

Текстови компоненти

Клас JTextField (javax.swing) – текстово поле

JTextField се използва за текст от един ред. При натискане на **<Enter>** текстовите контроли вдигат събитието **ActionEvent**, изпраща се съобщението **actionPerformed** на интерфейса **ActionListener** за всички слушатели, които са се регистрирали чрез метода **addActionListener**.

```
public JTextField()
public JTextField(String text)
public JTextField(int columns)
public JTextField(String text, int columns)
public JTextField(Document doc, String text, int columns)
```

text – начален текст (по подразбиране **null**);
columns – брой колони (по подразбиране **0**);
doc – определен модел за запазване на текста (по подразбиране **null**).

```
public void addActionListener(ActionListener l)
```

Добавя слушател l към текстовото поле, за да регистрира събитието ActionEvent.

```
public String getText()
```

Връща текста, който се съдържа в текстовото поле.

```
public void setText(String t)
```

Установява текста t в текстовото поле.

Клас JTextArea (javax.swing) – текстова област

JTextArea се използва за текст от много редове. За да използва плъзгачи, **JTextArea** се разполага в компонента **JScrollPane**. Промяна на текста се предава от модела за запазване на текста чрез събитието **DocumentEvent**, изпращат се съобщенията **changeUpdate**, **insertUpdate**, **removeUpdate** на интерфейса **DocumentListener** за всички слушатели, които са се

```
public JTextArea()
public JTextArea(String text)
public JTextArea(int rows, int columns)
public JTextArea(String text, int rows, int columns)
public JTextArea(Document doc)
```

text – начален текст (по подразбиране **null**);
rows – брой редове (по подразбиране **null**);
columns – брой колони (по подразбиране **0**);
doc – определен модел за запазване на текста (по подразбиране **null**).

```
public void append(String str)
```

Добавя даден текст str към края на документа.

Клас JScrollPane (javax.swing)

Представя плъзгач за компонента.

```
public JScrollPane(Component view, int vsbPolicy, int hsbPolicy)
```

view – компонентата, която ще използва плъзгач;

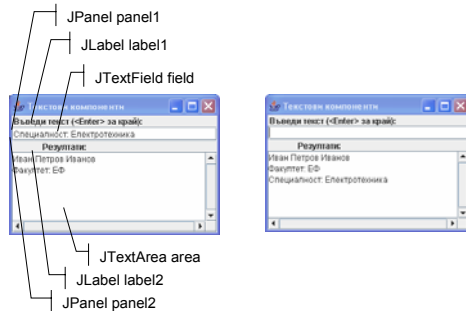
vsbPolicy – определя вида на вертикалния плъзгач:

```
VERTICAL_SCROLLBAR_ALWAYS
VERTICAL_SCROLLBAR_AS_NEEDED
VERTICAL_SCROLLBAR_NEVER
```

hsbPolicy – определя вида на хоризонталния плъзгач:

```
HORIZONTAL_SCROLLBAR_ALWAYS
HORIZONTAL_SCROLLBAR_AS_NEEDED
HORIZONTAL_SCROLLBAR_NEVER
```

Пример: Приложение, което въвежда текст в текстово поле и при натискане на **<Enter>** въведеният текст се добавя в текстова област. Използва събитието **ActionEvent** за текстовото поле.



```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

public class TextDemo extends JFrame implements ActionListener {
    private JPanel panel1;
    private JLabel label1;
    private JTextField field;
    private JPanel panel2;
    private JLabel label2;
    private JTextArea area;

    public TextDemo () {
        super("Текстови компоненти");
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        Container container = this.getContentPane();

        panel1 = new JPanel(); // панел с етикет и текстово поле
        panel1.setLayout(new BorderLayout(panel1, BorderLayout.Y_AXIS));
        label1 = new JLabel("Въведи текст (<Enter> за край);");
        panel1.add(label1);
```

```
        field = new JTextField(10);
        field.addActionListener(this);
        panel1.add(field);

        container.add(panel1, BorderLayout.PAGE_START);

        panel2 = new JPanel(); // панел с етикет и текстова област
        panel2.setLayout(new BorderLayout(panel2, BorderLayout.Y_AXIS));
        label2 = new JLabel("Резултати:");
        panel2.add(label2);

        area = new JTextArea(5, 10);
        JScrollPane scroll = new JScrollPane(area,
            JScrollPane.VERTICAL_SCROLLBAR_ALWAYS,
            JScrollPane.HORIZONTAL_SCROLLBAR_ALWAYS);
        panel2.add(scroll);

        container.add(panel2, BorderLayout.CENTER);
        this.setSize(300, 200);
        this.setVisible(true);
    }
}
```

