

# PROGRAMOZÁS 3 BEADANDÓ

Iskola nyilvántartó rendszer



Gáspár Alex (BC6HU9)  
Németh Zsolt (KBYKGJ)

## Tartalom

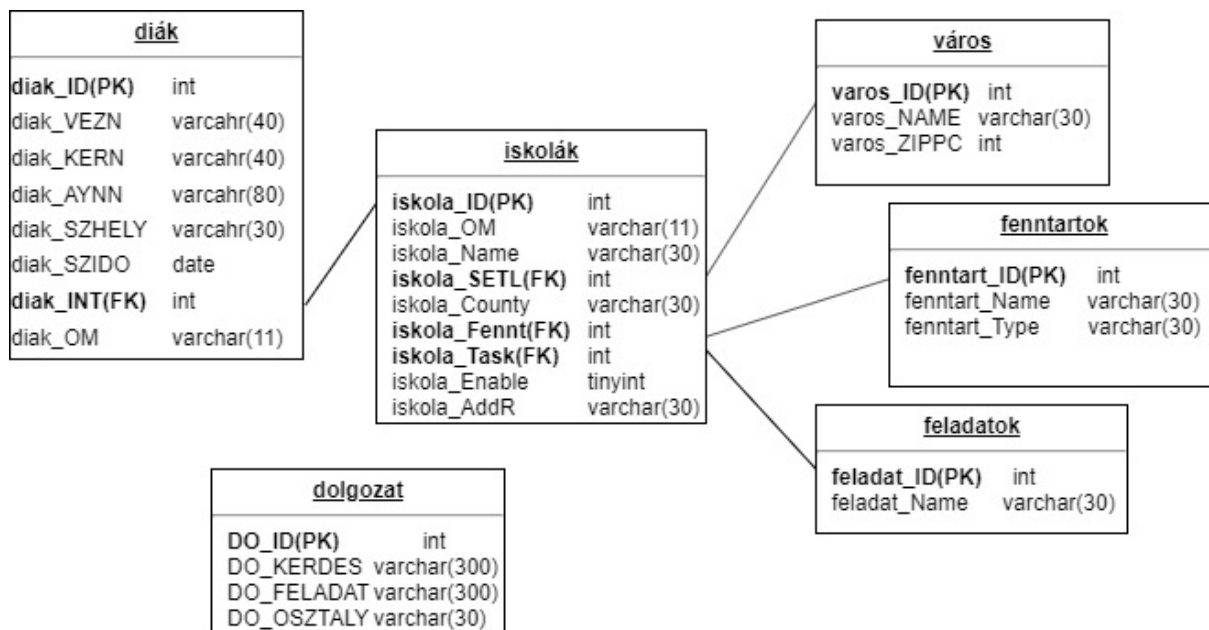
Specifikáció:.....	2
Feladat kidolgozása .....	3
Első lépés.....	3
1. kódrészlet.....	3
Lekérdezés modul.....	4
2. kódrészlet.....	4
3. kódrészlet.....	5
Feltöltés .....	5
4. kódrészlet.....	6
Iskolák tanulói .....	6
Dolgozat generátor .....	6
5. kódrészlet.....	8

## Specifikáció:

A program Java nyelven készült, adatbázisnak egy MySQL szervert használ. A feladat leírása:

- A program legyen könnyen kezelhető, áttekinthető.
- A program legyen képes az iskolák listázására, ez a lista legyen bővíthető.
- Lehessen az iskolák tanulói létszámából statisztikát készíteni.
- Legyen bővíthető (példaként bekerült egy dolgozat generáló modul).

Az adatbázis:



1. ábra Adatbázis tervezet 1.0

A kigondolt felület:



