

## Keret (ablak) kezelés

Az ablakok kezeléséhez a *javax.swing* csomag osztályait fogjuk használni, mivel ezek platformfüggetlenséget és könnyű kezelhetőséget biztosítanak.

Példa: Hozzunk létre egy üres keretet, fejléccel.

Megbeszélés:

1. *JFrame* osztály,
2. ablakméret beállítása: *setSize()* metódus,
3. fejléc szövege: *setTitle()* metódus,
4. ablak bezárása: *EXIT\_ON\_CLOSE*,
5. az ablak megjelenítése: *setVisible()* metódus.

Megoldás:

```
package jframe_1;
import javax.swing.JFrame;
public class JFrame_1 {
    public static void main(String[] args) {
        JFrame Keret = new JFrame();
        Keret.setSize(500, 500);
        Keret.setTitle("Az első ablakom");
        Keret.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        Keret.setVisible(true);
    }
}
```

Példa:

Készítsük el az előző feladatot úgy, hogy új osztályt hozunk létre az ablak kezelésére. Egészítsük ki a feladatot egy szöveg kiírással és az ablak automatikus méretezésével.

Megbeszélés:

1. *JLabel* osztály,
2. ablakméret beállítása: *setSize()* metódus,
3. ablakméret automatikus beállítása: *pack()* metódus.

Megoldás:

```
package jframe_2;
import javax.swing.JFrame;
public class JFrame_2 {
    public static void main(String[] args) {
        Keret k = new Keret();
        k.Ablak();
    }
}
class Keret extends JFrame{
    public void Ablak(){
        setTitle("Az első ablakom");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        // setSize(500, 500);
        add(new JLabel("Ez egy új szöveg"));
        pack();
        setVisible(true);
    }
}
```

Példa:

Hozzunk létre egy tetszőleges méretű ablakot egy megadott pozícióba. Írjunk ki legalább három sor szöveget különböző beállításokkal.

Megbeszélés:

1. ablak helyének megadása: *setLocation()* metódus,
2. az elrendezés beállítása *setLayout()* metódus,
3. szövegdoboz helyének és méretének megadása *setBounds(x, y, w, h)*,
4. szövegdoboz háttérszínének megadása: *setBackground()* metódus,
5. szöveg színének megadása: *setForeground()* metódus,
6. a *Color* osztály,
7. szöveg vízszintes igazítása a szövegdobozon belül: *setHorizontalAlignment()* metódus,
8. szöveg függőleges igazítása a szövegdobozon belül: *setVerticalAlignment()* metódus,

Megoldás:

```
package JFrame_3;
import java.awt.Color;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.SwingConstants;

public class JFrame_3 {
    public static void main(String[] args) {
        Keret k = new Keret();
        k.Ablak();
    }
}

class Keret extends JFrame{
    JLabel szoveg;
    public void Ablak(){
        setTitle("A második ablakom");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setSize(500, 500);
        setLocation(200, 200);
        setLayout(null);
        szoveg = new JLabel("Ez az első sor szövege");
        szoveg.setOpaque(true);
        szoveg.setBackground(Color.blue);
        szoveg.setForeground(Color.WHITE);
        szoveg.setHorizontalAlignment(SwingConstants.LEFT);
        szoveg.setVerticalAlignment(SwingConstants.TOP);
        szoveg.setBounds(10, 10, 200, 50);
        add(szoveg);

        szoveg = new JLabel("Ez a második sor szövege");
        szoveg.setOpaque(true);
        szoveg.setBackground(Color.YELLOW);
        szoveg.setForeground(Color.CYAN);
        szoveg.setHorizontalAlignment(SwingConstants.CENTER);
```

```

        szoveg.setVerticalAlignment(SwingConstants.CENTER);
        szoveg.setBounds(100, 100, 200, 50);
        add(szoveg);

        szoveg = new JLabel("Ez a harmadik sor szövege");
        szoveg.setOpaque(true);
        szoveg.setBackground(Color.green);
        szoveg.setForeground(Color.red);
        szoveg.setHorizontalAlignment(SwingConstants.RIGHT);
        szoveg.setVerticalAlignment(SwingConstants.BOTTOM);
        szoveg.setBounds(200, 200, 200, 50);
        add(szoveg);

        setVisible(true);
    }
}

```

**Példa:**

Kérjünk be billentyűzetről neveket. Számoljuk meg a nevekben lévő karaktereket, majd mentjük el egy szöveges fájlba a nevek sorszámát, a neveket és a nevek hosszát. A fájl tartalmát egy ablakban írjuk ki.

**Megoldás:**

```

package label_1;
import java.awt.GridLayout;
import static java.lang.System.out;
import java.util.Scanner;
import java.util.StringTokenizer;
import java.io.File;
import java.io.PrintStream;
import java.io.IOException;
import javax.swing.JFrame;
import javax.swing.JLabel;
public class Label_1 {
    public static void main(String[] args) {
        Beolvas be = new Beolvas();
        Kiír ki = new Kiír();
        ki.Ki();
    }
}
class Beolvas{
    public Beolvas(){
        Scanner billentyu = new Scanner (System.in, "ISO8859_2");
        String nev;
        int ssz = 1;
        try{
            try (PrintStream file = new PrintStream("Fájl.txt")) {
                out.print("Kérek egy nevet: ");
                nev = billentyu.nextLine();
                while(nev.length()!=0){

```

```

        file.printf("%d-%s-%d\n", ssz, nev, nev.length());
        ssz++;
        out.print("Kérek egy nevet: ");
        nev = billentyu.nextLine();
    }
}
}
catch (IOException error) {
    System.err.println("Hiba: " + error.getMessage());
}
}
}

class Kiír extends JFrame {
    public void Ki() {
        String sor;
        int sorok=0;
        StringTokenizer token;
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        try (Scanner file = new Scanner(new File("Fájl.txt"))) {
            while(file.hasNext()) {
                sor = file.nextLine();
                sorok++;
                token = new StringTokenizer(sor, "-");
                add(new JLabel(" " + (String) token.nextElement()));
                add(new JLabel((String) token.nextElement()));
                add(new JLabel((String) token.nextElement()));
            }
            file.close();
        }
        catch (IOException error) {
            System.err.println("Hiba: " + error.getMessage());
        }
        finally{
            setTitle("Névsor");
            setLayout(new GridLayout(sorok, 3, 200, 30));
            pack();
            setVisible(true);
        }
    }
}

```

**Feladatok:**

1. Írjon programot, amely e program forrásállományát (.java fájl) kiírja a képernyőre egy keretben.
2. Írjon programot, amely e program forrásállományát (.java fájl) kiírja a képernyőre egy keretben, két színű, soronként váltakozó háttérszinnal.
3. Írjon programot, amely e program forrásállományát (.java fájl) kiírja a képernyőre egy keretben.
4. Írjon programot, amely e program forrásállományát (.java fájl) kiírja a képernyőre két keretbe úgy, hogy minden páratlan sort az egyikbe és minden páros sort a másikba ír.
5. Írjon programot, amely e program forrásállományát (.java fájl) kiírja a képernyőre két keretbe úgy, hogy minden páratlan sort az egyikbe és minden páros sort a másikba ír, valamint a szöveg karakterszínét soronként váltogatja.
6. Írjon programot, amely e program forrásállományát (.java fájl) kiírja a képernyőre soronként egy-egy keretbe úgy, hogy a keret címe legyen a sor szövege.
7. Írjon programot, amely e program forrásállományát (.java fájl) kiírja a képernyőre egy keretbe úgy, hogy a szöveg két hasábos legyen.