

Példa:

Próbáljunk ki néhány rajzoló metódust.

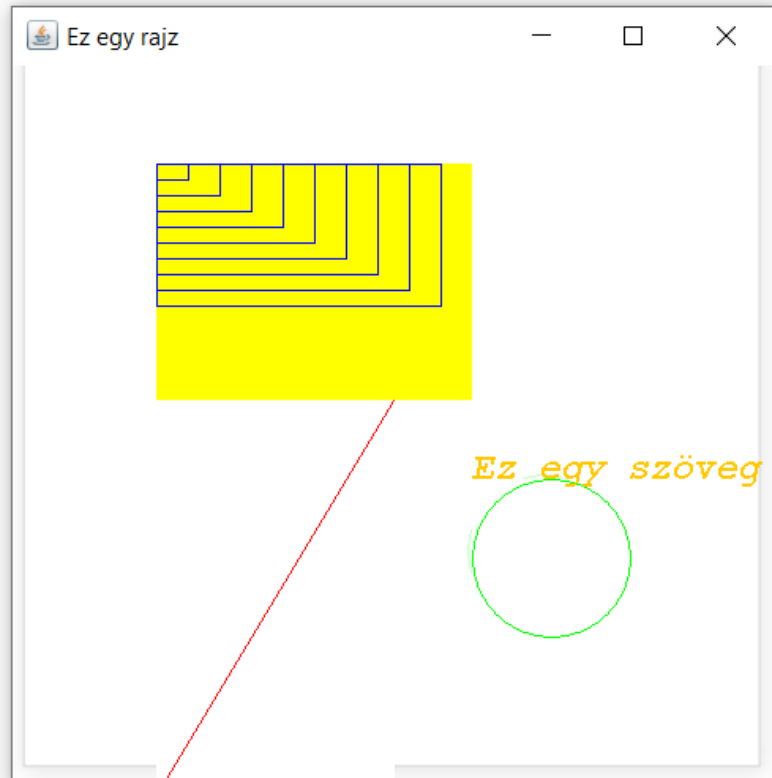
Megbeszélés:

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);  
}  
}
```



## Párbeszédablak

A párbeszéd- vagy dialógusablakot általában adatbevitelre és üzenetküldésre használjuk.

Példa:

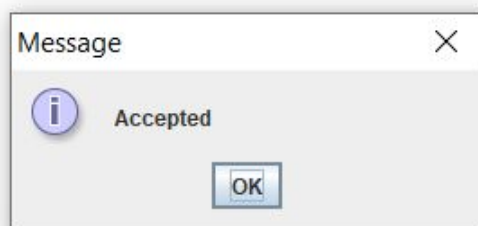
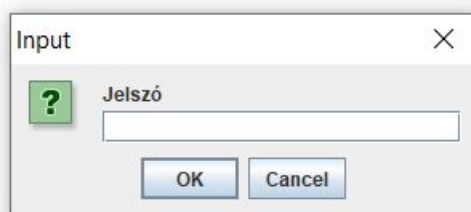
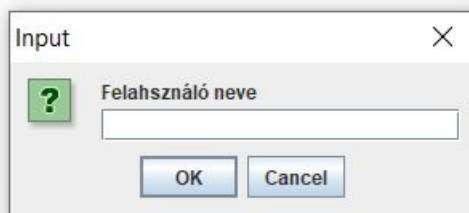
Egy korábban elkészített jelszóellenőrző programunkat módosítsuk úgy, hogy az adatbevitelre és a kiírásra dialógus ablakokat használunk.

Megbeszélés:

1. a *JOptionPane* osztály,
2. a `showInputDialog()` metódus,
3. a `showMessageDialog()` metódus.

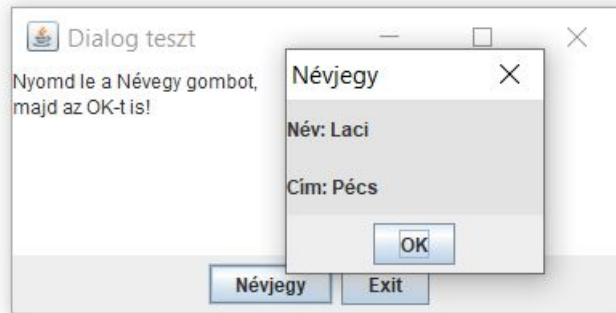
Megoldás:

