

# Java Programozás

## 7. Gy: Adatkezelő alkalmazás 2.

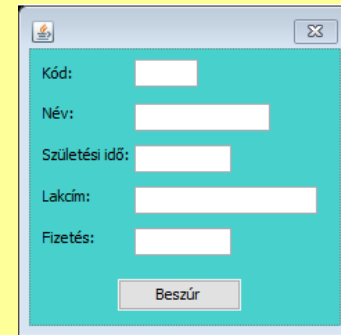
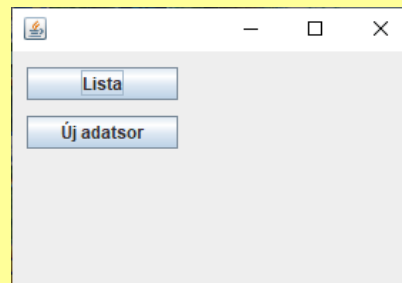
Swing, fájlkezelés



v: 2021.03.24

# Új adatsor beszúrása a fájlba

- Ahhoz, hogy adatsor tudjunk beszúrni a fájlba:
  - Szükséges egy panel, amin bekérjük az adatokat
  - Le kell ellenőriznünk az adatokat
    - Kivan töltve minden adatmező
    - A számadat mezőben valóban számok vannak
    - A beírt dátum valóban dátum
  - Be kell szúrunk a fájlban meglévő adatok után az új adatsort
  - A **Program**ot is módosítani kell, fel kell szerelni egy **Új adatsor** nyomógombot, amely elindítja a beszúrás műveletét

A screenshot of a small application window with a light blue background. It contains five text input fields labeled 'Kód:', 'Név:', 'Születési idő:', 'Lakcím:', and 'Fizetés:'. Below these fields is a button labeled 'Beszúr'.

# Új adatsor beszúrása a fájlba

---

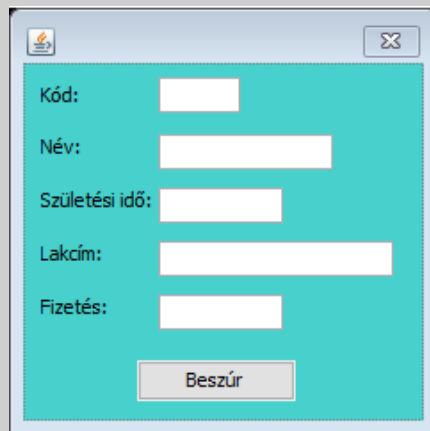
## 1. Panel létrehozása:

New\Swing\JDialog – Neve: NewEmp

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

## 2. Panelen az elrendezés menedzser kikapcsolása:

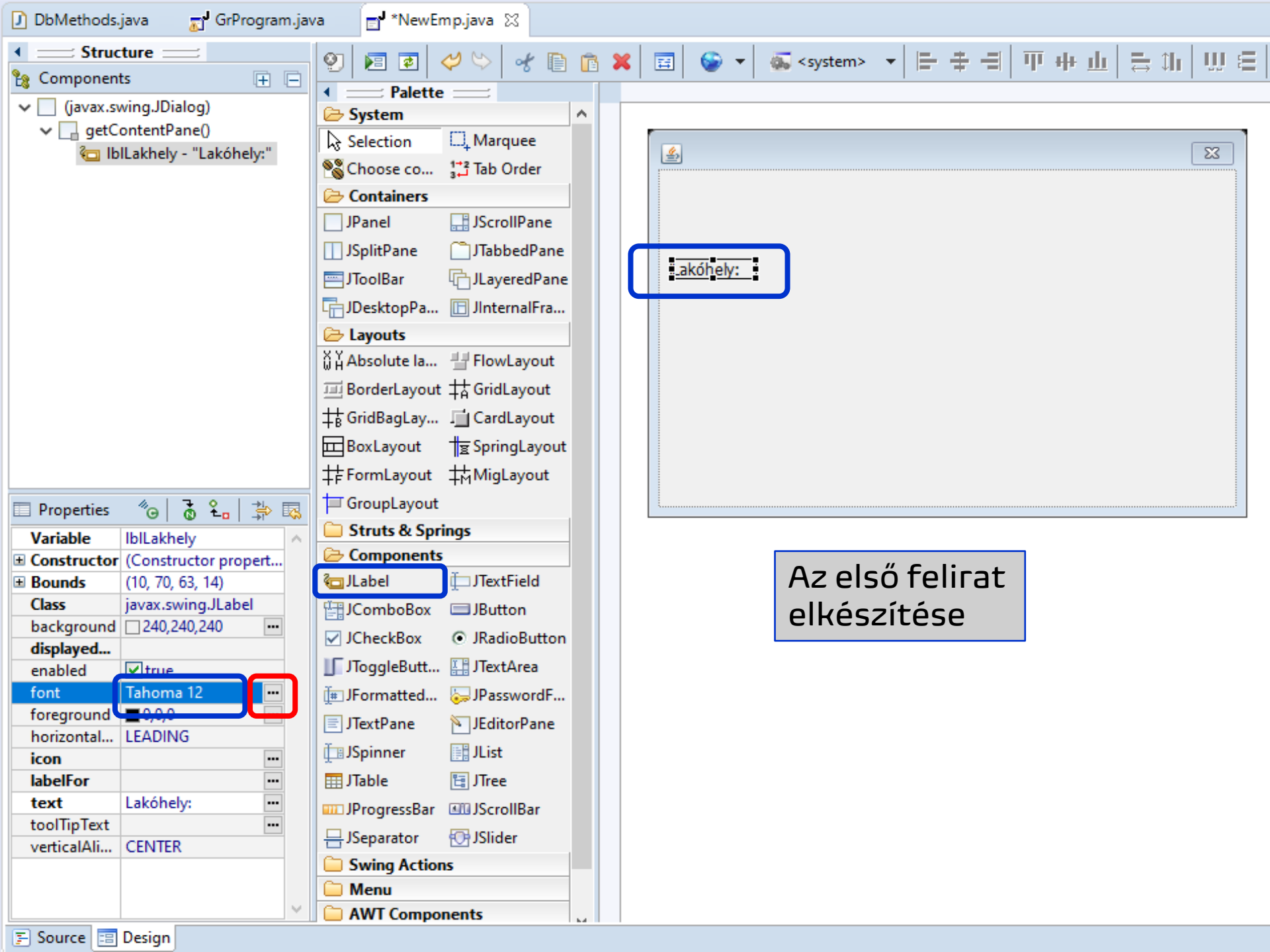
- Jobb klikk a panel közepére\Set layout\Absolute layout



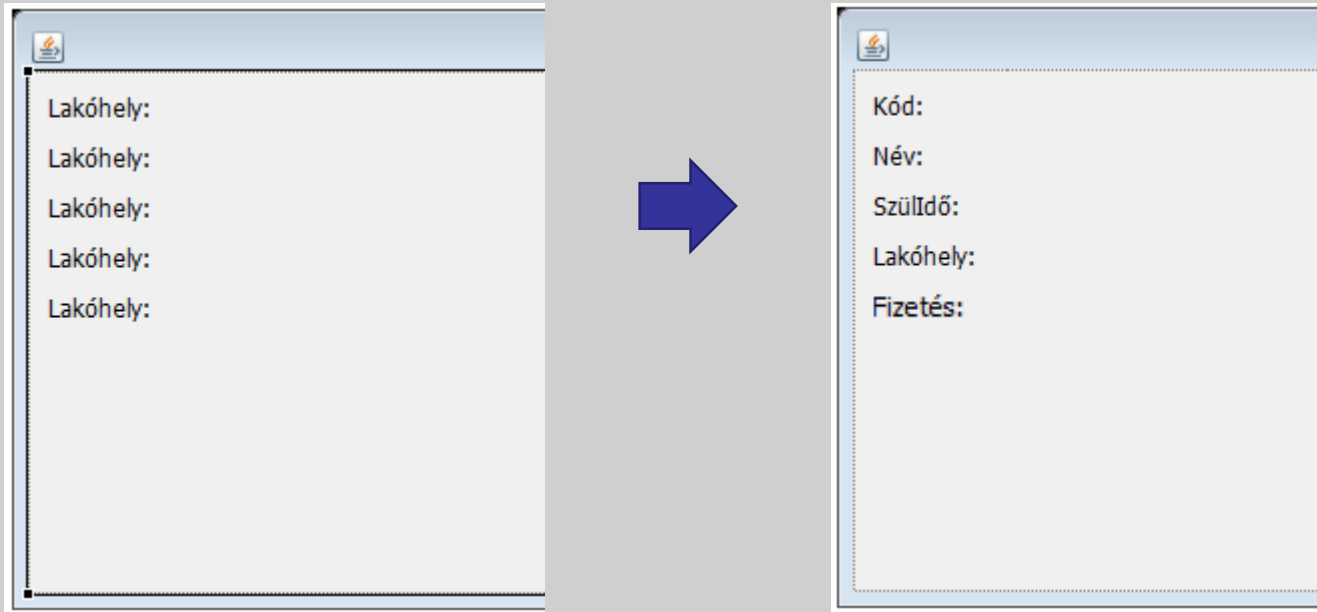
# Új adatsor beszúrása a fájlba

---

2. Rakjuk fel a panelre a feliratokat (JLabel) és szövegmezőket (JTextField)
  - Célszerű a leghosszabb feliratot felrakni, beállítani a betűméretét 12-esre, akkorára venni a felirat helyét, hogy bőven elférjen benne a szöveg, azután másolással létrehozni a többi feliratot.

