



Programozási technológia I.

Grafikus felhasználói felületek

Java - Grafikus felhasználói felületek

Keretek

UI események kezelése

Gomb használata

Keretikon megváltoztatása

Kattintások számlálása

Kilépés gomb

Kilépés egységes
kezelése

Ehelyezkedési módok

Dr. Szendrei Rudolf
Informatikai Kar
Eötvös Loránd Tudományegyetem



Tartalom

1 Java - Grafikus felhasználói felületek

Keretek

UI események kezelése

Gomb használata

Keretikon megváltoztatása

Kattintások számlálása

Kilépés gomb

Kilépés egységes kezelése

Elhelyezkedési módok



Keretek

UI események kezelése

Gomb használata

Keretikon megváltoztatása

Kattintások számlálása

Kilépés gomb

Kilépés egységes
kezelése

Ehelyezkedési módok

Java - Grafikus felhasználói felületek

Grafikus felhasználói felület

- Két csomag elemeiből lehet a felületet elkészíteni
 - `awt`: „heavy weight” komponensek
 - `swing`: „light weight” komponensek
- Mi hogyan használjuk?
 - `awt`: eseménykezelés, speciális elemek
 - `swing`: „látható” komponensek



Java - Grafikus felhasználói felületek

Swing komponensek

- Közös őszülő osztály: `JComponent`
- Származtatott osztályok:
 - `JFrame` keret
 - `JLabel` címke
 - `JButton` gomb
 - `JPanel` panel
 - `JComboBox` kombobox (legördülő menü)
 - `JList` lista
 - `JSlider` csúszka
 - `JSpinner` spin vezérlő
 - `JTable` táblázat



Keretek

Üres keret létrehozása

Két osztályt kell készítenünk:

- Főprogram osztálya: `Keret`

```
public class Keret
{
    public static void main(String[] args)
    {
        new ÜresKeret ();
    }
}
```

- Üres keret osztálya: `ÜresKeret`

```
import javax.swing.JFrame;

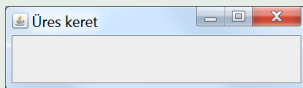
public class ÜresKeret extends JFrame
{
    public ÜresKeret ()
    {
        setTitle("Üres_keret");
        setSize(200, 60);
        setVisible(true);
    }
}
```



Keretek

Üres keret futtatása

A program futtatása során a következő ablak jelenik meg:



- A bezárás gombra kattintva a program ablaka bezárul, de **a program nem fejeződik be.**
 - parancssorban nem kapjuk vissza a promptot
 - a futtatási konzol ablak mutatja, hogy a program fut
- A bezárás gomb megnyomásával csak a keret tűnik el
- Program leállítása:
 - parancssorból `Ctrl+C` lenyomásával
 - NetBeans környezetben a `Run` menü `Stop Build/Run` pontjával



Keretek

Program leállítása a keret bezárásakor

- Az alapértelmezett megvalósítás szerint bezárásakor csak eltűnik a keret.
- Lehetséges megoldások:
 - A bezárás kori viselkedés megadása
 - A bezárás eseményének figyelése eseménykezelő hozzárendelésével



Keretek

Program leállítása a keret bezárásakor – automatikusan

A kerethez rendelt automatikus bezáró művelet előírása:

- `setDefaultCloseOperation` művelet segítségével
- Paramétere: adott konstans érték lehet (`WindowConstants`)

```
import javax.swing.*;  
  
public class ÜresKeret extends JFrame  
{  
    public ÜresKeret ()  
    {  
        setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);  
        setTitle("Üres_keret");  
        setSize(200, 60);  
        setVisible(true);  
    }  
}
```




Keretek

Program leállítása a keret bezárásakor – eseménykezelővel

- A kezelőt a `WindowAdapter` osztályból kell származtatni
- A kezelőt az `addWindowListener` művelettel kell a kerethez rendelni
- A kezelő egyetlen feladata az ablak bezárásának kezelése, amit a `windowClosing` művelet átdefiniálásával tehetünk meg
- A művelet paramétere a bezárást okozó `WindowEvent` típusú esemény, amit nem használunk
- A műveletben ki kell lépni a programból
- Ha kilépéskor egyéb tevékenységet is akarunk végezni, akkor mindenképpen ezt az utat kell követni