```
package dialog_1;
import javax.swing.JOptionPane;
public class Dialog_1 {
    public static void main(String[] args) {
        String név = JOptionPane.showInputDialog("Felhasználó neve");
        String jelszó = JOptionPane.showInputDialog("Jelszó");
        if(név != null && jelszó != null && név.equals("Iaci") && jelszó.equals("1234"))
            JOptionPane.showMessageDialog(null, "Accepted");
        else
            JOptionPane.showMessageDialog(null, "Rejected");
    }
}
```



Példa:

Jelenítsünk meg egy dialógusablakot, rajta a képen látható szöveggel és tiltsuk le a bezárását (csak az „Exit” gombra záródjon be). A „Névjegy” gomb lenyomására jelenjen meg egy „Névjegy” ablak, amit nem lehet átméretezni.



Megoldás:

```
package dialog_2;

import java.awt.BorderLayout;
import java.awt.Container;
import java.awt.GridLayout;
import java.awt.SystemColor;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.JButton;
import javax.swing.JDialog;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JTextArea;

public class Dialog_2 {
    public static void main(String[] args) {
        DialogTeszt teszt = new DialogTeszt();
    }
}

class DialogTeszt extends JFrame implements ActionListener {
    Container cont = getContentPane();
    JButton Név, Exit;

    public DialogTeszt () {
        setDefaultCloseOperation(DO_NOTHING_ON_CLOSE);
        setBounds(200,200,400,200);
        cont.add(new JTextArea("Nyomd le a Névegy gombot,\nmajd az OK-t is!"));
        JPanel panel = new JPanel();
        setTitle("Dialog teszt");
        panel.add(Név = new JButton("Névjegy"));
        panel.add(Exit = new JButton("Exit"));
        cont.add(panel, BorderLayout.SOUTH);
        Név.addActionListener(this);
    }
}
```

```

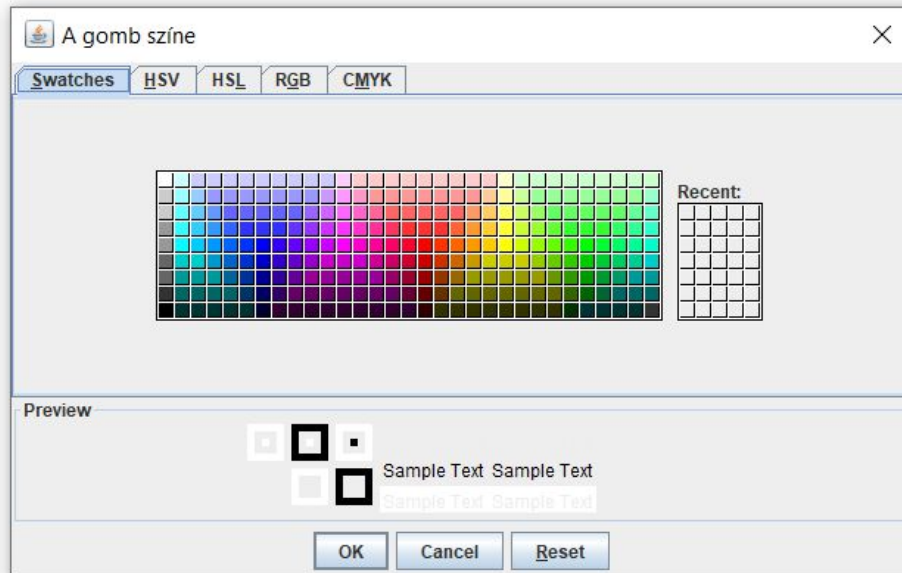
        Exit.addActionListener(this);
        setVisible(true);
    }
    @Override
    public void actionPerformed(ActionEvent e) {
        if(e.getSource() == Név) new Névjegy(this);
        else if(e.getSource() == Exit)System.exit(0);
    }
}

