

Példa:

Igazítsuk középre az ablakot és tiltsuk le az átméretezés lehetőségét.

Megbeszélés:

1. az ablak méretének lekérdezése *getScreenSize()* metódus,
2. ablak méretének tárolása, *Dimension* osztály.
3. *setResizable(false)* metódus,
4. *FlowLayout()* metódus.

Megoldás:

```
package label_2;
import java.awt.*;
import javax.swing.*;
public class Label_2 {
    JFrame ablak;
    JLabel cimke1;
    JLabel cimke2;
    public static void main(String[] args) {
        Ablak ob = new Ablak();
    }
}
class Ablak extends JFrame {
    public Ablak(){
        JFrame ablak = new JFrame("Ablak középre");
        JLabel szöveg1 = new JLabel("Első sor");
        JLabel szöveg2 = new JLabel("Második sor");
        ablak.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        ablak.setLayout(null);
        ablak.setSize(800, 600);
        setCenter(ablak);
        ablak.setResizable(false);
        // ablak.setLayout(new FlowLayout());
        szöveg1.setBounds(10, 10, 200, 30);
        szöveg2.setBounds(30, 30, 200, 30);
        ablak.add(szöveg1);
        ablak.add(szöveg2);
        ablak.setVisible(true);
    }
    private static void setCenter(Window frame){
        Dimension közép = Toolkit.getDefaultToolkit().getScreenSize();
        int x = (int) ((közép.getWidth() - frame.getWidth()) / 2);
        int y = (int) ((közép.getHeight() - frame.getHeight()) / 2);
        frame.setLocation(x, y);
    }
}
```

Példa:

Írjuk ki egy szöveges fájl tartalmát egy görgethető ablakba.

Megbeszélés:

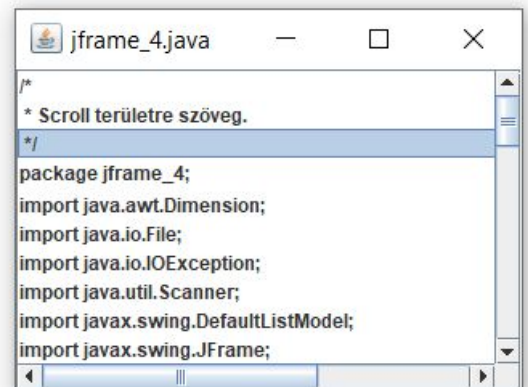
1. a *JList* osztály,
2. a *JScrollPane* osztály.

Megoldás:

```
package jframe_4;
import java.awt.Dimension;
import java.io.File;
import java.io.IOException;
import java.util.Scanner;
import javax.swing.DefaultListModel;
import javax.swing.JFrame;
import static javax.swing.JFrame.EXIT_ON_CLOSE;
import javax.swing.JList;
import javax.swing.JScrollPane;

public class Jframe_4 {
    public static void main(String[] args) {
        Kiír k = new Kiír();
        k.Ki();
    }
}

class Kiír extends JFrame {
    public void Ki() {
        String sor;
        DefaultListModel<String> lm = new DefaultListModel<>();
        JList<String> jl = new JList<>(lm);
        JScrollPane scroll = new JScrollPane(jl);
        scroll.setPreferredSize(new Dimension(300, 200));
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        setLocation(400,400);
        try (Scanner file = new Scanner(new File("src\\jframe_4\\jframe_4.java"))) {
            while(file.hasNext()) {
                sor = file.nextLine();
                lm.addElement(sor);
            }
            file.close();
        }
        catch (IOException error) {
            System.err.println("Hiba: " + error.getMessage());
        }
        finally{
            add(scroll);
            setTitle("jframe_4.java ");
            pack();
            setVisible(true);
        }
    }
}
```



Példa:

Jelenítsünk meg egy képet egy teljesképernyős ablakban.

Megbeszélés:

1. a *paintComponent()* metódus a grafikus megjelenítéshez szükséges, a rendszer hívja meg,
2. várakozás a kép betöltésére: *MediaTracker* osztály, *waitForID()* metódus,
3. a *getImage()* metódus,
4. a *gr.drawImage(image, xbf, ybf, xja, yja, null)* metódus.

Megoldás:

```
package image_2;
import java.awt.Graphics;
import java.awt.Image;
import java.awt.MediaTracker;
import java.awt.Toolkit;
import javax.swing.JFrame;
import javax.swing.JPanel;
public class Image_2 {
    public static void main(String[] args) {
        Kép k = new Kép();
        k.kép();
    }
}
class Kép extends JFrame{
    public void kép(){
        setSize(Toolkit.getDefaultToolkit().getScreenSize());
        getContentPane().add(new Kép_be());
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setVisible(true);
    }
}
class Kép_be extends JPanel{
    private Image im;
    public Kép_be(){
        MediaTracker tr = new MediaTracker(this);
        im = Toolkit.getDefaultToolkit().getImage("Kép.jpg");
        tr.addImage(im, 0);
        try{
            tr.waitForID(0);
        }
        catch(InterruptedException err){}
        finally{
            tr.removeImage(im, 0);
        }
    }
    @Override
    protected void paintComponent(Graphics gr){
        super.paintComponent(gr);
        gr.drawImage(im, 100, 100, 400, 400, this);
    }
}
```

Komponensek használata, eseménykezelés, interfész

A különböző komponensek használatához szintén a *javax.swing* csomag osztályait fogjuk használni.