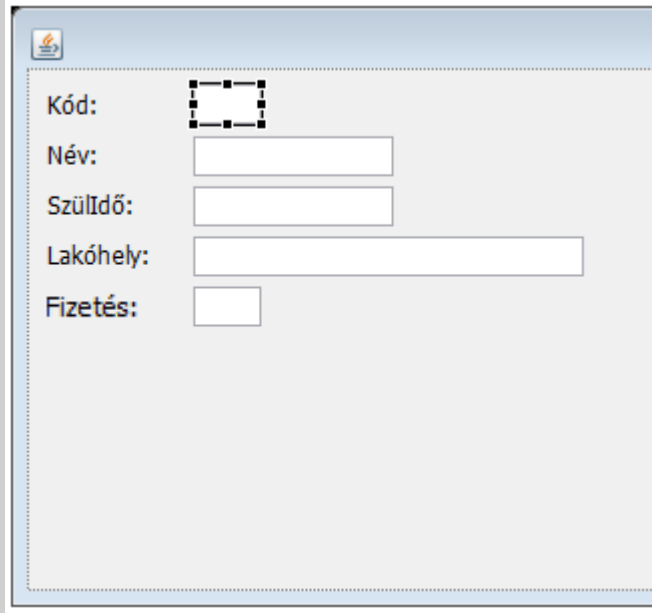


# Új adatsor beszúrása az adatbázisba



- A helyére toltam a legfelső feliratot, aztán kijelöltem, lemásoltam: **CTRL** + **C** majd beillesztettem **CTRL** + **V**, és helyére tettem a lemásolt feliratot.
- Ezt megcsináltam még háromszor.
- Aztán átírtam a szövegüket ([text](#)) a [Properties](#) panelen.
- A változónevük most lényegtelen.

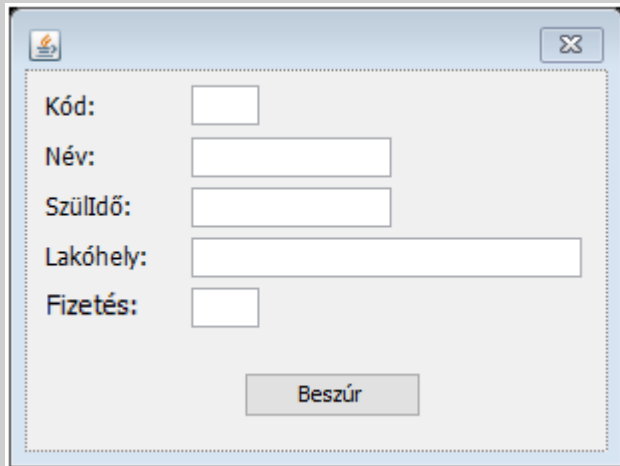
# Új adatsor beszúrása az adatbázisba



- Ugyanígy készültek a szövegmezők ([JTextField](#)) is.
- Másolás után szemre méreteztem őket, és a Properties panelen átírtam a változónevüket ([Variable](#)):
  - A változónevek rendre: [kod](#), [nev](#), [szid](#), [lak](#), [fiz](#)

# Új adatsor beszúrása az adatbázisba

- Ezután felraktam a **Beszúr** nyomógombot,
- A panel külső élére kattintva, a megjelenő méretezőpontok segítségével kisebbre húztam a panelt.
- Aztán a nyomógombhoz hozzáadtam egy eseménykezelőt (dupla kattintás a gombra).



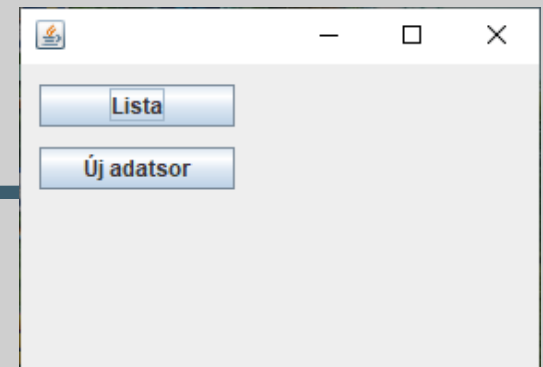
A screenshot of a small, light blue dialog box with a standard Windows-style title bar (minimize, maximize, close buttons). The dialog box contains five text input fields, each preceded by a label: 'Kód:', 'Név:', 'Születő:', 'Lakóhely:', and 'Fizetés:'. Below these fields is a single button labeled 'Beszúr'. The dialog box is outlined with a dashed border, indicating it is a floating window.



# A Program módosítása

- Szereljük egy Új adatsor nevű gombot a panelre, a változó neve legyen: `btnUjAdat`
- Adjunk hozzá `eseménykezelőt`
- Írjuk meg a kódját:

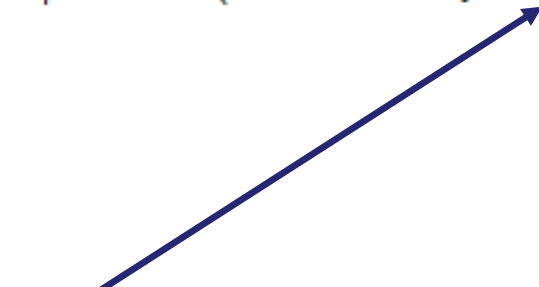
```
JButton btnUjAdat = new JButton("\u00DAj adatsor");  
btnUjAdat.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent e) {  
        NewEmp ne = new NewEmp();  
        ne.setVisible(true);  
    }  
});
```



# 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("FM.Insert: Adatok kiírva!", 1);  
    } catch (IOException ioe) {  
        SM("FM.Insert: "+ioe.getMessage(), 0);  
    }  
}
```



**True** érték megadásakor **hozzáfűzzük** az adatsort a fájl végéhez.

Az **SM** metódus kódja. Feladata: a megkapott szöveg megjelenítése grafikus üzenőpanelen.

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

# Üzenőpanel paraméterek

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

**JOptionPane.showMessageDialog(T, Ü, F, I)** - Üzenetablak megjelenítése.

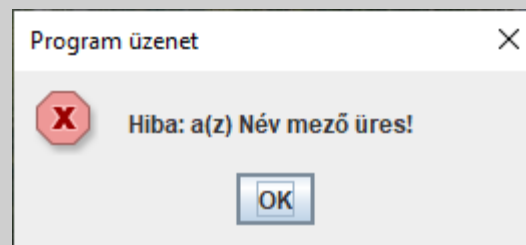
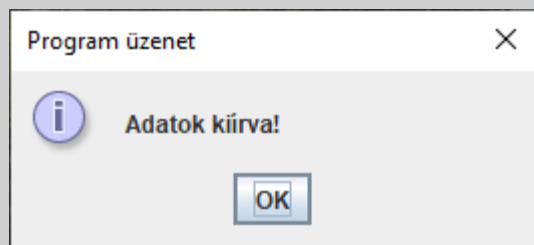
A megjelenő ablak modális, ez kb. azt jelenti, hogy amíg le nem **OK**ézzuk, nem folytathatjuk a főprogram használatát!

**T** - A tulajdonos (most **null**, vagyis nincs definiálva)

**Ü** - A panelen megjelenő üzenet szövege

**F** - A panel címe, felirata (most: "**Program üzenet**")

**I** - A panel ikonja (0-Stop, 1-Info, **2**-Figyelmeztetés, 3-Kérdés)



# 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();
}
}
```

Már működőképes a kód, de nincsenek a mezőkbe beírt adatok ellenőrizve.  
Készítsük el az ellenőrzéseket!

