Zoé | 2000.01.01 | Eger Hó u. 33 | 184990 |
| <input type="checkbox"/> | 18 | Joó Oszkár | 1999.11.11 | Pécs Kő u. 1 | 385000 |

X= 38 64 142 224 342 422

Módosítás

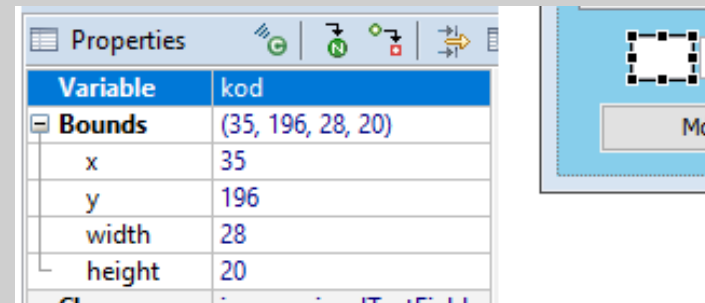
Bezár

Az EmpMod panel életre keltése

- Az egyes grafikus alkatrészeknek a méretmegadása a **Properties** területen, a **Bounds** paraméterekkel történik.

- Logika:

- Meg kell adni a bal felső sarkot (**x**, **y**)
- Az alkatrész szélességét (**width**)
- Az alkatrész magasságát (**height**)



- A szövegmezők között 2-3 pixelnyi helyet érdemes kihagyni!

- Pl: Az első szövegmező beállítása:

- **x** = 38
- **width** = $64 - 38 = 26$, ebből levonok 2 pixelt = 24

| | | | | | | |
|--------------------------|----|------------|------------|--------------|--------|-----|
| <input type="checkbox"/> | 18 | Joó Oszkár | 1999.11.11 | Pécs Kő u. 1 | 385000 | |
| X= | 38 | 64 | 142 | 224 | 342 | 422 |

- Ha nagyjából kész vagyunk, indítsuk el a programot, és ellenőrizzük a művünket!

Az EmpMod panel életre keltése

- A panelen ellenőriznünk kell, hogy pontosan 1 rekord legyen kijelölve (ezt a kódot örököltük a törlés paneltől), és ellenőriznünk kell azt is, hogy legalább 1 módosító adat be legyen írva az alsó szövegmezőkbe. Készítsük el a szükséges kódokat.
- Az EmpMod osztályba készítsünk egy olyan metódust, amely megszámolja, hogy hány alsó adatmező van kitöltve.
 - Ehhez szükséges egy olyan **filled** metódus a **Checker** osztályban, amely nem ír ki hibát, ha egy mező nincs kitöltve. Legelőször ezt készítsük el:

```
public boolean filled(JTextField a) {  
    String s = RTF(a);  
    if (s.length() > 0) return true;  
    else return false;  
}
```

Az EmpMod panel életre keltése

- Mivel az EmpTM táblamodellben a kód és a fizetés Integer típusú adat, szükség lesz egy olyan metódusra, mely a szövegmezőből kiolvasott szöveget Integer-ré alakítja. Tegyük ezt is a **Checker** osztályba, legyen a neve: **stringToInt**.

```
public int stringToInt (String s) {  
    int x = -1;  
    try {  
        x=Integer.valueOf(s);  
    } catch (NumberFormatException nfe) {  
        SM("stringToInt: "+nfe.getMessage(), 0);  
    }  
    return x;  
}
```

Az EmpMod panel életre keltése

- Most a Checker osztály filled metódusát felhasználva készítsük el az EmpMod osztályba a modDataPc metódust. Az EmpMod osztályban már példányosítottuk a Checker osztályt, a példány neve: c

```
public int modDataPc() {  
    int pc = 0;  
    if (c.filled(kod)) pc++;  
    if (c.filled(nev)) pc++;  
    if (c.filled(szip)) pc++;  
    if (c.filled(lak)) pc++;  
    if (c.filled(fiz)) pc++;  
    return pc;  
}
```

