

```

// Импортиране на класове:
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.applet.*;
// Клас на аплета:
public class KZ00P2 extends JApplet{
    long s1;
    Graphics aGraphics;
    private TextField mNumber1Field = new TextField();
    private Button mCalcButton = new Button();
    private TextField mPField = new TextField();
    private TextField mSField = new TextField();
    private Color mLastBackgroundColor;
    // Лента с менюта:
    private JMenuBar mb;
    // Елементи на менюто:
    private JMenu font,color;
    // Опционални команди за менюто:
    private JCheckBoxMenuItem mItalic,mBold;
    // Команди на менюто - превключватели:
    private JRadioButtonMenuItem mRed,mBlack,mBlue;
    // Команда на менюто:
    private JMenuItem mReset;
    // Етикет:
    private JLabel text;
    // Полета за отметка:
    private JCheckBox italic,bold;
    // Бутон:
    private JButton reset;
    // Превключватели:
    private JRadioButton red,black,blue;
    // Метод за определяне на текста и шрифта за етикета:
    private void setAll(){
        // Цвят:
        Color clr;
        // Текст за етикета:
        String txt=" текст на сив фон";
        // Стил (начална стойност):
        int style=Font.PLAIN;
        // Определяне на стила:
        if(italic.isSelected()){
            style|=Font.ITALIC; // Курсив
        }
        if(bold.isSelected()){
            style|=Font.BOLD; // Удебелен шрифт
        }
        // Уточняване на текста и определяне на цвета:
        if(red.isSelected()){
            txt="Червен"+txt;
            clr=Color.RED;
        }
        else{
            if(black.isSelected()){
                txt="Черен"+txt;
            }
        }
    }
}

```

```

        clr=Color.BLACK;
    }
    else{
        txt="Син"+txt;
        clr=Color.BLUE;
    }
}
// Текст за етикета:
text.setText(txt);
// Шрифт за етикета:
text.setFont(new Font("Arial",style,20));
// Цвят на текста за етикета:
text.setForeground(clr);
}
// Инициализация на аплета:
public void init(){
    // Променливи:
    int w,h;
    // Широчина и височина на аплета:
    w=getWidth();
    h=getHeight();
    // Изключване мениджъра на разположението:
    setLayout(null);
    // Създаване на панел:
    JPanel pnl=new JPanel();
    // Разположение и размери на панела:
    pnl.setBounds(55,5,w-10,h-25);
    // Рамка за панела:
    pnl.setBorder(BorderFactory.createEtchedBorder());
    // Изключване мениджъра на разположението за панела:
    pnl.setLayout(null);
    // Създаване на етикет:
    text=new JLabel();
    // Режим на подравняване на текста по центъра:
    text.setHorizontalAlignment(JLabel.CENTER);
    // Разположение и размери на етикета:
    text.setBounds(5,5,pnl.getWidth()-10,pnl.getHeight()/3);
    // Рамка около етикета:
    text.setBorder(BorderFactory.createEtchedBorder());
    // Добавяне на етикета към панела:
    pnl.add(text);
    // Създаване на полета за отметка:
    italic=new JCheckBox("Курсивен шрифт");
    bold=new JCheckBox("Удебелен шрифт");
    // Създаване на бутон:
    reset=new JButton("Изчисти");
    // Разположение и размери на полетата за отметка
    // и бутона:
    italic.setBounds(text.getX(),text.getY()+text.getHeight()+5,text.getWidth()/2-
5,30);

    bold.setBounds(italic.getX(),italic.getY()+italic.getHeight()+5,italic.getWidth(),ita
lic.getHeight());

```

```

reset.setBounds(bold.getX(), bold.getY()+bold.getHeight()+5, bold.getWidth(), bold.getHe
ight());
// Добавяне на полетата за отметка и бутона в панела:
pn1.add(italic);
pn1.add(bold);
pn1.add(reset);
// Създаване на група превключватели:
ButtonGroup bg=new ButtonGroup();
// Създаване на превключватели:
red=new JRadioButton("Червен", true);
black=new JRadioButton("Черен", false);
blue=new JRadioButton("Син", false);
// Добавяне на превключвателите в групата:
bg.add(red);
bg.add(black);
bg.add(blue);
// Разположение и размери на превключвателите:

red.setBounds(italic.getX()+italic.getWidth()+5, italic.getY(), italic.getWidth(), itali
c.getHeight());
black.setBounds(red.getX(), bold.getY(), red.getWidth(), red.getHeight());
blue.setBounds(black.getX(), reset.getY(), black.getWidth(), black.getHeight());
// Добавяне на превключвателите в панела:
pn1.add(red);
pn1.add(black);
pn1.add(blue);
// Създаване на лента с менюта:
mb=new JMenuBar();
// Създаване на елемент на менюто:
font=new JMenu("Шрифт");
// Създаване на опционални команди за менюто:
mItalic=new JCheckBoxMenuItem("Курсив", false);
mBold=new JCheckBoxMenuItem("Удебелен", false);
// Създаване на команда за менюто:
mReset=new JMenuItem("Изчисти");
// Добавяне на опционалните команди в менюто:
font.add(mItalic);
font.add(mBold);
// Добавяне на разделител:
font.addSeparator();
// Добавяне на команда в менюто:
font.add(mReset);
// Създаване на елемент на менюто:
color=new JMenu("Цвят");
// Създаване на група превключватели:
ButtonGroup mBG=new ButtonGroup();
// Създаване на команди превключватели:
mRed=new JRadioButtonMenuItem("Червен", true);
mBlack=new JRadioButtonMenuItem("Черен", false);
mBlue=new JRadioButtonMenuItem("Син", false);
// Добавяне на командите превключватели в групата:
mBG.add(mRed);
mBG.add(mBlack);
mBG.add(mBlue);