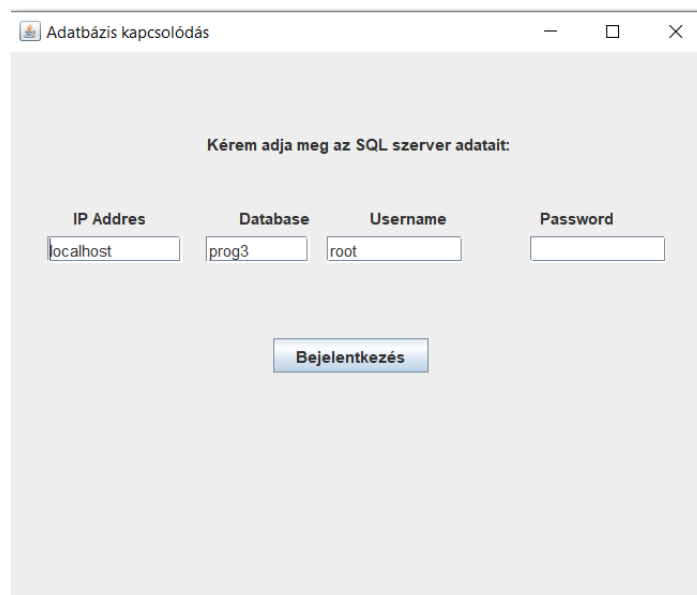
2. ábra Tervezési felület

Az adatbázisra való bejelentkezést egy külön ablakban valósítjuk meg. Minden modulnak saját füle van. Így áttekinthető és feladat specifikus kezelőfelületet kapunk. Próbáltunk mindent külön osztályokba rendezni a későbbi egyszerűbb fejleszthetőség illetve rendezettebb kód érdekében.

## Feladat kidolgozása

### Első lépés

Amikor a program elindul bekéri tőlünk az adatbázis eléréséhez szükséges adatokat majd ellenőrzi azokat, amennyiben sikeres volt tovább lép a fő programra, ha nem, értesít mindet.



3. ábra Adatbázis adatai

A megadott adatokat a kód összefűzi és ennek segítségével megpróbál kapcsolódni a szerverre, majd ellenőrzésként adatokat lekérni.

#### 1. kódrészlet

```
String con=jTextField1.getText();
String user=jTextField2.getText();
String pass=jPasswordField1.getSelectedText();
String data=jTextField3.getText();
SQL kapcsolat=new SQL(con,user,pass,data);
kapcsolat.query("select * from iskolák");

if((String.valueOf(kapcsolat.connect())).equals("true"))
{
    this.setVisible(false);
    new MainForm().setVisible(true);
    MainForm form=new MainForm();
    form.ip=con;
    form.felhasz=user;
    form.Data=data;
```

Amennyiben sikeres volt a kapcsolódás a főablakba kerülünk.

Iskola OM	Iskola neve	Település	Megye	Fenntartó	Cím	Feladat	Aktív
-----------	-------------	-----------	-------	-----------	-----	---------	-------

4. ábra Főablak

## Lekérdezés modul

A lekérdezés modulban lehetőségünk van az egyes iskolák és azok adatainak lekérdezésére. Az első lekérdezés után elérhetővé válnak az egyes szűrési funkciók, melyeket ki és be lehet kapcsolni.

### 2. kódrészlet

```
private void lekerdez_ButtonActionPerformed(java.awt.event.ActionEvent evt) {  
    if(!szur_akt_CheckBox.isSelected()){  
        controler.feltolt_tabla(iskolaTable);  
        controler.feltolt_combo(megye_ComboBox, telep_ComboBox, fenntarttip_ComboBox, feladattip_ComboBox, diaks  
    }else{  
        szur_akt_CheckBox.setSelected(false);  
        controler.feltolt_tabla(iskolaTable);  
        controler.feltolt_combo(megye_ComboBox, telep_ComboBox, fenntarttip_ComboBox, feladattip_ComboBox, d  
        szur_akt_CheckBox.setSelected(true);  
    }  
}
```

Az egyes szűrők közül egyszerre csak egy használható, mindig a legutoljára kiválasztott lesz érvényben. ha kikapcsoljuk a szűrőket a táblázat összes eleme ismét megjelenik.

### 3. kódrészlet

```
private void szur_akt_CheckBoxActionPerformed(java.awt.event.ActionEvent evt) {  
    if (!szur_akt_CheckBox.isSelected()) {  
        iskolaTable.setRowSorter(null);  
        megye_ComboBox.setEnabled(false);  
        telep_ComboBox.setEnabled(false);  
        fenntarttip_ComboBox.setEnabled(false);  
        feladattip_ComboBox.setEnabled(false);  
        aktiv_CheckBox.setEnabled(false);  
        kereso_sor.setEnabled(false);  
    } else {  
        megye_ComboBox.setEnabled(true);  
        telep_ComboBox.setEnabled(true);  
        fenntarttip_ComboBox.setEnabled(true);  
        feladattip_ComboBox.setEnabled(true);  
        aktiv_CheckBox.setEnabled(true);  
        kereso_sor.setEnabled(true);  
    }  
}
```

## Feltöltés

A feltöltés modulban adatokat vihetünk fel az adatbázisba. Itt újabb fűlek jelennek meg, ahol kiválaszthatjuk mivel szeretnénk bővíteni az adatbázist:

- diák
- fenntartó
- iskola

6. ábra Feltöltés

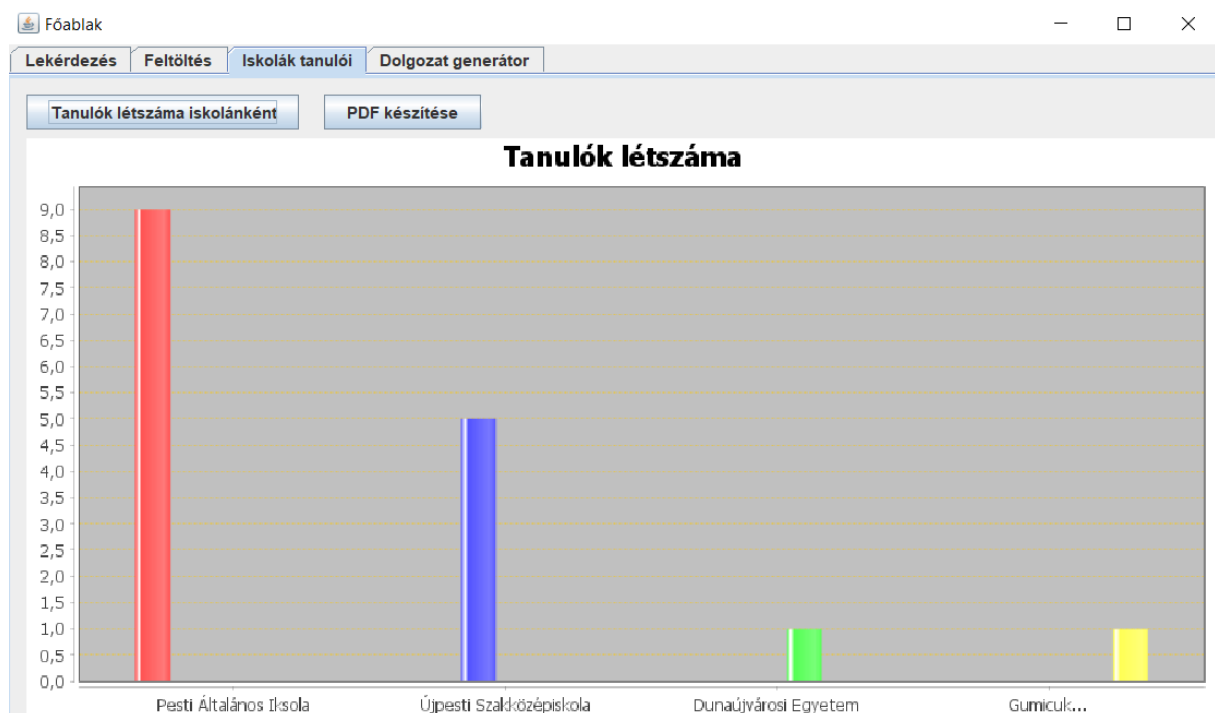
Az adatok bevitele után a „Feltöltés” gombbal indíthatjuk az adatok feltöltését, minden mező kitöltése kötelező, ha valamit kihagytunk hiba üzenetet kapunk, sikeres feltöltés esetén pedig erről kapunk üzenetet.

#### 4. kódrészlet

```
if(!"".equals(veznev)&&!"".equals(anyjanev)&&!"".equals(szuletesiIdo)&&!"".equals(diakOM)){
    SQL feltolt_Sql=new SQL(ip,felhasz,jelszo,Data);
    feltolt_Sql.Insert_Query(insertquery);
    diakveznev_TextField.setText(null);
    diakkerneve_TextField.setText(null);
    diakannya_TextField.setText(null);
    diakszulido_TextField.setText(null);
    diakom_TextField.setText(null);
    JOptionPane.showMessageDialog(null, "Sikeres feltöltés!", "Siker",1);
}else{
    JOptionPane.showMessageDialog(null, "Minden mező kitöltése kötelező!", "Figyelmeztetés",2);
}
```

### Iskolák tanulói

Ezen a felületen a tanulók létszámát kérhetjük le iskolánkénti bontásban, erről diagrammot kapunk melyet ezután kimenthetünk PDF-be.A diagram alkotáshoz és a PDF generáláshoz a jFreeChart, illetve az OrsonPDF pluginenket használtuk.