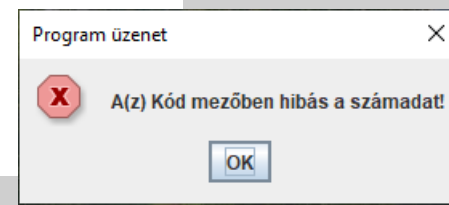
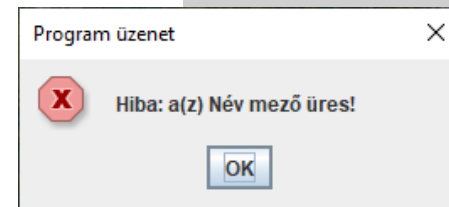
# A **NewEmp** osztály kódjának módosítása

---

- 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
  - Írjunk egy szövegmezőben lévő adat kiolvasását elvégző metódust (**RTF**)
  - Írjunk egy szövegmező kitöltöttségét ellenőrző metódust (**filled**)
  - Írjunk egy numerikus értéket ellenőrző metódust (**goodInt**), mely előbb meghívja a **filled** metódust is, így a kitöltöttséget is ellenőrzi
  - Szükség lesz egy üzeneteket megjelenítő (**SM**) metódusra is

# A Checker osztály kódja

```
public class Checker {  
  
    public boolean filled(JTextField a, String an) {  
        String s = RTF(a);  
        if (s.length() > 0) return true;  
        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);  
        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);  
}  
}
```

- 1 Egy szövegmező kitöltöttségét ellenőrző metódus. A mező nevét is átadjuk paraméterként, így az is kiírható hiba esetén.
- 2 Egész számmá alakítható-e egy szövegmezőben lévő adat. Ha nem, kiírja a hibás adatot tartalmazó mező nevét. A metódus ellenőrzi a mező kitöltöttségét is.
- 3 Egy szövegmező tartalmát kiolvasó metódus.
- 4 Egy üzenetet megjelenítő metódus.

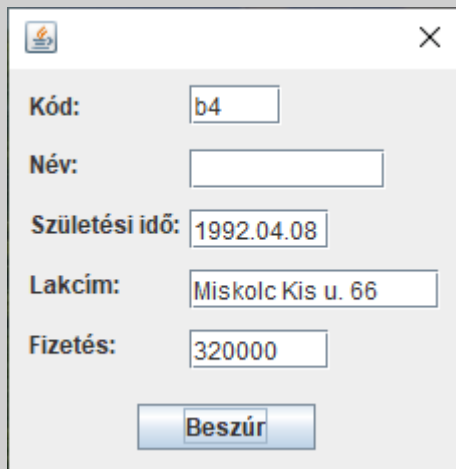


# A NewEmp osztály kódjának módosítása

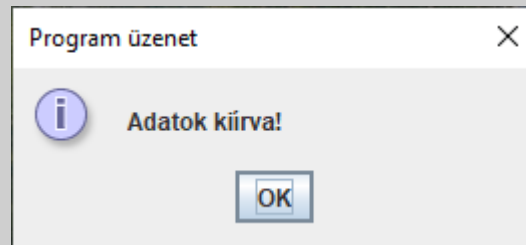
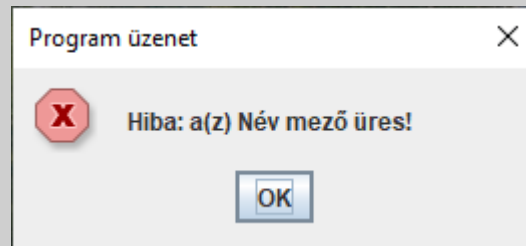
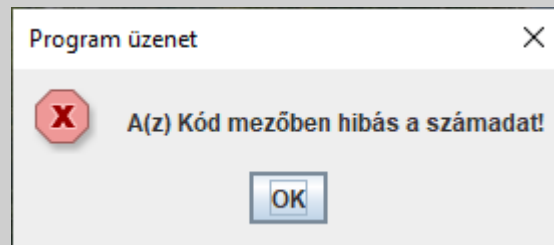
- A Beszúrás előtt ellenőrizni kell az adatokat:
  - Szövegnél csak a kitöltöttséget (filled)
  - Egész számnál a kitöltöttséget, és a konvertálhatóságot
  - Csak akkor történik beszúrás, ha minden adat rendben van

```
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'. It contains several input fields: 'Kód:' with the value 'b4', 'Név:' which is empty, 'Születési idő:' with the value '1992.04.08', 'Lakcím:' with the value 'Miskolc Kis u. 66', and 'Fizetés:' with the value '320000'. At the bottom is a 'Beszúr' button.



# A NewEmp – egy kis extra



- Nem ártana leellenőrizni azt, hogy a dátum mezőbe beírt szöveg valóban dátum-e! Oldjuk meg!
- Az ellenőrzés logikája:
  - Készítünk egy szöveges dátum formátumot, melyben kódokkal megadjuk, hogy hogyan kell kinéznie a dátumnak:
    - yyyy.MM.dd – év 4 szájjeggyel, aztán pont, hónap 2 számjeggyel, aztán pont, és a nap 2 számjeggyel

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

- Az ellenőrzés logikája:
  - Megpróbáljuk a megadott formátum szerinti dátummá alakítani a kapott szöveget:

```
Date date = RDF.parse(SDate);
```

# A NewEmp – egy kis extra



- Írjunk metódust a **Checker** osztályba a **dátum** ellenőrzésére:

```
public boolean DateFormatChecker(String SDate) {  
    try {  
        Date date = RDF.parse(SDate);  
        if (!RDF.format(date).equals(SDate)){return false;}  
        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;  
    }  
}
```

Dátum ellenőrzés

A kapott szöveg-  
mező kitöltöttségét,  
és a benne lévő  
adat dátummá  
alakíthatóságát  
ellenőrző metódus

A **parse** metódus túl intelligens, képes a 1992.13.10 dátumot 1993.01.10 dátummá alakítani!, ezért kell egy ellenőrzés a metódusba (az eredeti dátum szöveges alakja megegyezik-e a képzett dátum szöveges alakjával)

# A NewEmp – egy kis extra



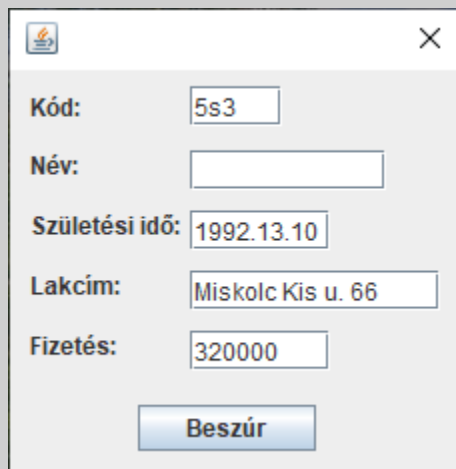
- A `DateFormatChecker` metódus működéséhez szükséges, hogy a `Checker` osztályban létrehozzuk az `RDF` példányt:

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

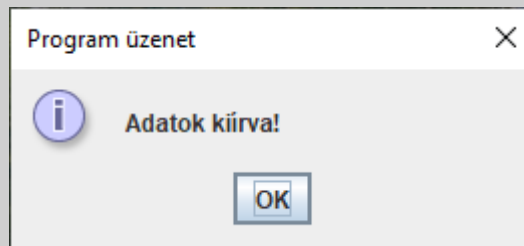
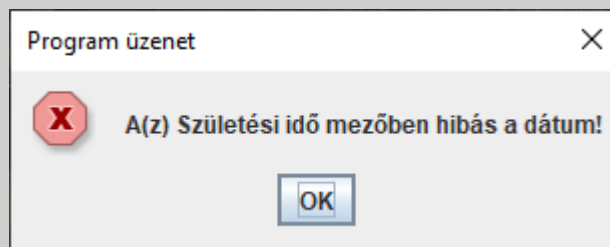
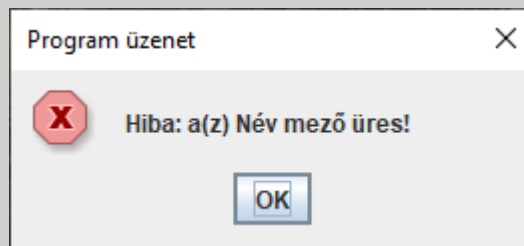
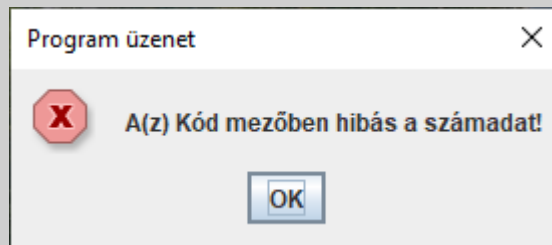
- A beszúrás gomb kódja a `NewEmp` osztályban a dátum ellenőrzésével:

```
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(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-style form titled 'NewEmp'. It contains five input fields: 'Kód:' with '5s3', 'Név:' (empty), 'Születési idő:' with '1992.13.10', 'Lakcím:' with 'Miskolc Kis u. 66', and 'Fizetés:' with '320000'. A 'Beszúr' button is at the bottom.



Ha elakadtál  
küldj ímélt!

