**JFrame**

O fereastra GUI pentru a adauga componente.

* Import javax.Swing.JFrame;

Odata create, fereastra nu e vizibila, de aceea trebuie setata:

* setVisible(true) - o face vizibila
* setSize(420,420) – redimensioneaza
* setTitle() – seteaza titlu
* setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE) – face ca programul sa opreasca la apasarea lui X

EXIT\_ON\_CLOSE – o opreste

HIDE\_ON\_CLOSE

DO\_NOTHING\_ON\_CLOSE – nu permite sa o inchizi

* setResiazable(false) – nu permite modificarea marimii
* setIconImage(icon.getImage());

javax.Swing.ImageIcon;

ImageIcon icon = new ImageIcon(“”); pentru a crea o iconca

* getContentPane().setBackground(Color.culoare); - modifica culoarea

import java.awt.Color;

* getContentPane().setBackground(new Color(0,0,0)); - culoare RGB
* getContentPane().setBackground(new Color(0xFFFFFF)); - culoare hexadecimala
* add(label…) – adauga un strat
* setLayout(label) – seteaza un strat ca principal pe fereastra
* pack() – face ca toate obiectele sa fie redimensionate automat odata cu fereastra

**Labbels**

Pe el se pun imagini, text si el se pune pe fereastra

* setText(“”);
* JLabel label = new JLabel(“text”);
* setIcon(ImageIcon);
* setHorizontalTextPosition(JLabel.Center\Left\Right) – se ia In raport cu celelate obiecte
* setVerticalTextPosition(JLabel.Top\Center\Bottom) – se ia in raport cu celelalte obiecte
* setForeground(culoare) – modifica culoarea textului
* setFont(new Font(“Font”,Font.BOLD,marime)); - seteaza fontul,culoarea si marimea
* setIconTextGap(100) – indeparteaza textul de obiect
* setBackground(color)
* setOpaque(true) – arata backgroundul

Border border = new BorderFactory.createLineBorder(Color,marime)

* setBorder(border) – seteaza linii de jur imprejur
* setVerticalAlignment(JLabel.)
* setHorizontalAlignment(JLabel.) – aliniaza obiectele in label
* setBounds(x,y,lungime,latime) – seteaza pozitia si marimea stratului
* setLocation(x,y) – seteaza pozitia

import javax.swing.\*;  
import javax.swing.border.Border;  
import java.awt.\*;  
  
public class MyFrame extends JFrame  
{  
 public MyFrame()  
 {  
 System.*out*.println("Frame creat.");  
 }  
 public void start()  
 {  
 //--------------------------------------------------------------------  
 this.setVisible(true);  
 this.setTitle("MyFirstApp");  
 //this.setSize(480,480);  
 this.setDefaultCloseOperation(*EXIT\_ON\_CLOSE*);  
 this.getContentPane().setBackground(new Color(0xFFFFFF));  
 ImageIcon icon = new ImageIcon("C:\\Users\\user\\Desktop\\icon.png");  
 this.setIconImage(icon.getImage());  
 //--------------------------------------------------------------------  
 JLabel label = new JLabel();  
 label.setText("Prima mea aplicatie");  
 ImageIcon image = new ImageIcon("C:\\Users\\user\\Desktop\\test.png");  
 label.setIcon(image);  
 label.setHorizontalTextPosition(JLabel.*CENTER*);  
 label.setVerticalTextPosition(JLabel.*CENTER*);  
 label.setForeground(Color.*GREEN*);  
 label.setFont(new Font("Verdana",Font.*BOLD*,50));  
 label.setIconTextGap(150);  
 label.setBackground(new Color(0x333333));  
 label.setOpaque(true);  
  
 Border border = BorderFactory.*createLineBorder*(Color.*GREEN*,3);  
 label.setBorder(border);  
 label.setVerticalAlignment(JLabel.*CENTER*);  
 label.setHorizontalAlignment(JLabel.*CENTER*);  
 label.setBounds(100,0,480,480);  
 this.setLayout(null); //face ca sa nu fie un strat principal pe toata fereastra. Aceasta permite sa  
 //adaugam pete el mai multe straturi, dar nu cat toata fereastra, ci mai mici si in acelasi timp  
 this.add(label);  
 this.pack();  
 }  
}

**Pannels**

Componenta container ce stocheaza alte obiecte pe ea. Peste panel se pune si Label. Un panel pate contine si alte panele cu JPannels, dar un JLabeles nu poate aranja mai multe Labele

* add(obiect)-poate adauga si labels
* setLayout() – adauga un strat
* setPreferredSize() – seteaza o marime standarta ce poate varia
* toate functiile de la labels
* import javax.swing.\*;  
  import javax.swing.border.Border;  
  import java.awt.\*;  
    
  public class MyFrame extends JFrame  
  {  
   public MyFrame()  
   {  
   System.*out*.println("Frame creat.");  
   }  
   public void start()  
   {  
   JLabel label = new JLabel();  
   label.setText("Test text");  
   label.setFont(new Font("Verdana",Font.*BOLD*,25));  
   ImageIcon icon = new ImageIcon("C:\\Users\\user\\Desktop\\icon.png");  
   label.setIcon(icon);  
   label.setBounds(0,0,100,100);  
   label.setVerticalTextPosition(JLabel.*TOP*);  
    