Фреймът е слушател за събитието `ActionEvent`, което се вдига при натискане на `<Enter>` във `field`.

```
// Извиква се автоматично при натискане на <Enter> във field.
public void actionPerformed(ActionEvent e) {
    // Взима въведения текст във field
    String input = field.getText();
    // и го добавя в area.
    area.append(input + "\n");
    // Изчиства field.
    field.setText("");
}


public static void main(String[] args) {
    SwingUtilities.invokeLater(new Runnable() {
        public void run() {
            new TextDemo();
        }
    });
}
```

Пример: Клас КОМПЮТЪР

- полета:
 - име на модел;
 - цена;
 - доказана ефективност в точки;
- методи:
 - намиране на по-добрия компютър (с високо съотношение точки към цена);
 - символно представяне на класа – информация за отношението точки към цена.

Аплет с бутони:

- Добави – добавя данните за текущия компютър;
- Най-добър компютър – изобразява информация за най-добрия за купуване компютър (с максимално отношение точки към цена).

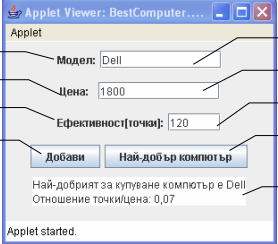


```
import java.text.DecimalFormat;

public class Computer {
    private String model; // Компютър // модел
    private double price; // цена
    private int score; // ефективност в точки
    public Computer (String model, double price, int score) { // конструктор
        this.model = model;
        this.price = price;
        this.score = score;
    }

    public boolean isBetterThan (Computer nextModel) { // метод, определящ
        if (score/price > nextModel.score/nextModel.price) // по-добрия
            return true; // компютър
        else
            return false;
    }

    public String toString () { // символно
        DecimalFormat fmt = new DecimalFormat("0.##"); // представяне
        return "Най-добрият за купуване компютър е " + model + "\n\r" +
            "Отношение точки/цена: " + fmt.format(score/price);
    }
}
```



Labels on the left: `JLabel modelLabel`, `JLabel priceLabel`, `JLabel scoreLabel`, `JButton addButton`

Labels on the right: `JTextField model`, `JTextField price`, `JTextField score`, `JButton bestComputerButton`, `JTextArea display`

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

public class BestComputer extends JApplet implements ActionListener {
    private JTextField model = new JTextField(12);
    private JTextField price = new JTextField(12);
    private JTextField score = new JTextField(5);
    private JLabel modelLabel = new JLabel("Модел:");
    private JLabel priceLabel = new JLabel("Цена: ");
    private JLabel scoreLabel = new JLabel("Ефективност[точки]:");
    private JButton addButton = new JButton("Добави");
    private JButton bestComputerButton = new JButton("Най-добър компютър");
    private JTextArea display = new JTextArea();
    private JPanel p1 = new JPanel();
    private JPanel p2 = new JPanel();
    private JPanel p3 = new JPanel();
    private JPanel p4 = new JPanel();
    private Computer[] computers = new Computer[100];
    private static int count = 0;
}
```