Amikor egy grafikus felhasználói felületet (GUI-t) használunk, az azon elvégzett bármilyen műveletet (billentyűleütés, egérmozgatás, egérgomb lenyomás stb.) eseménynek nevezzük. Ahhoz, hogy a programunk reagálni tudjon eseményre, azt figyelnie kell és rendelkeznie kell egy eseménykezelő programrésszel (kóddal).

Az **interfész** metódusfejeket definiál, a metódusoknak csak a formáját adja meg, azok az adott osztályban kerülnek megvalósításra. A megvalósító (implementáló) osztály egy speciális utóda az interfésznek, egy osztály több interfészt is örökölhet, és más metódusokat is tartalmazhat. Egy interfész implementálásakor az interfész minden metódusát ki kell fejteni akkor is, ha nem használjuk.

Példa:

Készítsük el egy korábbi kitalálós játék új verzióját, használjunk keretet, szövegmező, gomb és címke komponenseket.

Megbeszélés:

1. a *JButton*, *JLabel*, *TextField* osztályok,
2. az *addActionListener()* metódus,
3. az *actionPerformed()* metódus,
4. egy interfész metódusának implementációja: *@Override* annotáció,

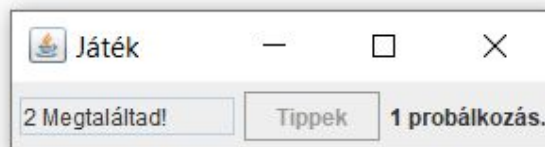
Megoldás:

```
package button_2;
import java.awt.FlowLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.util.Random;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JTextField;
public class Button_2 {
    public static void main(String[] args) {
        Uj_jatek ob = new Uj_jatek();
        ob.jatek();
    }
}
class Uj_jatek extends JFrame implements ActionListener {
    int random = new Random().nextInt(5)+1;
    int probak = 0;
    JTextField text = new JTextField(10);        //10 oszlop szélességű mező
    JButton button = new JButton("Tippek");
    JLabel label = new JLabel(probak + " próbálkozás");
    public void jatek() {
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLocation(400, 400);
        setTitle("Játék");
    }
}
```

```

        setLayout(new FlowLayout());           //Komponensek egymás után
        add(text);
        add(button);
        add(label);
        pack();
        setVisible(true);
        button.addActionListener(this);       //Eseményfigyelő
    }
    @Override
    public void actionPerformed(ActionEvent e) {
        String szoveg = text.getText();
        if(Integer.parseInt(szoveg) == random) {
            button.setEnabled(false);
            text.setText(text.getText() + " Megtaláltad!");
            text.setEditable(false);
        }
        else {
            text.setText("");
            text.requestFocus();
            probak++;
            label.setText(probak + " próbálkozás.");
        }
    }
}

```



Példa:

Készítsük egy programot, amely forint-euró átváltást valósítja meg.

Megbeszélés:

1. a *KeyListener*, *ItemListener*, *MouseListener* interfészek,

Megoldás:

```
package button_3;
import java.awt.Color;
import java.awt.FlowLayout;
import java.awt.event.ItemEvent;
import java.awt.event.ItemListener;
import java.awt.event.KeyEvent;
import java.awt.event.KeyListener;
import java.awt.event.MouseEvent;
import java.awt.event.MouseListener;
import java.text.NumberFormat;
import java.util.Locale;
import javax.swing.JComboBox;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JTextField;

public class Button_3 {
    public static void main(String[] args) {
        Valto pénz = new Valto();
        pénz.valto();
    }
}

class Valto extends JFrame implements KeyListener, ItemListener, MouseListener {
    JLabel sourceLabel = new JLabel(" ");
    JTextField text = new JTextField(5);
    JLabel label = new JLabel(" ");
    JComboBox<String> combo = new JComboBox<>();
    NumberFormat forint = NumberFormat.getCurrencyInstance(Locale.getDefault());
    NumberFormat euro = NumberFormat.getCurrencyInstance(Locale.FRANCE);

    public void valto() {
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLayout(new FlowLayout());
        setLocation(400, 400);
        setTitle("Pénzváltó");
        add(sourceLabel);
        add(text);
        add(label);
        combo.addItem("Forintról Euróra"); //A legördülő lista összeállítása
        combo.addItem("Euróról Forintra");
        add(combo);
        setSize(400, 100);
    }
}
```

```

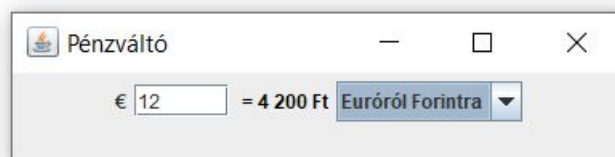
        setVisible(true);
        text.addKeyListener(this);
        label.addMouseListener(this);
        combo.addItemListener(this);
    }
    void cimke() {
        String amountSt = "";
        String fromSt = "";
        try {
            double amount = Double.parseDouble(text.getText());
            (combo.getSelectedItem().equals("Forintról Euróra")) {
                amountSt = " = " + euro.format(amount * 0.002857);
                fromSt = "Ft";
            }
            (combo.getSelectedItem().equals("Euróról Forintra")) {
                amountSt = " = " + forint.format(amount * 350);
                fromSt = "\u20AC";
            }
        }
        catch (NumberFormatException e) {}
        label.setText(amountSt);
        sourceLabel.setText(fromSt);
    }
    @Override
    public void keyReleased(KeyEvent k) {        //Amikor felengedtek egy billentyűt
        cimke();
    }
    @Override
    public void keyPressed(KeyEvent k) {        //Lenyomtak egy billentyűt
    }
    @Override
    public void keyTyped(KeyEvent k) {        //Alfanumerikus billentyűt lenyomtak,
        //majd felengedték
    }
    @Override
    public void itemStateChanged(ItemEvent i) { // A legördülő listából kiválasztottak
        cimke();        //egy elemet
    }
    @Override
    public void mouseEntered(MouseEvent m) { //Egér a komponens felett
        label.setForeground(Color.RED);
    }
    @Override
        public void mousePressed(MouseEvent m) { //Amig az egérgomb nyomva
    }

```

```

@Override
public void mouseReleased(MouseEvent m) { //Egérgomb felengedése
    label.setForeground(Color.green);
}
@Override
    public void mouseClicked(MouseEvent e) { //Egér kattintás egy komponensen
}
@Override
public void mouseExited(MouseEvent e) { //Egér elhagyja a komponens
    label.setForeground(Color.BLACK);
}
}

```



Példa:

Készítsük egy programot, amely a szövegmező hátterszínét változtatja különböző gombok megnyomására. Egészítsük ki egy Exit (kilépés) gombbal.

Megoldás:

```

package button_3;
import java.awt.Color;
package jframe_teszt;
import java.awt.Color;
import java.awt.FlowLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JTextField;
public class Jframe_teszt {
    public static void main(String[] args) {
        Teszt t = new Teszt();
        t.teszt();
    }
}

```




```

class Teszt extends JFrame implements ActionListener {
    JFrame ablak = new JFrame();
    JTextField text = new JTextField(10);
    JButton Rbutton = new JButton("Piros");
    JButton Ybutton = new JButton("Sárga");
    JButton Gbutton = new JButton("Zöld");
    JButton Exit = new JButton("Exit");
    public void teszt() {
        ablak.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        ablak.setLayout(new FlowLayout());
        ablak.setLocation(400, 400);
        ablak.setTitle("Teszt");

        ablak.add(Rbutton);
        ablak.add(Ybutton);
        ablak.add(Gbutton);
        ablak.add(Exit);
        ablak.add(text);

        ablak.setSize(400, 150);
        ablak.setVisible(true);

        Rbutton.addActionListener(this);
        Ybutton.addActionListener(this);
        Gbutton.addActionListener(this);
        Exit.addActionListener(this);
    }
    @Override
    public void actionPerformed(ActionEvent e) {
        if(e.getSource() == Rbutton)text.setBackground(Color.red);
        else if(e.getSource() == Ybutton)text.setBackground(Color.yellow);
        else if(e.getSource() == Gbutton)text.setBackground(Color.green);
        else if(e.getSource() == Exit)System.exit(0);
    }
}

```

Feladatok:

1. Egy keretben jelenítsen meg három képet egyszer vízszintes, majd függőleges elrendezésben.
2. Módosítsa a színező programot (utolsó megbeszél programot) úgy, hogy legalább 9 színt jelenít meg. A gombokat jelenítse meg 3x3-as mátrixban.
3. Készítsük el az előző feladatot úgy, hogy a színeket legördülő mezőből jelöljük ki.
4. Egy keretben jelenítsen meg két szövegmezőt 4 gombot és egy „label”-t. A két szövegmezőbe írjunk számokat, a gombok segítségével adjuk össze, vonjuk ki egymásból, szorozzuk össze és osszuk el azokat. Az eredményt a „label”-be írjuk ki.
5. Egy keretben jelenítsen meg egy képet véletlen helyen. Amikor az egérrel rámegyünk a képre, az ugorjon egy másik véletlen helyre.