Keretek

Program leállítása a keret bezárásakor – eseménykezelővel

```
import javax.swing.*;
import java.awt.event.*;

public class ÜresKeret extends JFrame
{
    public ÜresKeret ()
    {
        addWindowListener(new WindowAdapter ()
        {
            @Override
            public void windowClosing(WindowEvent e)
            {
                System.exit(0);
            }
        });

        setTitle("Üres_keret");
        setSize(200, 60);
        setVisible(true);
    }
}
```



Keretek

Program leállítása a keret bezárásakor – eseménykezelővel

Az „implicit” objektum helyett az objektumot felvehetjük az osztályon belül

```
import javax.swing.*;
import java.awt.event.*;

public class ÜresKeret extends JFrame
{
    private WindowAdapter kilépés = new WindowAdapter ()
    {
        @Override
        public void windowClosing(WindowEvent e)
        {
            System.exit (0);
        }
    };

    public ÜresKeret ()
    {
        addWindowListener(kilépés);
        setTitle("Üres_keret");
        setSize(200, 60);
        setVisible(true);
    }
}
```



Keretek

Program leállítása a keret bezárásakor – eseménykezelővel

Létrehozhatjuk az eseménykezelőt külön osztályként is, így azt több keret esetében újra felhasználhatjuk majd

```
import java.awt.event.*;

public class Kilépés extends WindowAdapter
{
    @Override
    public void windowClosing(WindowEvent e)
    {
        System.exit(0);
    }
}
```



Keretek

Program leállítása a keret bezárásakor – eseménykezelővel

Az eseménykezelő osztályt ezután az alábbi módon használhatjuk

```
import javax.swing.*;  
  
public class ÜresKeret extends JFrame  
{  
    public ÜresKeret ()  
    {  
        addWindowListener(new Kilépés ());  
        setTitle("Üres_keret");  
        setSize(200, 60);  
        setVisible(true);  
    }  
}
```



UI események kezelése

UI események

- A felhasználói akciók a felületi elemeken eseményt váltanak ki: `ActionEvent`
- Ezeket kell kezelni az elemekkel úgy, hogy eseményfigyelőt rendelünk hozzájuk (`addActionListener` vagy `Action` objektum beágyazása)
- Események figyelése: `ActionListener` interfész
- Egyetlen művelet: `actionPerformed`, aminek paramétere az esemény



Példa - Kattintások számlálása

Gomb elhelyezése a kereten

- A számláló gombot a keret „adatterületére” helyezzük el, amit a `getContentPane()` művelet ad meg
- Ehhez az `add` művelettel vehetünk hozzá elemet
- A gombot a `SzámlálóGomb` osztály segítségével valósítjuk meg
- Ezt a `JButton` osztályból kell származtatni, és ki kell egészíteni eseménykezeléssel is
- Az osztálynak meg kell valósítania ezért az `ActionListener` interfészt



Keret ikonjának beállítása

- A keret ikonját a `setIconImage` művelettel adhatjuk meg
- A művelet paramétere az ikon, amit a megfelelő eszközön (`Toolkit`) keresztül rendelünk a kerethez.

```
setIconImage(Toolkit.getDefaultToolkit().getImage("gomb.png"));
```

- A `Toolkit` az `awt` csomag része
- Így a `jar` állományból nem töltődik be az ikon. Ehhez a fájl nevét URL-ként kell megadni.

```
java.net.URL url =  
Gomb.class.getResource("gomb.png");  
setIconImage(Toolkit.getDefaultToolkit().getImage(url));
```



```
import java.awt.*;
import javax.swing.*;

public class ÜresKeret extends JFrame
{
    public ÜresKeret ()
    {
        addWindowListener (new Kilépés ());
        setTitle ("Kattintások_számlálása");
        setSize (600, 100);
        getContentPane ().add (new SzámlálóGomb ());
        java.net.URL url = ÜresKeret.class.getResource ("gomb.png");
        setIconImage (Toolkit.getDefaultToolkit ().getImage (url));
        setVisible (true);
    }
}
```



```
import java.awt.event.*;
import javax.swing.*;

public class SzámlálóGomb extends JButton implements ActionListener
{
    private int kattintások = 0;
    public SzámlálóGomb ()
    {
        setText ("Kattintás:_0");
        addActionListener (this);
    }

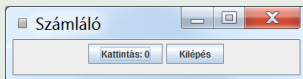
    @Override
    public void actionPerformed (ActionEvent e)
    {
        setText ("Kattintás:_ " + ++kattintások);
    }
}
```