class Névjegy extends JDialog implements ActionListener{
    Container cont = getContentPane();
    JButton OK;
    public Névjegy(JFrame own){
        super(own, "Névjegy", true);
        setLocation(getParent().getX() + 200, getParent().getY() + 50);
        setResizable(false);
        cont.setBackground(SystemColor.controlHighlight);
        cont.setLayout(new GridLayout(3,1));
        cont.add(new JLabel("Név: Laci", JLabel.LEFT));
        cont.add(new JLabel("Cím: Pécs", JLabel.LEFT));
        JPanel ok = new JPanel();
        cont.add(ok);
        ok.add(OK=new JButton("OK"));
        OK.addActionListener(this);
        pack();
        setVisible(true);
    }
    @Override
    public void actionPerformed(ActionEvent e) {
        dispose();        //Ablak bezárása és a memória felszabadítása-
    }
}

```

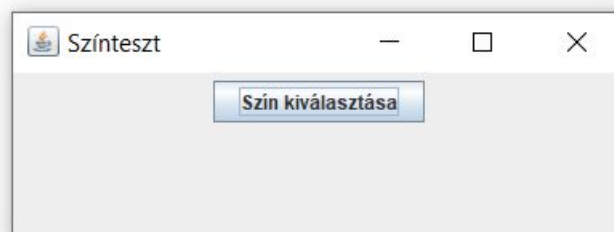
## Színdialógus használata

A felhasználó, a színválasztó dialógus (*JColorChooser*) segítségével tetszőleges színt kiválaszthat, illetve összeállíthat, ami egy színobjektumban (*Color*) kerül visszaadásra. A dialógus a program grafikus felhasználói felületén belül bárhol elhelyezhető.



Példa:

Helyezzünk el egy keretbe nyomógombot. A gomb lenyomására jelenjen meg a színdialógus, az azon kiválasztott színnel színezzük át a gomb hátterét.



Megbeszélés:

1. a *Color* osztály,
2. a *JColorChooser* osztály.

Megoldás:

```
package dialog_3;
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.JColorChooser;
import javax.swing.JFrame;
public class Dialog_3 {
    public static void main(String[] args) {
```

```

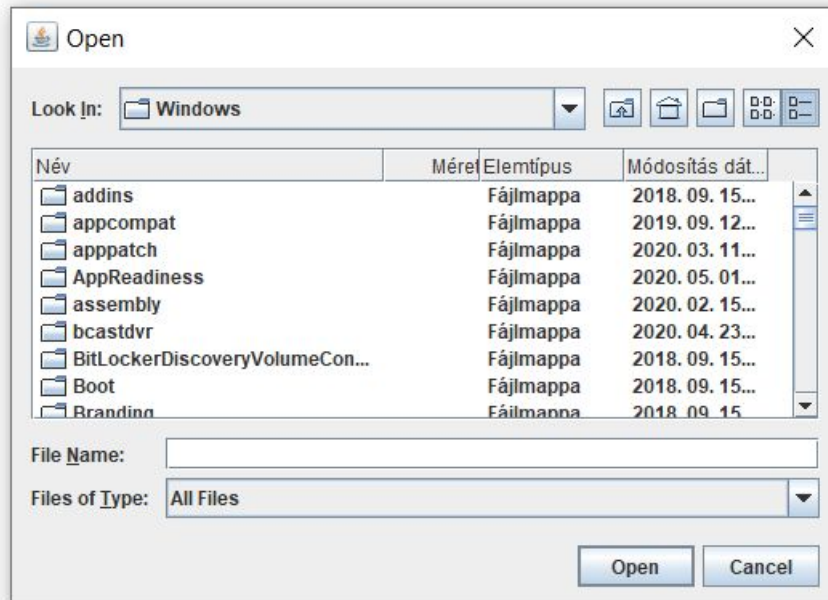
        ColorTest colorTest = new ColorTest();
        colorTest.Test();
    }
}