```

```

// Добавяне на командите превключватели в менюто:
color.add(mRed);
color.add(mBlack);
color.add(mBlue);
// Добавяне на елементите в меню лентата:
mb.add(font);
mb.add(color);
// Добавяне на лентата в аплета:
setJMenuBar(mb);
// Прилагане на текст и шрифт към етикета:
setAll();
// Добавяне на панела в аплета:
add pnl);
// Регистрация на обработчици за събитията
// на класа ActionEvent:
mItalic.addActionListener(e->italic.setSelected(mItalic.isSelected()));
mBold.addActionListener(e->bold.setSelected(mBold.isSelected()));
italic.addActionListener(e->mItalic.setSelected(italic.isSelected()));
bold.addActionListener(e->mBold.setSelected(bold.isSelected()));
mRed.addActionListener(e->red.setSelected(mRed.isSelected()));
mBlack.addActionListener(e->black.setSelected(mBlack.isSelected()));
mBlue.addActionListener(e->blue.setSelected(mBlue.isSelected()));
red.addActionListener(e->mRed.setSelected(red.isSelected()));
black.addActionListener(e->mBlack.setSelected(black.isSelected()));
blue.addActionListener(e->mBlue.setSelected(blue.isSelected()));
// Регистрация на обработчици за събитията
// на класа ItemEvent:
italic.addItemListener(e->setAll());
bold.addItemListener(italic.getItemListeners()[0]);
red.addItemListener(italic.getItemListeners()[0]);
black.addItemListener(italic.getItemListeners()[0]);
blue.addItemListener(italic.getItemListeners()[0]);
// Обработчик за бутона:
reset.addActionListener(e->{
    italic.setSelected(true);
    // Програмно "кликване" в поле за отметка:
    italic.doClick();
    bold.setSelected(true);
    // Програмно "кликване" в поле за отметка:
    bold.doClick();
    // Програмно "кликване" на превключвател:
    red.doClick();
});
// Обработчик за команда в менюто:
mReset.addActionListener(e->reset.doClick()); this.setBackground(Color.black);

// Set layout manager to null
this.setLayout(null);

// Create the first text field
mNumber1Field.setBounds(new Rectangle(20, 50, 60, 25));
mNumber1Field.setBackground(Color.white);
this.add(mNumber1Field, null);
mNumber1Field.setText("10");

```

```

// Create the "calculate sum" button
mCalcButton.setBounds(new Rectangle(170, 50, 90, 25));
mCalcButton.addActionListener(
    new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            calcSum();
        }
    });
mCalcButton.setLabel("Calculate");
this.add(mCalcButton, null);

// Create the result text field1
mPField.setEditable(false);
mPField.setBackground(Color.gray);
mPField.setForeground(Color.white);
mPField.setBounds(new Rectangle(20, 85, 240, 25));
this.add(mPField, null);
// Create the result text field2
mSField.setEditable(false);
mSField.setBackground(Color.gray);
mSField.setForeground(Color.white);
mSField.setBounds(new Rectangle(280, 85, 240, 25));
this.add(mSField, null);
s1 = new Long(mNumber1Field.
    getText()).longValue();
}

```

```

public boolean mouseDown(Event aEvent, int aX, int aY) {
    mLastBackgroundColor = this.getBackground();
    this.setBackground(Color.red);
    return true;
}

```

```

public boolean mouseUp(Event aEvent, int aX, int aY) {
    this.setBackground(mLastBackgroundColor);
    return true;
}

```

```

public void paint(Graphics aGraphics) {
    super.paint(aGraphics);
    Font font = new Font("Dialog", Font.BOLD, 23);
    aGraphics.setFont(font);
    // aGraphics.setColor(Color.gray);
    // aGraphics.drawString("Applet", 20, 32);
    // aGraphics.setColor(Color.white);
    // aGraphics.drawString("Applet", 18, 30);
    aGraphics.setColor(Color.green);
    s1 = new Long(mNumber1Field.

```

```

        getText()).longValue();
        int a=(int)s1;
        aGraphics.drawRect(10,150,5*a,5*a);
        aGraphics.drawRect(10,150,10*a,10*a);
    }

    private void calcSum() {
        try {
            s1 = new Long(mNumber1Field.
                getText()).longValue();

            long s=s1*s1;
            long p=4*s1;
            mSField.setText(" S= " + s);
            mPField.setText(" P= " + p);
        } catch (Exception ex) {
            mSField.setText("Error!");
            mPField.setText("Error!");
        }

        repaint(); }
    public static void main(String[] aArgs) {
        Frame frame = new Frame("Sumator");
        frame.setSize(400,200);
        KZ00P2 applet = new KZ00P2();
        applet.init();
        frame.add(applet);
        frame.setVisible(true);
        applet.start();
    }
}

```