

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 (billentyűütés, egérmozgatás, egérgomb lenyomás stb.) eseménynek nevezünk. 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 interfészét is örökölhet, és más metódusokat is tartalmazhat. Egy interfész implementálásakor az interfész minden metódusát ki kell fejteni akkor is, ha nem használjuk.

Példa:

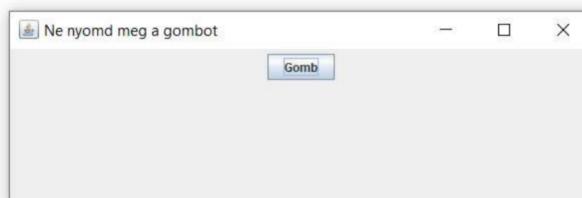
Hozzunk létre egy keretet, egyetlen gombbal.

Megbeszélés:

1. a *JButton* osztály,
2. komponensek elhelyezése *FlowLayout()*.

Megoldás:

```
package button_1;
import java.awt.FlowLayout;
import javax.swing.JButton;
import javax.swing.JFrame;
public class Button_1 {
    public static void main(String[] args) {
        Gomb ob = new Gomb();
        ob.keret();
    }
}
class Gomb extends JFrame {
    public void keret() {
        setTitle("Ne nyomd meg a gombot");
        setLayout(new FlowLayout()); //Komponensek egymás után
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        add(new JButton ("Gomb"));
        setSize(600,200);
        setVisible(true);
    }
}
```



Hallgatói feladat: Adjunk a kerethez három gombot más-más felirattal.

Példa:

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

Megbeszélés:

3. *JLabel*, *JTextField* osztályok,
4. az *addActionListener()* metódus,
5. az *actionPerformed()* metódus,
6. egy interfész metódusának implementációja: *@Override* annotáció,

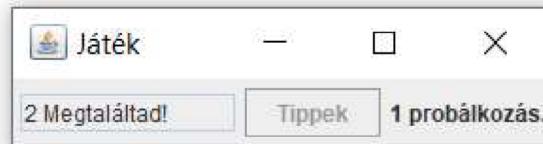
Megoldás:

```
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());
        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.");
    }
}
}

```



Hallgatói feladat: Készítsen egy kockadobó játékot két kockával. Gomb kiklikkésre indul a dobás, labelben jelenik meg a dobott érték.

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());
            if(combo.getSelectedItem().equals("Forintról Euróra")) {
                amountSt = " = " + euro.format(amount * 0.002857);
                fromSt = "Ft";
            }
            else if(combo.getSelectedItem().equals("Euróról Forintra")) {
                amountSt = " = " + forint.format(amount * 350);
                fromSt = "\u02AC";
            }
        }
        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) { //Egér gomb felengedése
    label.setForeground(Color.green);
}
@Override
    public void mouseClicked(MouseEvent e) { //Egér kattintás egy komponensen
}
@Override
public void mouseExited(MouseEvent e) {           //Egér elhagyja a komponenst
    label.setForeground(Color.BLACK);
}

```



Hallgatói feladat: Egy keretben jelenítsen meg két szövegmezőt és egy labelt. Hasonlítsa össze a szövegmezőbe írt számokat, a kisebb értékét írja ki a labelben.

Példa:

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

Megoldás:

```
package button_4;
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 button_4{
    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);  
}  
}
```



Hallgatói feladat: Egy keretben jelenítsen meg két szövegmezőt, írjon beléjük tetszőleges számot. Hasonlítsa össze a számokat, a kisebb háttérszínét változtassa zöldre, a nagyobbét pirosra.

Hallgatói feladat: Készítsen programot, amely agy akkumulátor töltöttségét jelzi. Egy szövegmezőbe írjon számokat, amely az akkumulátor töltöttséget mutatják százalékban. Ha a beírt szám 50% feletti, akkor zöld, ha 20-50% közötti akkor sárga és ha 20% alatti akkor piros legyen a szövegmező háttérszíne.

Példa:

Próbálunk ki néhány rajzoló metódust.

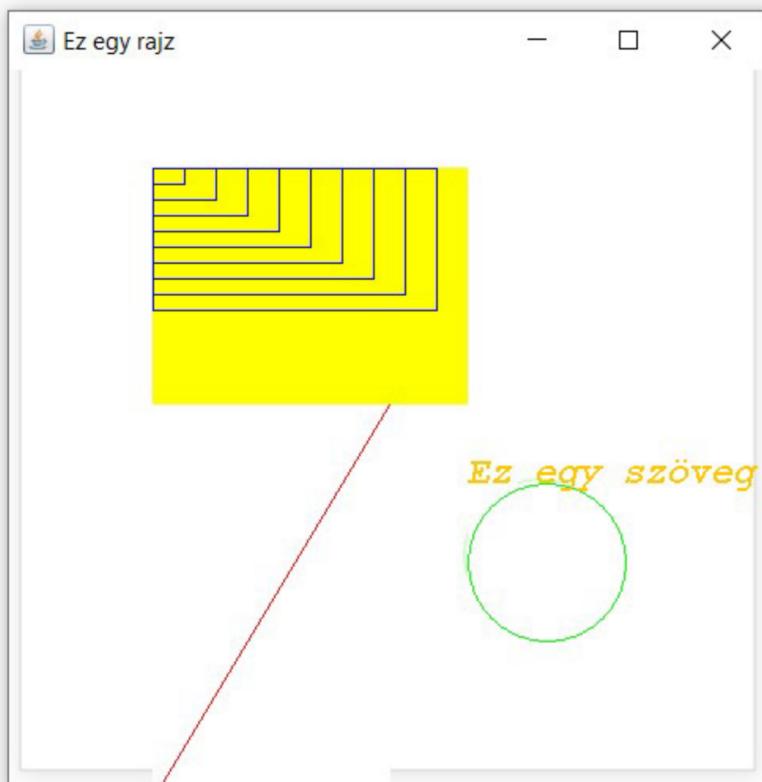
Megbeszélés:

1. a *setColor()* és *setFont()* metódusok,
2. a *drawRect()* és *fillRect()* metódusok,
3. *drawOval()* és *drawLine()* metódusok,
4. *drawString()* metódus.

Megoldás:

```
package graphics_1;
import java.awt.Color;
import java.awt.Font;
import java.awt.Graphics;
import javax.swing.JFrame;
public class Graphics_1 {
    public static void main(String[] args) {
        Keret keret = new Keret(new Rajz());
        keret.setVisible(true);
    }
}
class Keret extends JFrame {
    Rajz rajz;
    public Keret (Rajz rajz) {
        this.rajz = rajz;
        set();
    }
    private void set(){
        setTitle("Ez egy rajz");
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        setSize(500, 500);
        setLocation(1200,200);
    }
    @Override
    public void paint(Graphics g) {
        rajz.paint(g);
    }
}
class Rajz {
    public void paint ( Graphics gr ){
        gr.setColor( Color.yellow );
        gr.fillRect( 100, 100, 200, 150 );
        gr.setColor( Color.blue );
        for(int i=1; i<10; i++)
            gr.drawRect(100, 100, i*20, i*10 );
        gr.setColor( Color.green );
        gr.drawOval(300, 300, 100, 100);
```

```
        gr.setColor( Color.red );
        gr.drawLine(250, 250, 100, 500);
        gr.setColor(Color.ORANGE);
        gr.setFont(new Font ("Courier New", 3, 24));
        gr.drawString("Ez egy szöveg", 300, 300);
    }
}
```



Feladatok:

1. Egy keretben jelenítsen meg három képet egyszer vízszintes, majd függőleges elrendezésben.
2. Módosítsa a színező programot (utolsó megbeszélt 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 írunk 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.
6. Egy keretben jelenítsen meg két szövegmezőt, írjon beléjük tetszőleges szöveget. Hasonlítsa össze a szövegeket. Ha a két szöveg megegyezik, minden szövegmező háttérszínét változtassa zöldre, ellenkező esetben pirosra.