Példa - Kattintások számlálása

+1: Kilépés gomb

- Egészítsük ki az előző programot úgy, hogy a keretben helyezzünk el egy kilépésre lehetőséget adó gombot is



- El kell készítenünk az előzőekhez hasonlóan a kilépésre szolgáló gombot, ahol az esemény kezelése a program befejezését jelenti
- A kereten most nem egy, hanem két gombot kell elhelyeznünk, megadva azok elhelyezkedési módját (`setLayout`)
- A legegyszerűbb lehetőség a `FlowLayout`, amelybe folytonosan helyezhetünk el elemeket



```
import java.awt.*;
import javax.swing.*;

public class ÜresKeret extends JFrame
{
    public ÜresKeret ()
    {
        addWindowListener(new Kilépés ());
        setTitle ("Számláló");
        setSize (400, 100);
        getContentPane ().setLayout (new FlowLayout ());
        getContentPane ().add (new SzámlálóGomb ());
        getContentPane ().add (new KilépésGomb ());

        java.net.URL url = ÜresKeret.class.getResource ("gomb.png");
        setIconImage (Toolkit.getDefaultToolkit ().getImage (url));
        setVisible (true);
    }
}
```



Példa - Kattintások számlálása

Kilépés egységes kezelése

- A programból kilépéskor ugyanaz következzen be, bárhogy is lépünk ki a programból (még a mostani egyszerű esetben is)
- Célszerű az ablak bezárásához és a kilépés gomb lenyomásához ugyanazt az eljárást rendelnünk (kilép)
- A bezárásához eseménykezelőt rendelünk, amelyben ezt hívjuk meg, és a gomb megnyomásakor is ezt hívjuk meg
- A kilépés gombot most másként hozzuk létre
- Egy gomb konstruktorának paramétere lehet egy `Action` típusú objektum
- Ebben meg lehet adni az esemény kezelését, illetve a gomb további tulajdonságait
- Az `Action` interfészt megvalósító `AbstractAction` osztály, alapértelmezésben üres megvalósításokkal látja el (csak a számunkra „érdekeseket” kell megvalósítanunk)



```
import java.awt.event.*;
import javax.swing.*;

public class KilépésGomb extends JButton
{
    public KilépésGomb ()
    {
        super(new AbstractAction("Kilépés")
        {
            @Override
            public void actionPerformed(ActionEvent e)
            {
                new Kilépés().windowClosing(null); // System.exit(0);
            }
        });
    }
}
```



Ehelyezkedési módok

Ehelyezkedési módok

- `FlowLayout` – az elemek folyamatos elhelyezése sorban
- `BorderLayout` – az elemek elhelyezése 5 helyre:
 - középre (`CENTER`)
 - alulra (`SOUTH`) / felülre (`NORTH`)
 - balra (`WEST`) / jobbra (`EAST`)
- `GridLayout` – az elemek folyamatos elhelyezése megadott méretű rácsban, ugyanolyan méretben
- `GridBagLayout` – az elemek elhelyezése rácsban, az elemek kiterjedése több egység is lehet
- `BoxLayout` – elemek elhelyezése adott méretben, adott távolságra vízszintes vagy függőleges irányban
- `CardLayout` – kártyapakli elhelyezés (csak egy elem látszik, ami változtatható)
- `OverlayLayout` – elemek egymásra helyezése
- <https://docs.oracle.com/javase/tutorial/uiswing/layout/visual.html>