OnLine

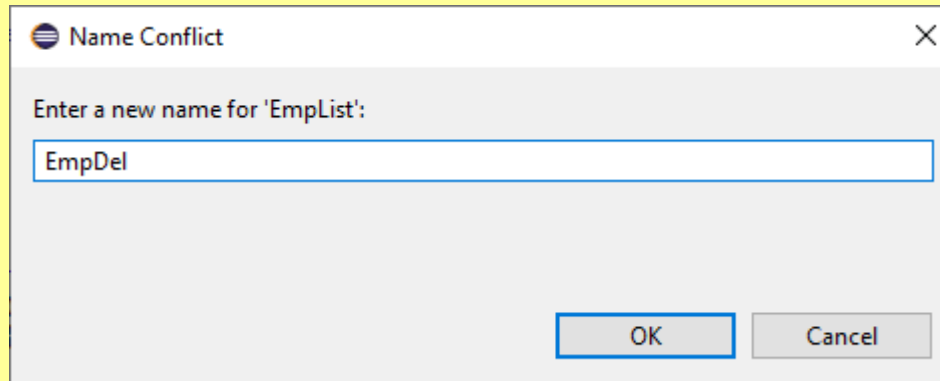
NonLine



Vagy tedd  
fel a  
táblád!

# A törlés

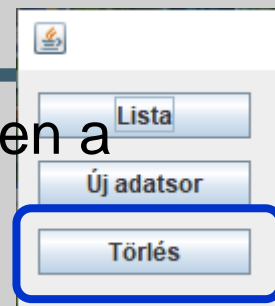
- A legegyszerűbb megoldás rászerezni egy **Törlés** gombot a **Lista** panelre. Ezért másoljuk le az **EmpList** panelt, legyen az új neve: **EmpDel**.
- Jelöljük ki a projektünkben az **EmpList.java** osztályt (egyszerűen kattintsunk rá)
- Adjuk ki a **CTRL + C**, majd a **CTRL + V** parancsokat
-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



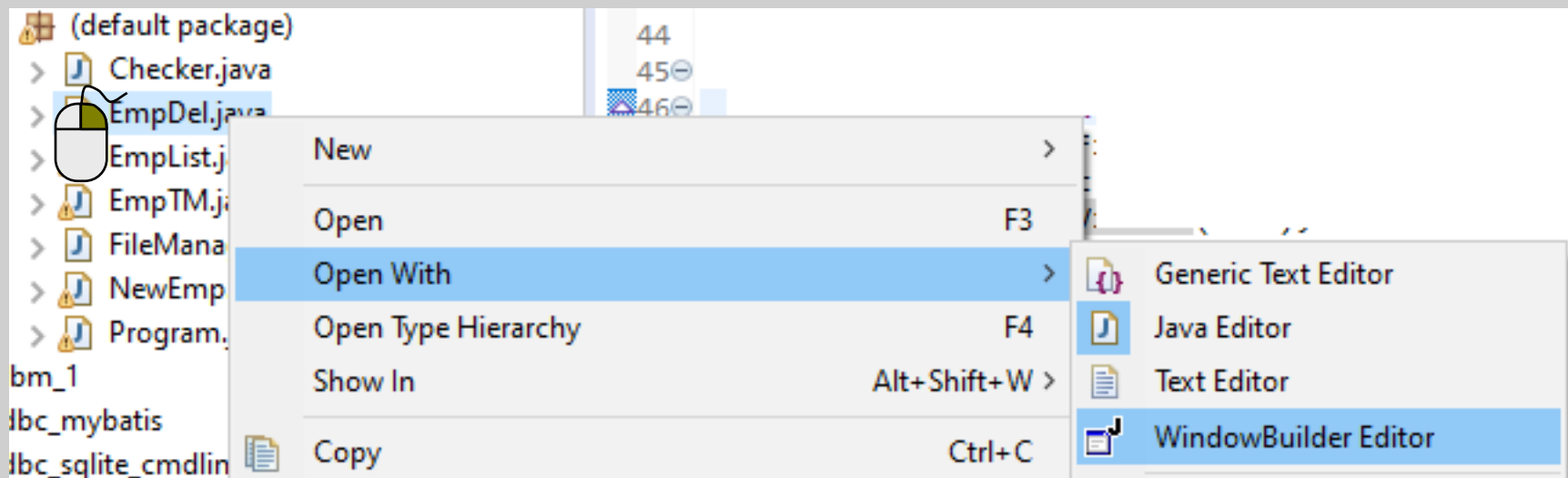
```
 JButton btnTrls = new JButton("T\u00F6rl\u00E9s");  
 btnTrls.addActionListener(new ActionListener() {  
     public void actionPerformed(ActionEvent e) {  
         etm = FileManager.CsvReader();  
         EmpDel ed = new EmpDel(Program.this, etm);  
         ed.setVisible(true);  
     }  
 });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

- Indítsuk el a programot, próbáljuk ki.



# Az EmpDel panel életre keltése

- Nyissuk meg az EmpDel osztályt a WindowBuilder-ben

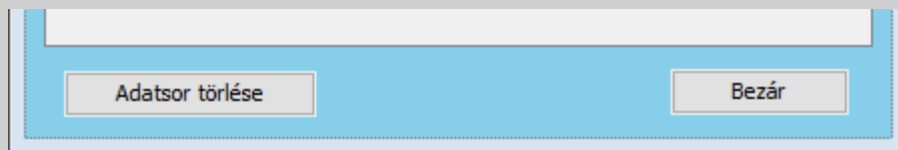


- Felül a Dolgozók listája szöveget javítsuk ki Dolgozók törlése szövegre

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

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

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

# Magyarázatok

- 1 Egy ciklussal végigmegyünk a táblamodell sorain
- 2 Ha az érintett sorban be van kapcsolva az első oszlop **jelölője**, akkor növeljük a kijelölt sorok számát 1-el, és megjegyezzük a kijelölt sor számát
- 3 Ha a kijelölt sorok száma nulla, kiírjuk, hogy nincs kijelölt adatsor
- 4 Ha a kijelölt sorok száma egynél több, kiírjuk, hogy több kijelölt adatsor van
- 5 Ha a kijelölt sorok száma pontosan egy, akkor:
  - 6
    - kitöröljük a táblamodellből a kijelölt rekordot
    - kiírjuk, hogy töröltük a rekordot

<input type="checkbox"/>	33	Keksz Zoé
<input checked="" type="checkbox"/>	53	Kő Sára
<input type="checkbox"/>	55	BitMan

# Új Insert metódus a FileManager-ben

- Ez a metódus nem hozzáfűz, hanem felülírja a fájl teljes tartalmát a paraméterként kapott táblamodell adatai alapján