class ColorTest extends JFrame implements ActionListener{
    private JButton btszin = new JButton("Szín kiválasztása");
    private Color color;
    public void Test(){
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        setSize(400,150);
        setTitle("Színteszt");
        setLayout(new FlowLayout());
        add(btszin);
        setVisible(true);
        btszin.addActionListener(this);
        color=btszin.getBackground();
    }
    @Override
    public void actionPerformed(ActionEvent e) {
        color=JColorChooser.showDialog(this, "A gomb színe", color);
        if(color != null)btszin.setBackground(color);
    }
}

```

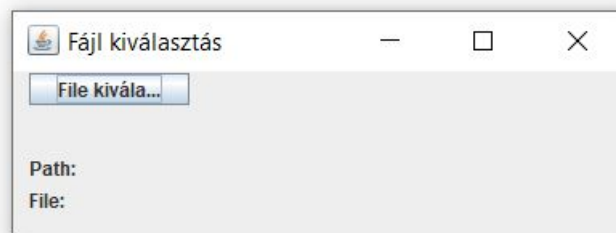
## Állománykiválasztó dialógus használata

Az állománykiválasztó dialógusablak (*JFileChooser*) feladata egy abszolút elérési útvonal bekérése. A dialógusablak grafikus felhasználói felületet biztosít a fájlrendszerben való navigáláshoz, a fájl vagy könyvtár kiválasztásához, vagy azok megadásához. A dialógus a program grafikus felhasználói felületén belül bárhol elhelyezhető.



Példa:

Helyezzünk el egy keretbe nyomógombot és két címkét (labelt). A gomb lenyomására jelenjen meg az állománykiválasztó dialógusablak, az azon kiválasztott fájl elérési útvonalát és a fájl nevét jelenítsük meg.



Megbeszélés:

1. a *JFileChooser* osztály,
2. ha kiválasztottunk egy állományt: *JFileChooser.APPROVE\_OPTION*.

Megoldás:

```
package dialog_file;

import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.File;
import javax.swing.JButton;
import javax.swing.JFileChooser;
import javax.swing.JFrame;
import static javax.swing.JFrame.EXIT_ON_CLOSE;
```



```

import javax.swing.JLabel;
public class Dialog_file {
    public static void main(String[] args) {
        FileTest file = new FileTest();
        file.Test();
    }
}

class FileTest extends JFrame implements ActionListener{
    private JButton bt = new JButton("File kiválasztása");
    private JFileChooser fc = new JFileChooser();
    private JLabel lbDir = new JLabel("Path: ");
    private JLabel lbFile = new JLabel("File: ");
    public void Test(){
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        setSize(400,150);
        setTitle("Fájl kiválasztás");
        setLayout(null);
        bt.setBounds(10, 1, 100, 20);
        lbDir.setBounds(10, 50, 400, 20);
        lbFile.setBounds(10, 70, 400, 20);
        add(bt);
        add(lbDir);
        add(lbFile);
        setVisible(true);
        bt.addActionListener(this);
    }
    @Override
    public void actionPerformed(ActionEvent e) {
        if (e.getSource() == bt)
            if(fc.showOpenDialog(this) == JFileChooser.APPROVE_OPTION){
                File sf = fc.getSelectedFile();
                if(sf != null){
                    lbDir.setText("Path: " + fc.getCurrentDirectory().getAbsolutePath());
                    lbFile.setText("File: " + sf.getName());
                }
            }
    }
}

```

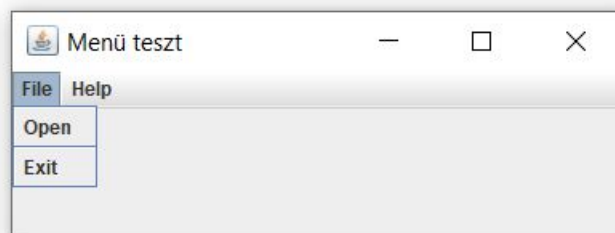
## Menüsor

A menük használatával felhasználóbaráttá tehetjük a programunkat. Csoportosíthatjuk, hierarchikus rendbe szedhetjük a program különböző funkcióit.

A menü-hierarchia tetején egy menüsor (*JMenuBar*) van, amely lelógó (pull down) menükből (*JMenu*) áll. A menü tovább bontható: menütételre (*JMenuItem*), jelölő-menütételre (*JCheckBoxMenuItem*) és rádió-menütételre (*JRadioButtonMenuItem*).

### Példa:

Készítsük el az ábrán látható menüsört. A **Open**-re egy fájlválasztó dialógusablak jelenjen meg, az **Exit**-re lépjen ki a program.



### Megoldás:

```
package menü_1;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.JFileChooser;
import javax.swing.JFrame;
import javax.swing.JMenu;
import javax.swing.JMenuBar;
import javax.swing.JMenuItem;
public class Menü_1 {
    public static void main(String[] args) {
        Menü m = new Menü();
        m.menü("Menü teszt");
    }
}
class Menü extends JFrame implements ActionListener{
    JMenuItem exit;
    JMenuItem open;
    private final JFileChooser fc = new JFileChooser();
    public void menü(String cim){
        setSize(400,150);
        setTitle(cim);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        JMenuBar mb;
        JMenu File;
        setJMenuBar(mb=new JMenuBar());
```