   JPanel redpanel = new JPanel();  
   JPanel bluepanel = new JPanel();  
   JPanel greenpanel = new JPanel();  
   this.setVisible(true);  
   this.setSize(480,480);  
   this.setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);  
   redpanel.setBackground(Color.*red*);  
   redpanel.setBounds(0,0,200,200);  
   redpanel.setLayout(null);  
   bluepanel.setBackground(Color.*blue*);  
   bluepanel.setBounds(200,0,200,200);  
   bluepanel.setLayout(null);  
   greenpanel.setBackground(Color.*green*);  
   greenpanel.setBounds(0,200,400,200);  
   greenpanel.add(label);  
   greenpanel.setLayout(null);  
   this.add(redpanel);  
   this.add(bluepanel);  
   this.add(greenpanel);  
   this.setLayout(null);  
    
   }  
  }

Buttons

Pentru a atribui un evenimet unui buton, trebuie implementat interfata ActionListener

@Override  
public void actionPerformed(ActionEvent e) {  
   
}

Pentru a verifica ca in metoda data e vorba de butonul creat, spre exemplu de butonul Button:

If(e.getSource()==Button)

Aceasta metoda se executa de fiecare data cand apasam pe un buton

* button.addActionListener(this) – leaga butonul de obiectul current. Anume metoda actionPerformed() de la obiectul this va face actiunea

Mai putem folosi si un fel de lamba system, fara a implementa ceva

button.addActionListener(e -> System.out.println(“”));

* setText(“”);
* setFocusable(false) – face ca rama din jurul butonului sa dispara, adica sa nu mai fie un obiect focusabil
* setIcon()
* setVerticalTextPosition(JButton.CENTER)
* setHorizontalTextPosition(JButton.CENTER) – plaseaza un obiect in raport cu textul
* setFont()
* setIconTextGap() – plaseaza un obiect in raport cu iconul
* setForeground() – schimba culoarea din prim plan
* setBackground() – schimba culoarea butonului
* setBorder(BorderFactory.createLineBorder()) – creaza niste linii in colturi
* setEnable(false) – dezactiveaza butonul
* import javax.swing.\*;  
  import javax.swing.border.Border;  
  import java.awt.\*;  
  import java.awt.event.ActionEvent;  
  import java.awt.event.ActionListener;  
    
  public class MyFrame extends JFrame  
  {  
   private JButton button;  
   public MyFrame()  
   {  
   System.*out*.println("Frame creat.");  
   }  
   public void start()  
   {  
   button = new JButton();  
   button.setBounds(225,230,100,60);  
   button.addActionListener(e -> System.*out*.println("Test"));  
   button.setText("Buton 1");  
   button.setFocusable(false);  
   button.setForeground(Color.*GRAY*);  
   button.setBackground(Color.*cyan*);  
   button.setFont(new Font("Verdana",Font.*BOLD*,12));  
   button.setBorder(BorderFactory.*createLineBorder*(Color.*red*,3));  
   this.setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);  
   this.setSize(500,500);  
   this.setVisible(true);  
   this.setLayout(null);  
   this.add(button);  
   }  
    
  }

**BorderLayout**

Acesta plaseasa orice layer in Nord, Sud, Est, Vest si centru. Orice layer suplimentar e pus in centru. El face ca fiecare layout sa isi modifice marimea impreuna cu fereastra.

Frame.setLayout(new BorderLayout()) – seteaza un borderlayout ca manager de layouturi. In mod normal, **orice frame in mod default foloseste un BorderLayuot.**

Frame.setLayout(new BorderLayout(10,20)) – seteaza un borderlayout ca manager de layouturi si intre layouturile din est si vest vor fi margini de 10 px, iar intre cele din nord si sus de 20 de px

Frame.add(Panel,BorderLayout.NORTH) – adauga un nous trat la managerul de straturi din Frame

Putem avea un BorderLayout si pentru un Panel exact ca la Frame.

**FlowLayout**

Acesta plaseaza componentele intr-o linie. Daca spatial orizontal devine prea mic, chiar si la resize, FlowLayout le pozitioneaza utilizand urmatorul rand.

Frame.setLayout(new FlowLayout()) – adauga un FlowLayout

Frame.setLayout(new FlowLayout(FlowLayout.Center)) – adauga si afiseaza obiectele in centru

Frame.setLayout(new FlowLayout(FlowLayout.Center,10,20)) – adauga si afiseaza obiectele in centru cu distanta orizontala(intre butoane) de 10 si verticala(intre randuri) de 20

**Frame.setVisible(true) trebuie plasat la urma!**

**Panel foloseste in mod default FlowLayout**

**GridLayout**

Pune componentele intr-o grila de cellule. Fiecare componenta ia tot spatial din celula sa, si fiecare celula are aceeasi marime.

Frame.setLayout(new GridLayout()) – adauga un GridLayout  
Frame.setLayout(new GridLayout(x,y)) – adauga un GridLayout cu x linii si y coloane

Frame.setLayout(new GridLayout(x,y,h,v)) – adauga un GridLayout cu x linii si y coloane cu distanta orizontala h intre butoane si distanta v verticala