```
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);  
    }  
}
```

Igazi OOP megoldás, **metódus túlterhelés**. A metódus neve megegyezik, de a bemenő paramétere más, mint a meglévő másik **Insert** metódusnak.

# 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

Listázás módosítás után

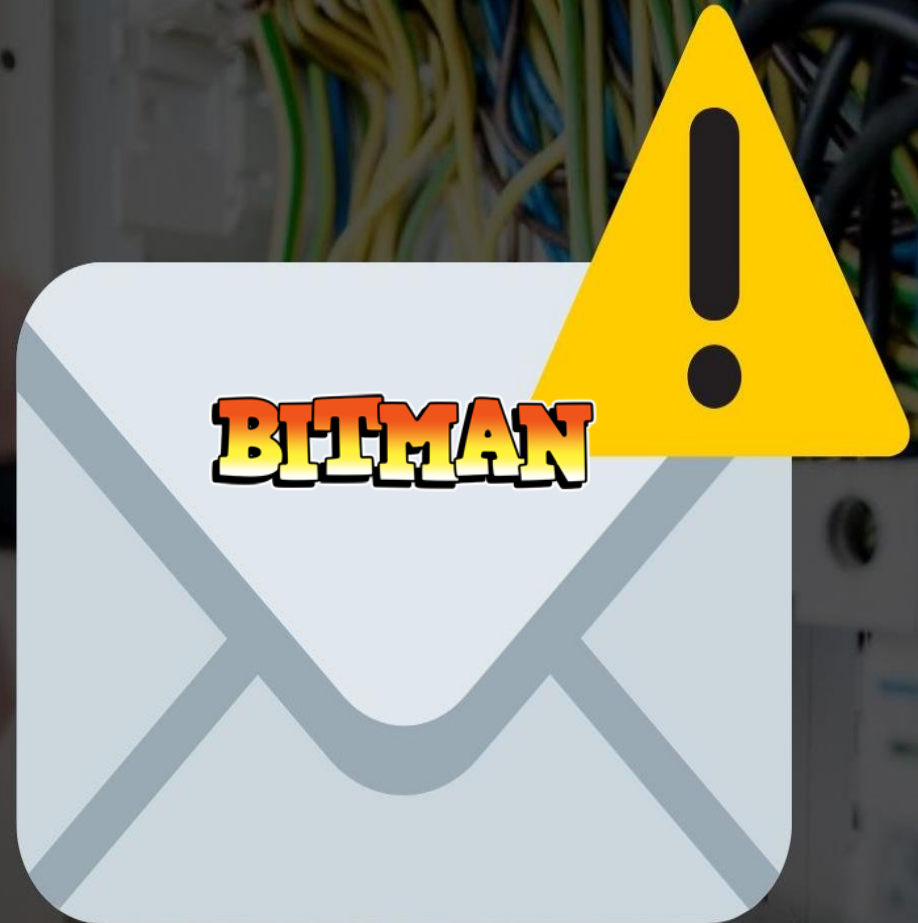
Bezár

Ha elakadtál  
küldj ímélt a  
címemre!

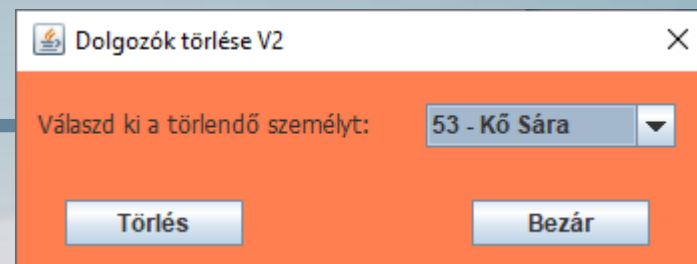
OnLine

NonLine

Vagy tedd fel  
a kezed!



# Törlés funkció v2 elkészítése



## ■ Komplikáltabb megoldás:

- Készítsünk egy olyan panelt, amin egy legördíthető listából lehet kiválasztani a törlendő adatsort.
- A listában a személyek kódja, és neve jelenjen meg.
- A Törlés gomb megnyomásakor a listában kiválasztott személy törlődjön az adatbázisból.

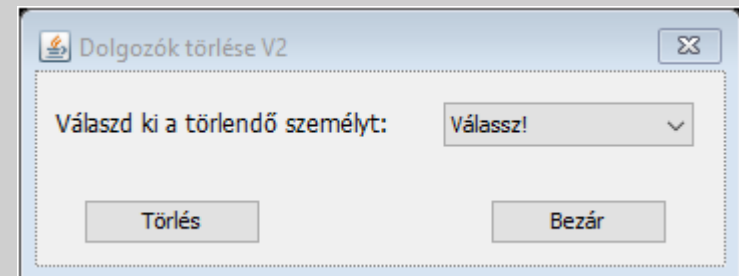
V2



Bónuszfeladat azoknak, akik több órai feladatot szeretnének!  
Átlagos ambíciókkal rendelkező hallgatók lapozzák át ezt a pár oldalas leírást!

# Törlés v2

- Készítsünk egy **DelEmpV2** nevű JDialog panelt
  - A *Generate JDialog with OK and Cancel buttons* opciót kapcsoljuk ki!
  - Váltunk **Design** nézetre
  - Panelen az elrendezés menedzsert kapcsoljuk ki:
    - Jobb klikk a panel közepére\Set layout\Absolute layout
  - Készítsük el a **Bezár** (**btnBezar**) és a **Törlés** (**btnTrls**) gombot, aztán adjunk hozzájuk eseménykezelőt
  - Váltunk vissza **Design** nézetre
  - Szereljük fel a feliratot (**JLabel**)
  - Szereljük a panelre egy **JComboBox**-ot (legördülő listát)
  - Méretezzük a panelt a képen látható módon



# Törlés V2

- Töröljük ki a main metódust a `DelEmpV2` panel kódjából.
- Készítsük el a `dbm` példányt
- Készítsük el az `etm` példányt
- Alakítsuk át a konstruktort
- A Bezár gombhoz adjuk meg a `dispose()` utasítást

```
public class DelEmpV2 extends JDialog {  
    private final JPanel contentPanel = new JPanel();  
    DbMethods dbm = new DbMethods();  
    private EmpTM etm;  
  
    public DelEmpV2(JFrame f, EmpTM betm) {  
        super(f, "Dolgozók törlése V2", true);  
        etm = betm;
```

```
        JButton btnBezr = new JButton("Bez\u00E1r");  
        btnBezr.addActionListener(new ActionListener() {  
            public void actionPerformed(ActionEvent e) {  
                dispose();  
            }  
        });  
    }  
}
```

# Törlés V2

- Másoljuk át az **RTM** metódust a konstruktor alá a **DelEmp** osztályból:

```
public String RTM(int row, int col) {  
    return etm.getValueAt(row, col).toString();  
}
```

- Töltsük fel a legördülő listát adatokkal.
  - Az első adat legyen a **Válasszon!** szöveg, ez fog megjelenni a panel indulásakor.
  - A többi adatot az adatbázisból töltjük be. Mikor példányosítjuk a panelt, bemenő paraméterként megkapja a **ReadAllData** metódus által (a **Programban**) feltöltött **etm** táblamodell példányt. Ebből rakjuk össze az adatpárokat, **Kód – Név** formában.

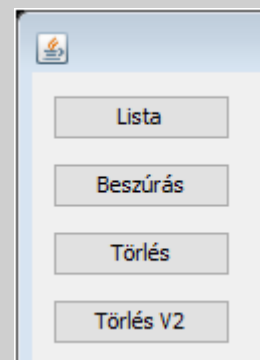
# Törlés V2

- A legördülő lista feltöltésének kódja:

```
JComboBox comboBox = new JComboBox();
comboBox.addItem("Válassz!");
for(int i = 0; i < etm.getRowCount(); i++) {
    comboBox.addItem(RTM(i,1)+" - "+RTM(i,2));
}
```

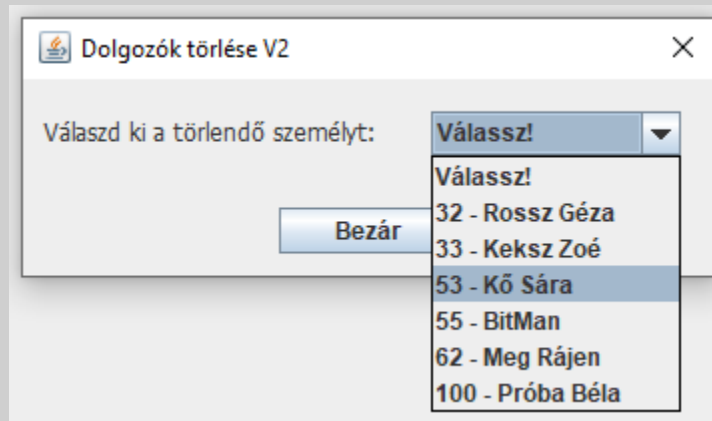
- A **Program**-ban szereljük fel egy **Törlés V2** gombot, adjunk hozzá akciókezelőt, és írjuk meg a hozzá tartozó kódot (a panel elindítását):

```
JButton btnTrlsV2 = new JButton("T\u00F6rl\u00E9s V2");
btnTrlsV2.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        etm = dbm.ReadAllData();
        DelEmpV2 de2 = new DelEmpV2(GrProgram.this, etm);
        de2.setVisible(true);
    }
});
```



# Törlés V2

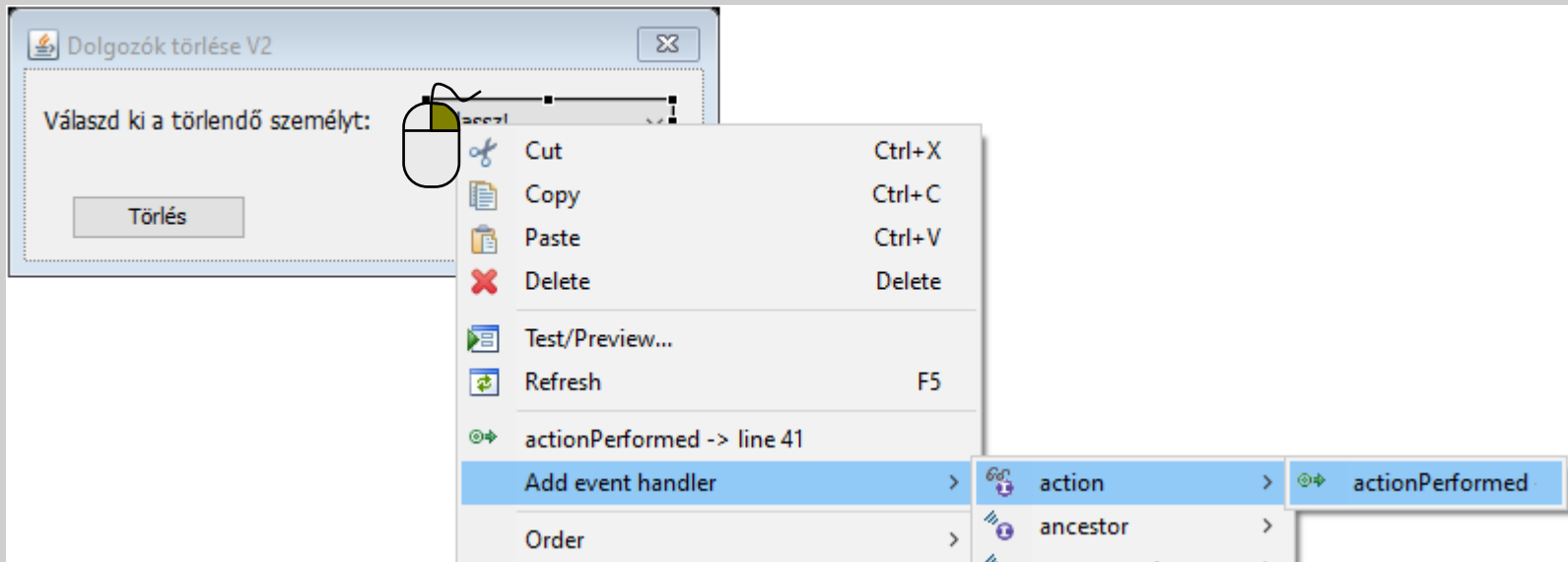
- Indítsuk el a programot, és ellenőrizzük, hogy működik-e a lista a Törlés V2 panelen!
- Nálam: OK!





# Törlés V2


- Bűvöljük tovább a **DelEmpV2** panelt
- Jelenítsük meg a grafikus felületét
- Adjunk akciókezelőt a JComboBox-hoz:
  - Sajnos itt a dupla kattintás nem eredményes, ezért:
  - Jobb klikk a JComboBox-ra\Add event handler\action\actionPerformed menü



# Törlés V2


- Írjuk meg a legördülő lista eseménykezelőjének a kódját.
- Ehhez először deklaráljunk egy Stringet, amibe a lista kiválasztott elemének értéke kerül. Alapértéke a lista kiinduló értéke, a Válassz! szöveg!