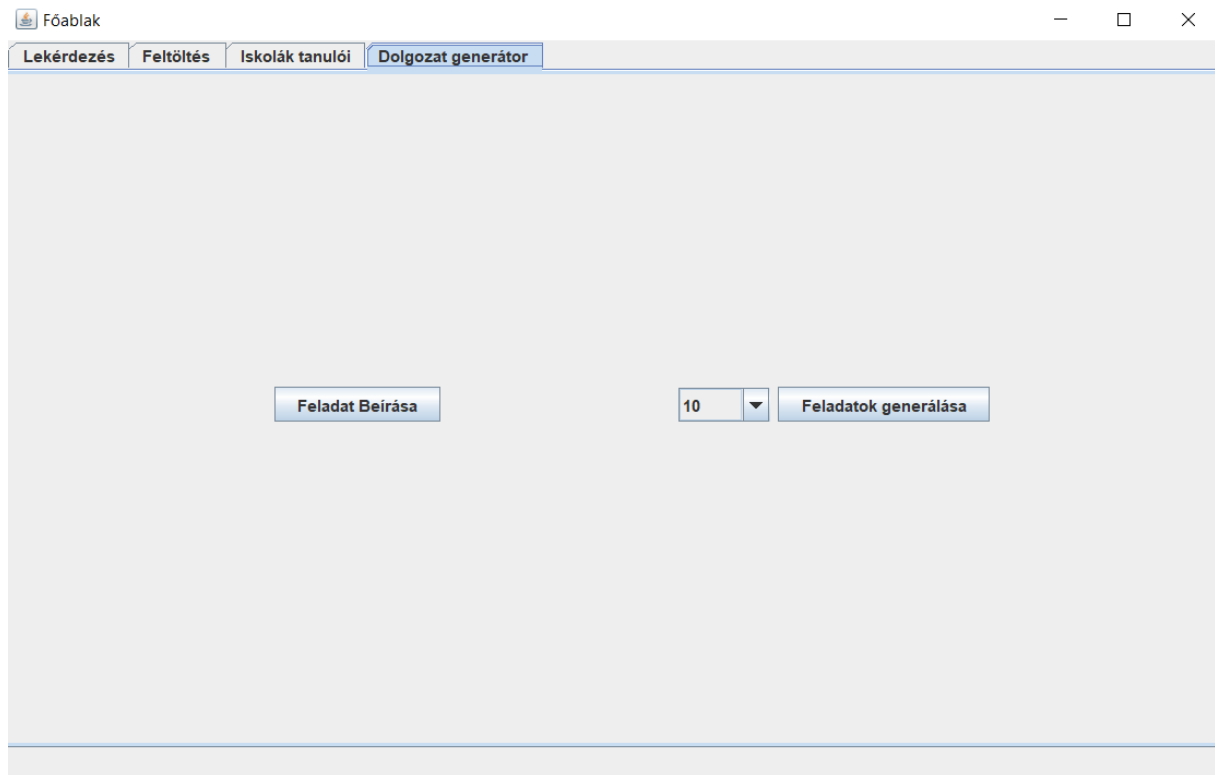


7. ábra Tanulók száma

### Dolgozat generátor

A dolgozat generátor modulban két dolgot tehetünk:

- kérdéseket tölthetünk fel az adatbázisba
- a feltöltött kérdésekből dolgozatot generálhatunk



8. ábra Dolgozat generátor

Amennyiben a „Feladat Beírása” menüt választjuk egy előugró eblakban lehetőségünk van jobb feladatok felvételére.



9. ábra Feladat beírása

Itt megadhatjuk a kérdés és a feladat szövegét, majd kiválaszthatjuk melyik osztályhoz szeretnénk a feladatot rendelni.

A másik funkció a „Feladatok generálása” ehhez előbb válasszuk ki hányadik osztályos tanulók részére szeretnénk feladatlapot generálni majd kattintsunk a gombra.



Feladatok Generálása

1 :Mi a kód végeredménye?

```

int a=5;
int b=10;
if(a>b)
    sysout("alma");
else
    sysout("körte");

```

2 :Mi a hiba a kódban?

```

for(int i=0;i<10<i+++);

```

3 :Mi a hiba a kódban?

```

int a="k";

```

4 :Mi a hiba a kódban?

```

int a="a";

```

5 :Mi a hiba a kódban?

```

int a="c";

```

*10. ábra Feladatok generálása*

A feladatok generálása során a program figyel, hogy ne legyen két egyforma feladat a listában. A feladatokról igény szerint kép készíthető.

### 5. kódrészlet

```

        visszateres=vegrehajjt.lekerdez(query);
        String[][] kerdesek=new String[row][2];
        int i=0;
        while(visszateres.next()){
            String feladat=visszateres.getString("DO_KERDES");
            String kerdes=visszateres.getString("DO_FELADAT");
            kerdesek[i][0]=feladat;
            kerdesek[i][1]=kerdes;
            i++;
        }
        String[] kerdes=new String[5];

        int y=0;
        ArrayList<JLabel> labels = new ArrayList<>();
        kerdes[0]=kerdesek[0][1];
        Random rnd=new Random();
        int szamlalo=0;
        boolean van=false;
JTextArea newLabel2 = new JTextArea(100,30);
        for(int j=0;j<5;j++)

```

A projekt github elérhetősége: <https://github.com/necrold/progos>