```
        mb.add(File=new JMenu("File"));
        mb.add(new JMenu("Help"));

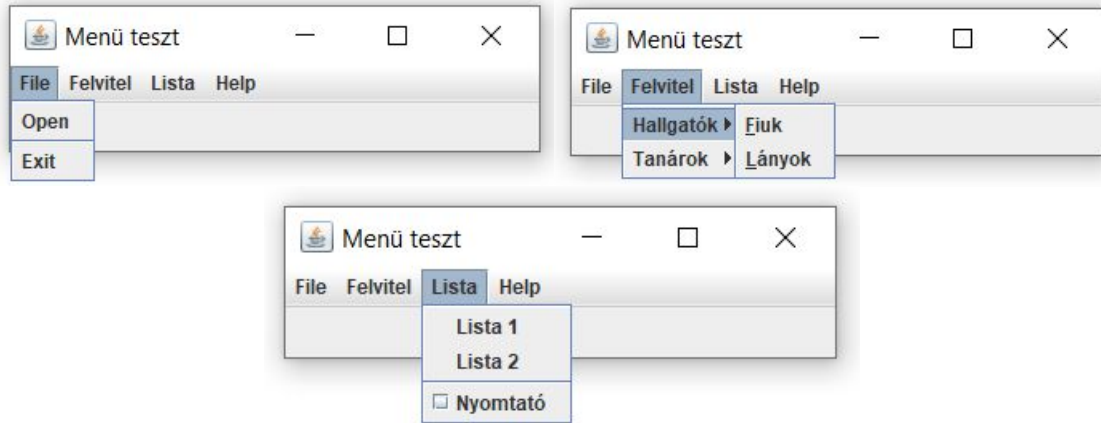
        File.add(open = new JMenuItem("Open"));
        File.addSeparator();
        File.add(exit = new JMenuItem("Exit"));

        setVisible(true);
        open.addActionListener(this);
        exit.addActionListener(this);
    }

    @Override
    public void actionPerformed(ActionEvent e) {
        if (e.getSource() == open)fc.showOpenDialog(this);
        else if(e.getSource()==exit) System.exit(0);
    }
}
```

Példa:

Készítsük el az ábrákon látható menüsört. A **File** főmenü megegyezik az előző programban látottal. A **Felvétel** főmenüben **Hallgatók** és **Tanárok** menü, a Hallgatók menüben **Fiuk**, **Lányok** menütételnél használjunk emlékeztető karaktert. A **Lista** főmenüben legyen egy **Nyomtató** jelölő tétel. A **Fiuk**, **Lányok**, **Nyomtató** egy emlékeztető sort írjon ki a konzolra, a **Lista1**, **Lista2** egy-egy keretet jelenítsen meg.

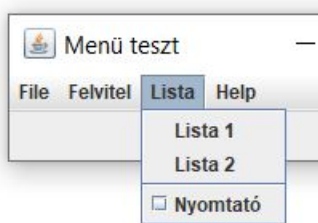
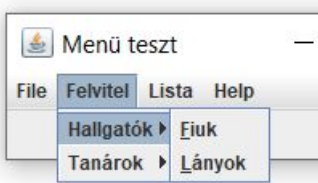


Megoldás:

```
package menü_2;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import static java.lang.System.out;
import javax.swing.ImageIcon;
import javax.swing.JCheckBoxMenuItem;
import javax.swing.JFileChooser;
import javax.swing.JFrame;
import static javax.swing.JFrame.EXIT_ON_CLOSE;
import javax.swing.JMenu;
import javax.swing.JMenuBar;
import javax.swing.JMenuItem;
import javax.swing.JMenuItem;

public class Menü_2 {
    public static void main(String[] args) {
        Menü m = new Menü();
        m.menü("Menü teszt");
    }
}

class Menü extends JFrame implements ActionListener{
    JMenuBar mb;
    JMenuItem miExit;
    JMenuItem miOpen;
    JMenuItem miFiu;
    JMenuItem miLány;
    JMenuItem miLista1, miLista2;
    JMenuItem miHasznál, miNévjegy;
    JCheckBoxMenuItem cbNyomtató;
    private final JFileChooser fc = new JFileChooser();
```