```
public class DelEmpV2 extends JDialog {  
    private final JPanel contentPanel = new JPanel();  
    DbMethods dbm = new DbMethods();  
    private EmpTM etm;  
    private String selected="Válassz!";
```



- Jöjjön a legördülő lista eseménykezelőjének a kódja:

```
comboBox.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent arg0) {  
        selected = comboBox.getSelectedItem().toString();  
    }  
});
```



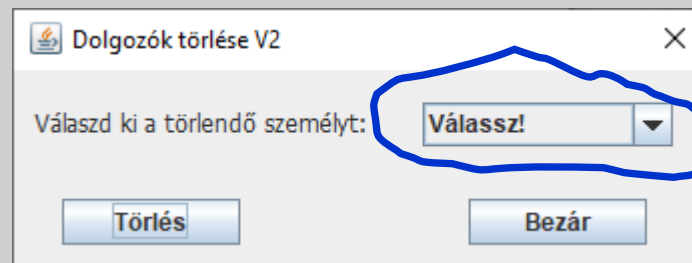
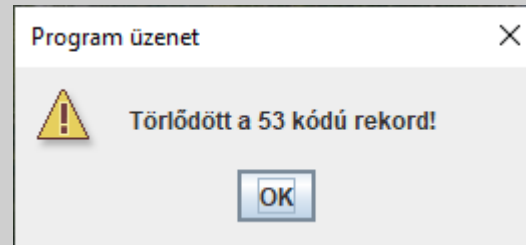
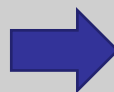
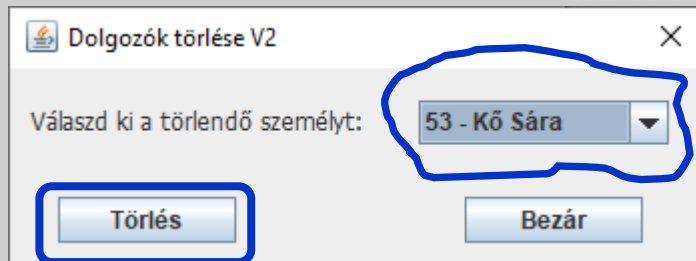
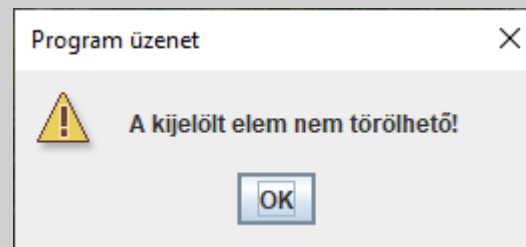
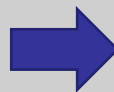
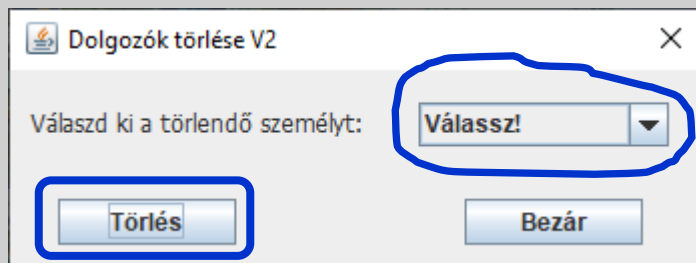
# Törlés V2

- A tényleges törlés akkor történik meg, ha megnyomjuk a **Törlés** gombot. Ekkor:
  - Ha a "Válassz!" listaelem van kiválasztva, küldünk egy hibaüzenetet,
  - Ha más listaelem van kiválasztva:
    1. Kilogikázzuk a kódot a **kód – név** szövegpárból,
    2. Kitöröljük a kóddal azonosított rekordot az adatbázisból,
    3. Kitöröljük az aktuális elemet a listából,
    4. Beállítjuk aktuálisnak a listában a "Válassz!" elemet.

```
JButton btnTrls = new JButton("T\u00F6rl\u00E9s");  
btnTrls.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent e) {  
        if (selected.contentEquals("Válassz!")) dbm.SM("A kijelölt elem nem törölhető!");  
        else {  
            1      int index = selected.indexOf(" - ");  
            2      String kod = selected.substring(0,index);  
            3      dbm.DeleteData(kod);  
            4      comboBox.removeItem(selected);  
                comboBox.setSelectedItem("Válassz!");  
        }  
    }  
});
```

# Törlés V2

## ■ Próba futtatás, ellenőrzés:



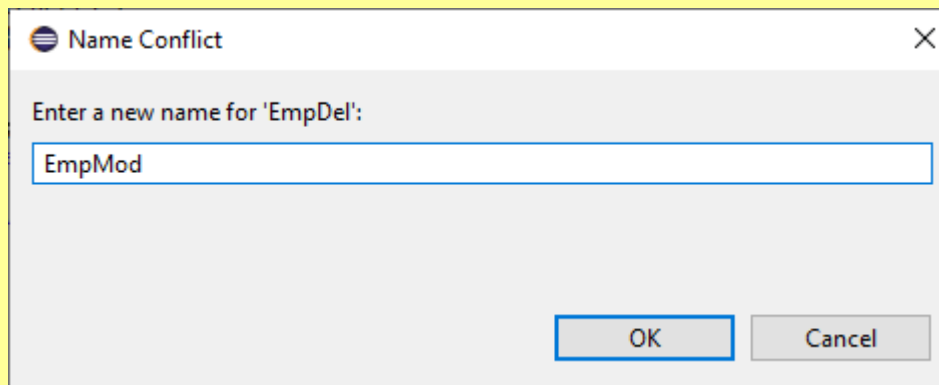
**Hurrá!** Minden rendben!

MEO:

**PERFECT**

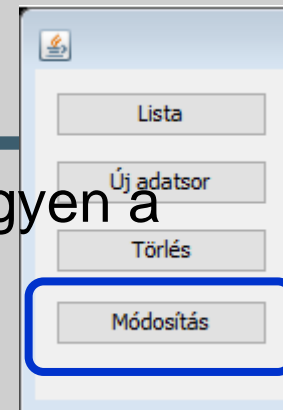
# A módosítás

- 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 példányosításának a kódját, és javítsuk ki:



```

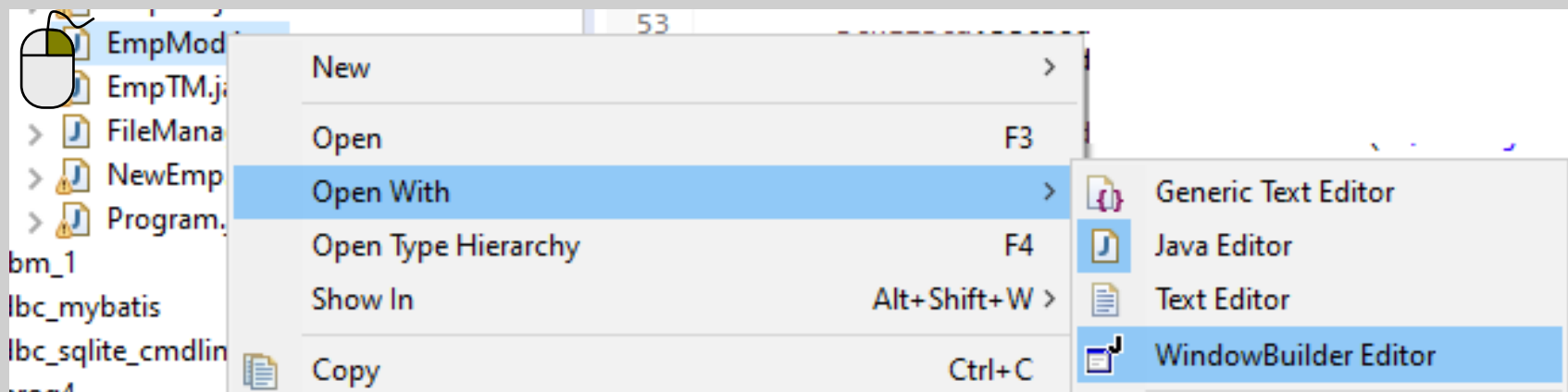
JButton btnMdosts = new JButton("M\u00F3dos\u00EDt\u00E1s");
btnMdosts.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        etm = FileManager.CsvReader();
        EmpMod em = new EmpMod(Program.this, etm);
        em.setVisible(true);
    }
});
```

- Indítsuk el a programot, próbáljuk ki.



# Az EmpMod panel életre keltése

- Nyissuk meg az EmpMod osztályt a WindowBuilder-ben

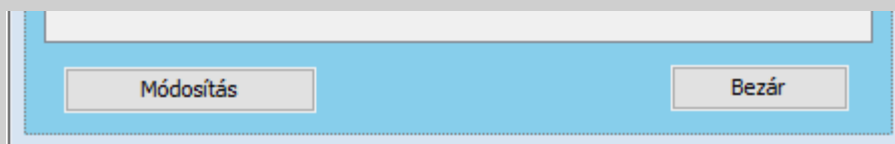


- Felül a konstruktor elején a Dolgozók törlése szöveget javítsuk ki Dolgozók módosítása szövegre

```
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

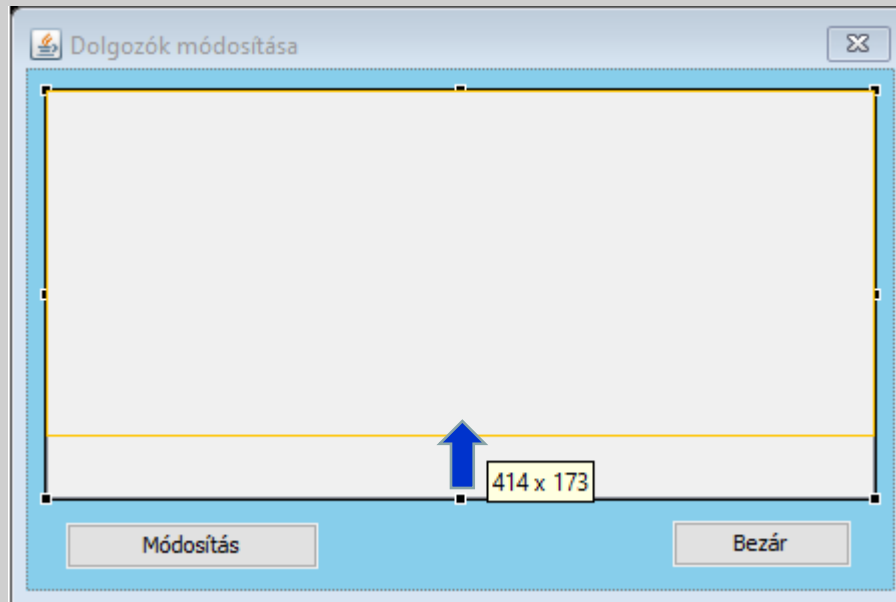


- Találjuk ki a módosítás logikáját, pl:
  - Helyezzünk mezőket a táblázat alá, ide lehet beírni a módosítandó adatokat
  - Ki kell jelölni 1 adatsort a táblázatban
    - Csak 1 adatsor módosítható egyszerre, ezt ellenőrizni kell!
  - A módosítás gombra kattintva:
    - Ellenőrizzük a mezők kitöltöttségét,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



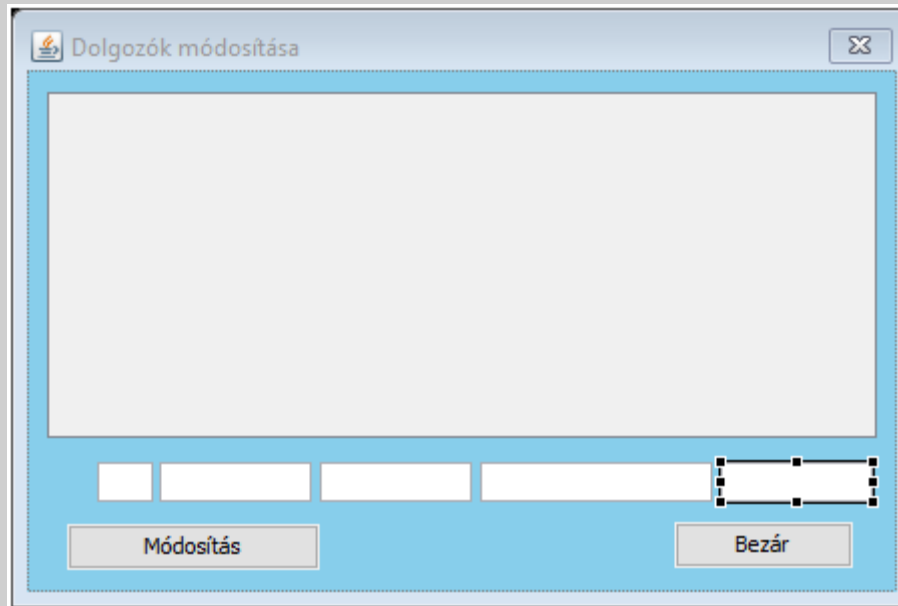
# Az EmpMod panel életre keltése

- Vegyük kisebbre a táblázatot a panelen



# Az EmpMod panel életre keltése

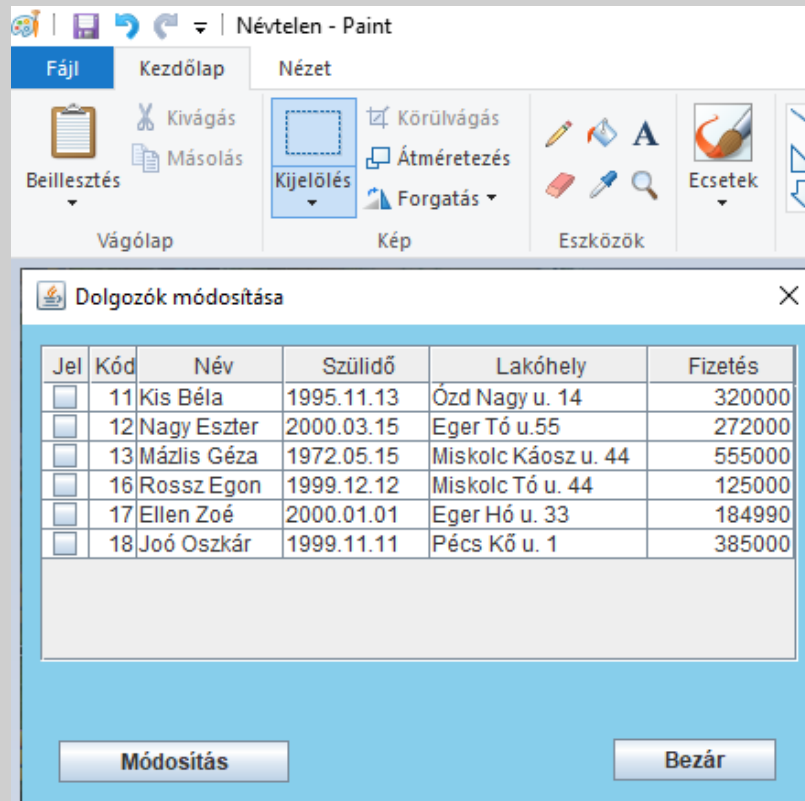
- Helyezzünk el szövegmezőket a panelen, legyen a változónevük rendre: **kod**, **nev**, **szid**, **lak**, **fiz**



- 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= 35 65 143 221 342 422

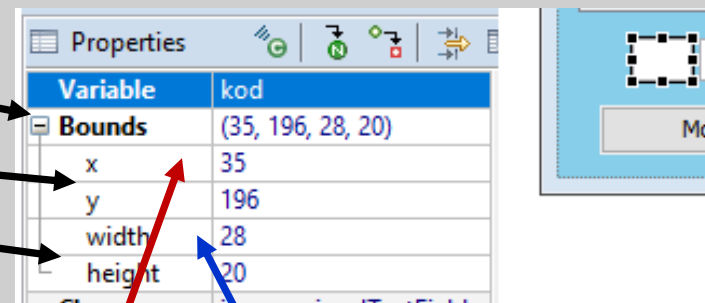
Módosítás Bezár

# Az EmpMod panel életre keltése

– Az egyes grafikus alkatrészeknek a méretmegadása a **Properties** területen, a **Bounds** paraméterekkel történik. (Nyissuk le!)