Az EmpMod panel életre keltése

- Írjuk meg azt a kódot, amely ha van módosító adat, végrehajtja a módosítást. A kód a Módosítás gomb eseménykezelőjébe kerül. Módosítsunk a meglévő kód szövegén is.

```
if (db==0) c.SM("Nincs kijelölve a módosítandó rekord!",0);  
if (db>1) c.SM("Több rekord van kijelölve!\nEgyszerre csak egy"  
              + " rekord módosítható!",0);  
if (db==1) {  
    if (modDataPc() > 0) {  
        boolean ok = true;  
        if (c.filled(kod)) ok = c.goodInt(kod, "Kód");  
        if (ok && c.filled(fiz)) ok = c.goodInt(fiz, "Fizetés");  
        if (ok) {  
            if (c.filled(kod)) etm.setValueAt(c.stringToInt(c.RTF(kod)), jel, 1);  
            if (c.filled(nev)) etm.setValueAt(c.RTF(nev), jel, 2);  
            if (c.filled(szip)) etm.setValueAt(c.RTF(szip), jel, 3);  
            if (c.filled(lak)) etm.setValueAt(c.RTF(lak), jel, 4);  
            if (c.filled(fiz)) etm.setValueAt(c.stringToInt(c.RTF(fiz)), jel, 5);  
            FileManager.Insert(etm);  
            c.SM("A rekord módosítva!",1);  
        }  
    } else {  
        c.SM("Nincs kitöltve egyetlen módosító adatmező sem!",1);  
    }  
}
```

Az EmpMod panel életre keltése

■ A módosítás teljes kódja:

```
int db=0, jel=0, x=0;
for(x = 0; x < etm.getRowCount(); x++)
if ((Boolean)etm.getValueAt(x,0)) {db++; jel=x;}
if (db==0) c.SM("Nincs kijelölve a módosítandó rekord!",0);



if (db>1) c.SM("Több rekord van kijelölve!\nEgyszerre csak egy"
               + " rekord módosítható!",0);
if (db==1) {
    if (modDataPc() > 0) {
        boolean ok = true;
        if (c.filled(kod)) ok = c.goodInt(kod, "Kód");
        if (ok && c.filled(fiz)) ok = c.goodInt(fiz, "Fizetés");
        if (ok) {
            if (c.filled(kod)) etm.setValueAt(c.stringToInt(c.RTF(kod)), jel, 1);
            if (c.filled(nev)) etm.setValueAt(c.RTF(nev), jel, 2);
            if (c.filled(szip)) etm.setValueAt(c.RTF(szip), jel, 3);
            if (c.filled(lak)) etm.setValueAt(c.RTF(lak), jel, 4);
            if (c.filled(fiz)) etm.setValueAt(c.stringToInt(c.RTF(fiz)), jel, 5);
        }
    }
    else {
        c.SM("Nincs kitöltve egyetlen módosító adatmező sem!",1);
    }
    //etm.removeRow(jel);
    //FileManager.Insert(etm);
    //dispose();
    //c.SM("A rekord módosítva!",1);
}
```



■ Míg próbálgatjuk a kódot, ezeket a sorokat tegyük megjegyzésbe!

Az EmpMod panel életre keltése

- Tesztelgessük a kódot. Ha megfelelően működik, tegyük bele a fájlba író utasítást a kódba. Ha úgy gondoljuk, hogy több adatot is módosíthatunk egymás után, akkor ne zárjuk be a módosítás után a panelt. A módosítás gomb kódja így nézzen ki:

```
if (db==1) {  
    if (modDataPc() > 0) {  
        boolean ok = true;  
        if (c.filled(kod)) ok = c.goodInt(kod, "Kód");  
        if (ok && c.filled(fiz)) ok = c.goodInt(fiz, "Fizetés");  
        if (ok) {  
            if (c.filled(kod)) etm.setValueAt(c.stringToInt(c.RTF(kod)), jel, 1);  
            if (c.filled(nev)) etm.setValueAt(c.RTF(nev), jel, 2);  
            if (c.filled(szip)) etm.setValueAt(c.RTF(szip), jel, 3);  
            if (c.filled(lak)) etm.setValueAt(c.RTF(lak), jel, 4);  
            if (c.filled(fiz)) etm.setValueAt(c.stringToInt(c.RTF(fiz)), jel, 5);  
             FileManager.Insert(etm);  
             c.SM("A rekord módosítva!",1);  
        }  
    } else {  
        c.SM("Nincs kitöltve egyetlen módosító adatmező sem!",1);  
    }  
}
```

Az EmpMod panel életre keltése

■ Teszt:

Dolgozók módosítása

| Jel | Kód | Név | Szülidő | Lakóhely | Fizetés |
|-------------------------------------|-----|-------------|------------|---------------------|---------|
| <input type="checkbox"/> | 11 | Kis Béla | 1995.11.13 | Ózd Nagy u. 14 | 320000 |
| <input type="checkbox"/> | 12 | Nagy Eszter | 2000.03.15 | Eger Tó u.55 | 272000 |
| <input type="checkbox"/> | 14 | Mázlis Géza | 1993.10.10 | Miskolc Káosz u. 44 | 555000 |
| <input checked="" type="checkbox"/> | 16 | Rossz Egon | 1999.12.12 | Miskolc Tó u. 44 | 125000 |
| <input type="checkbox"/> | 17 | Ellen Zoé | 2000.01.01 | Eger Hó u. 33 | 184990 |
| <input type="checkbox"/> | 18 | Joó Oszkár | 1999.11.11 | Pécs Kő u. 1 | 385000 |