```

public void menü(String cim){
    setBounds(500,200,350,100);
    setTitle(cim);
    setDefaultCloseOperation(EXIT_ON_CLOSE);

    setJMenuBar(mb=new JMenuBar());
    JMenu File;
    mb.add(File=new JMenu("File"));
    File.add(miOpen = new JMenuItem("Open"));
    File.addSeparator();
    File.add(miExit = new JMenuItem("Exit"));

    JMenu mFelvitel;
    mb.add(mFelvitel=new JMenu("Felvitel"));
    JMenu mHallgató = new JMenu("Hallgatók");
    mFelvitel.add(mHallgató);
    mFelvitel.add(new JMenu("Tanárok"));
    mHallgató.add(miFiu = new JMenuItem("Fiuk",'F'));
    mHallgató.add(miLány = new JMenuItem("Lányok",'L'));

    JMenu mLista = new JMenu("Lista");
    mb.add(mLista);
    mLista.add(miLista1 = new JMenuItem("Lista 1"));
    mLista.add(miLista2 = new JMenuItem("Lista 2"));
    mLista.addSeparator();
    mLista.add(cbNyomtató = new JCheckBoxMenuItem("Nyomtató"));

    JMenu mHelp;
    mb.add(mHelp = new JMenu("Help"));
    mHelp.add(miHasznál = new JMenuItem("Leírás", new
                                                ImageIcon("help.jpg")));
    mHelp.add(miNévjegy = new JMenuItem("Névjegy"));

    setVisible(true);
    miOpen.addActionListener(this);
    miExit.addActionListener(this);
    miFiu.addActionListener(this);
    miLány.addActionListener(this);
    miLista1.addActionListener(this);
    miLista2.addActionListener(this);
    cbNyomtató.addActionListener(this);
}

@Override
public void actionPerformed(ActionEvent e) {
    if (e.getSource() == miOpen)fc.showOpenDialog(this);
    else if(e.getSource()==miExit) System.exit(0);
    else if(e.getSource() == miLista1)lista1();
    else if(e.getSource() == miLista2)lista2();
    else if(e.getSource() == miFiu)out.println("Fiúk felvétele");
    else if(e.getSource() == miLány)out.println("Lányok felvétele");
}

```

```
        else if(e.getSource() == cbNyomtató)
            if(cbNyomtató.getState())out.println("Nyomtatóra");
            else out.println("Nem nyomtatóra");
    }
    void lista1(){
        JFrame Lista = new JFrame("Lista1");
        Lista.setDefaultCloseOperation(DISPOSE_ON_CLOSE);
        Lista.setLocation(getX() + 10, getY() + 60);
        Lista.setSize(300,100);
        Lista.setVisible(true);
    }
    void lista2(){
        JFrame Lista = new JFrame("Lista2");
        Lista.setDefaultCloseOperation(DISPOSE_ON_CLOSE);
        Lista.setLocation(getX() + 100, getY() + 60);
        Lista.setSize(300,100);
        Lista.setVisible(true);
    }
}
```

**Feladatok:**

1. Menü és dialógusablak felhasználásával adja meg egy képfájl nevét. A tartalmát egy külön keretben jelenítse meg.
2. Menü és dialógusablak felhasználásával adja meg egy szövegfájl nevét. A tartalmát egy külön keretben jelenítse meg.
3. Menük segítségével készítsen egy rajzoló programot. Menü segítségével válasszon ki rajzobjektumokat (vonal, kör, téglalap stb.) amelyek paramétereit a billentyűzetről kérje be. A rajzolás színét menü és dialógusablak segítségével adja meg.
4. Készítse el a tavalyi C-ben megírt adatnyilvántartó házi feladat menüjét. Csak a menüt kell összeállítani. A különböző menüpontok kijelölését egy-egy szövegsor kiírásával jelezze a konzolon. (Ld. utolsó mintafeladat.)
5. Egy keretben jelenítsen meg két szövegmezőt, amelyekbe billentyűzetről megadhatunk számokat. Készítsünk olyan menüt, amely segítségével összeadjuk, kivonjuk, összeszorozzuk és elosztjuk a beírt számokat. Az eredményt a „label”-be írjuk ki.