– Logika:

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

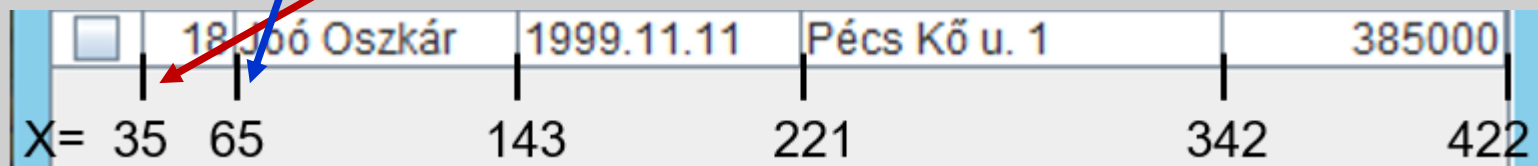


Variable	kod
<b>Bounds</b>	(35, 196, 28, 20)
x	35
y	196
width	28
height	20

– A szövegmezők között 3 pixelnnyi helyet érdemes kihagyni!

– Az első szövegmező beállítása:

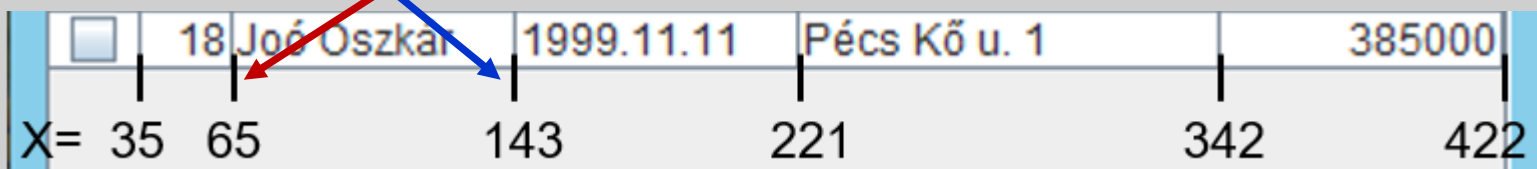
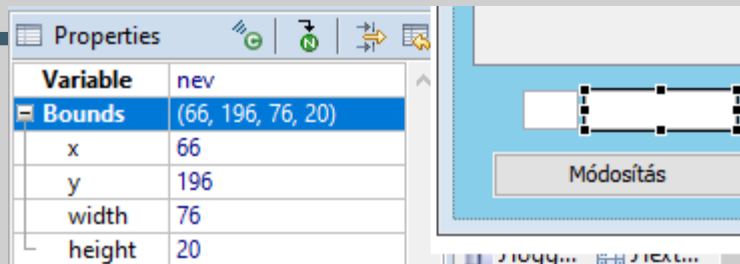
- Az **y** koordináta és **height** érték nem változik!
- **x** = 35 (az első leolvasott x koordináta)
- **width** =  $65 - 35 = 30$ , ebből levonok 2 pixelt helyre = 28



# Az EmpMod panel életre keltése

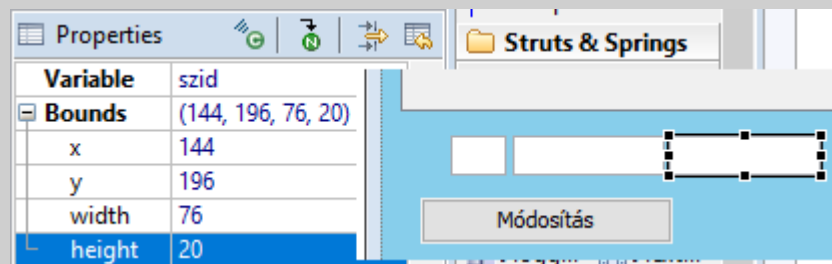
– A második szövegmező beállítása:

- $x = 65 + 1$  (hely) = 66
- $width = 143 - 65 = 78$ , ebből levonok 2 pixelt = 76



– A harmadik szövegmező beállítása:

- $x = 143 + 1$  (hely) = 144
- $width = 221 - 143 = 78$ , ebből levonok 2 pixelt = 76



– . . .

# 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 metódusokat.
- 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, csak logikai értéként visszaszadja, hogy ki van-e töltve egy szövegmező.
  - Legelőször ezt készítsük el:

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

Ismét egy elegáns  
**metódus túlterhelés!**  
Lassacskán, de  
megtanulunk Javában  
programozni!

# 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, mely megszámolja, hogy hány szövegmező van kitöltve.
- 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);
    }
    //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	Szüldő	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	Szüldő	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

i

A rekord módosítva!

OK

Dolgozók listája

4

Jel	Kód	Név	Szüldő	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

Listázás módosítás után

Bezár

70/61

BITMAN

# 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);  
}
```

- Oldjuk meg azt is, hogy a dátum ellenőrzése is megtörténjen!

Ha elakadtál küldj ímélt  
a címemre!

OnLine

**BITMAN**

HELP

Vagy tedd fel a táblád!

NonLine





# Az EmpMod2 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 Bezár

Adatok módosítása

Kód: 40

Név: Kis Pista Kis István

Szülidő: 1999.12.31

Lakóhely: Ózd Balatonboglár

Iq: 99 101 **3**

Módosít

Program üzenet

**4**

A rekord módosítva!

OK



# Az **EmpMod2** panel – Egyéni feladat



- Az új módosítás panelünkön oldjuk meg az egyes adatok ellenőrzését. (Hasonló a megoldás, mint az előbbi **EmpMod** panelen, a szükséges metódusok készen vannak)
- Oldjuk meg, hogy mikor megtörtént a módosítás:
  - Bezáródjon az **EmpMod2** panel,
  - Visszatérve a **Lista** panelre, szüntessük meg a kijelölést a listában, és töltsük újra a Lista panel adatait!

# Az EmpMod3 – Extra módosítás



- Gyakorlásként írjuk meg a módosítást úgy, hogy a módosítás funkció elindításakor a Törlés V2 funkciónál leírt listás panel jelenjen meg. Ha kiválasztottuk a listából a személyt, és megnyomtuk a Módosítás gombot, jelenjen meg a Módosítás2 panel, és ezen lehessen ténylegesen adatot módosítani!

A screenshot of a Windows-style dialog box titled "Dolgozók módosítása V2". It contains a label "Válaszd ki a módosítandó személyt:" followed by a dropdown menu with the text "Válassz!". A yellow circle with the number "1" is placed over the dropdown arrow. Below the dropdown are two buttons: "Módosítás" (with a yellow circle "2" over it) and "Bezár".

A screenshot of a dialog box titled "Adatok módosítása". It contains several input fields for personal data: "Kód:" (40), "Név:" (Kis Pista, Kis István), "Születő:" (1999.12.31), "Lakóhely:" (Ózd, Balatonboglár), and "Iq:" (99, 101). At the bottom right is a "Módosít" button with a yellow circle "3" over it.

A screenshot of a small dialog box titled "Program üzenet". It contains an information icon (i) and the text "A rekord módosítva!". At the bottom is an "OK" button with a yellow circle "4" over it.

# Jelenlét feladat

---

- Óra végéig küldjön egy emailt:
  - **Cím:** [szucs@iit.uni-miskolc.hu](mailto:szucs@iit.uni-miskolc.hu)
  - **Tárgy:** [JProg katalógus 210324](#)
  - A levél **szövegében** szerepeljen a [neve](#), kódja

# KATALÓGUS

# Pontot érő feladat



- Készítse el az extra módosítás funkciót
- Készítsen a két panelről képernyőképet (lehetnek egy képen is), és csatolja a jelenlét email-hez.
- A képeken látszódjon adatként a neve!

Adatok módosítása

Kód: 55

Név: BitMan

Szülidő: 2000.01.01

Lakóhely: Miskolc

Fizetés: 5000000

Módosít

Dolgozók módosítása V2

Válaszd ki a módosítandó személyt: 55 - BitMan

Módosítás Bezár

Aki nem a kért módon küldi a levelet, ne számítson pontra!

**BITMAN**



Question  
van?

**BITMAN**

# VÉGE