33 Jó Benedek Pécs Tó u. 11

Módosítás 1 Bezár


Dolgozók módosítása

| Jel | Kód | Név | Szülidő | Lakóhely | Fizetés |
|-------------------------------------|-----|------------|------------|----------------|---------|
| <input type="checkbox"/> | 11 | Kis Béla | 1995.11.13 | Ózd Nagy u. 14 | 320000 |
| <input checked="" type="checkbox"/> | 33 | Jó Benedek | 1999.12.12 | Pécs Tó u. 11 | 125000 |
| <input type="checkbox"/> | 17 | Ellen Zoé | 2000.01.01 | Eger Hó u. 33 | 184990 |
| <input type="checkbox"/> | 18 | Joó Oszkár | 1999.11.11 | Pécs Kő u. 1 | 385000 |

33 Jó Benedek Pécs Tó u. 11

Módosítás Bezár

Program üzenet 2

 A rekord módosítva!

OK

Dolgozók listája 4

| Jel | Kód | Név | Szülidő | Lakóhely | Fizetés |
|-------------------------------------|-----|-------------|------------|---------------------|---------|
| <input type="checkbox"/> | 11 | Kis Béla | 1995.11.13 | Ózd Nagy u. 14 | 320000 |
| <input type="checkbox"/> | 12 | Nagy Eszter | 2000.03.15 | Eger Tó u.55 | 272000 |
| <input type="checkbox"/> | 14 | Mázlis Géza | 1993.10.10 | Miskolc Káosz u. 44 | 555000 |
| <input checked="" type="checkbox"/> | 33 | Jó Benedek | 1999.12.12 | Pécs Tó u. 11 | 125000 |
| <input type="checkbox"/> | 17 | Ellen Zoé | 2000.01.01 | Eger Hó u. 33 | 184990 |
| <input type="checkbox"/> | 18 | Joó Oszkár | 1999.11.11 | Pécs Kő u. 1 | 385000 |

Bezár

Az EmpMod panel – kis extra

- Extraként írhatunk egy **reset** metódust, mely kitörli a szövegmezőkből a módosítás adatait, és kiveszi a pipát a kijelölt rekordból. A módosítás tényének kiírása után hívjuk meg a metódust.

```
public void reset(int i) {  
    kod.setText("");  
    nev.setText("");  
    szid.setText("");  
    lak.setText("");  
    fiz.setText("");  
    etm.setValueAt(false, i, 0);  
}
```


Ha elakadtál küldj ímélt a
help@help.com címre!

HELP

Vagy tedd fel a táblád!



Az EmpMod panel – Ötlet

- Gyakorlásként írjuk meg a módosítást úgy, hogy a listában csak kijelöljük a módosítandó rekordot, a Módosítás gomb megnyomásakor egy új panelen jelenítjük meg az adatokat, és ott végezzük el a módosítás! (A képek illusztrációk!)

1 Dolgozók listája

| Jel | Kód | Név | Szülidő | Lakóhely | IQ |
|-------------------------------------|-----|------------|------------|----------|-----|
| <input type="checkbox"/> | 31 | Jég Elek | 1985.05.06 | Miskolc | 112 |
| <input type="checkbox"/> | 32 | Rossz Géza | 1981.11.09 | Miskolc | 105 |
| <input type="checkbox"/> | 33 | Keksz Zoé | 1993.02.13 | Eger | 125 |
| <input checked="" type="checkbox"/> | 40 | Kis Pista | 1999.12.31 | Ózd | 99 |
| <input type="checkbox"/> | 41 | Nagy Béla | 2000.01.01 | Eger | 133 |

2 Módosítás

3 Adatok módosítása

Kód: 40

Név: Kis Pista Kis István

Szülidő: 1999.12.31

Lakóhely: Ózd Balatonboglár

Iq: 99 101

Módosít

4 ABKezelő Üzenet

Update OK!

OK

Az EmpMod panel – Egyéni feladat

- A módosítás panel most úgy működik, hogy ha ki van töltve egy módosítás mező, akkor az alapján módosítja az eredeti adatot a program, de nem ellenőrzi, hogy az egyes adatok (kód, fietség, születési idő) megfelelőek-e (számadat van-e a mezőben, ill. a születési idő valóban dátum-e).
- Írjuk át úgy a kódot, hogy az ellenőrzések is megtörténjenek.



Question
van?

VÉGE