```
public void init() {
    setSize(300,200);
    Container container = getContentPane();
    container.setLayout(new FlowLayout());
    p1.add(modelLabel);
    p1.add(model);
    container.add(p1);
    p2.add(priceLabel);
    p2.add(price);
    container.add(p2);
    p3.add(scoreLabel);
    p3.add(score);
    container.add(p3);
    p4.add(addButton);
    p4.add(bestComputerButton);
    container.add(p4);
    container.add(display);
    addButton.addActionListener(this);
    bestComputerButton.addActionListener(this);
}
```

```
public void actionPerformed(ActionEvent e) {
    if (e.getSource() == addButton) { // бутон Добави вдига събитието
        Computer c = new Computer(model.getText(),
            Double.parseDouble(price.getText()),
            Integer.parseInt(score.getText()));

        computers[count] = c;
        count++;
    }
    else if (e.getSource() == bestComputerButton) { // бутон Най-добър
        Computer best = computers[0]; // компютър вдига
        for (int i = 1; i < count; i++) // събитието
            if (computers[i].isBetterThan(best))
                best = computers[i];
        display.setText(best.toString());
    }
}
}
```

Компоненти за избор

Събития с избор на елементи

1. Клас **ItemEvent** – показва дали елементът е селектиран или деселектиран

2. Интерфейс **ItemListener**

```
public interface ItemListener {
    // Извиква се при селектиране или деселектиране на елемент.
    void itemStateChanged(ItemEvent e);
}
```

3. Добавяне на слушател **I** за събитието **ItemEvent** към компонента за избор

```
public void addItemListener(ItemListener l)
```

1. Бутон за избор

Клас **JCheckBox** (**javax.swing**)

Бутон за избор, който може да бъде селектиран (състояние „включено“ (**true**)) или деселектиран („изключено“ (**false**)). В една група могат да бъдат селектирани произволен брой такива компоненти.

```
public JCheckBox(String text)
public JCheckBox(String text, boolean selected)
text – текст на бутона за избор (по подразбиране null);
selected – състояние на бутона за избор (по подразбиране false).
```

```
public boolean isSelected()
```

Връща състоянието на бутона (**true/false**).

```
public void setSelected(boolean b)
```

Установява състоянието на бутона.

Събития

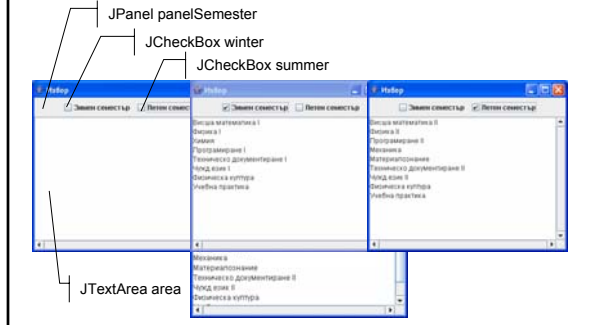
1. Събития с действие

- клас **ActionEvent**
- интерфейс **ActionListener**
 - метод **actionPerformed**
- регистриране на компонентата за събитието **ActionEvent** чрез метода **addActionListener**

2. Събития с избор на елемент

- клас **ItemEvent**
- интерфейс **ItemListener**
 - метод **itemStateChanged**
- регистриране на компонентата за събитието **ItemEvent** чрез метода **addItemListener**

Пример: Приложение, в което чрез бутони за избор се изобразяват учебните дисциплини, изучавани в зимен или летен семестър. Използва събитието `ItemEvent` за `JCheckBox`.



```
import javax.swing.*;
import java.awt.event.*;
import java.awt.*;

public class Choice extends JFrame implements ItemListener{
    private JPanel panelSemester;

    private JCheckBox winter;
    private JCheckBox summer;

    private JTextArea area;

    private String[] coursesWinter = {"Висша математика I", "Физика I", "Химия",
    "Програмиране I", "Техническо документиране I",
    "Чужд език I", "Физическа култура", "Учебна практика"};

    private String[] coursesSummer = {"Висша математика II", "Физика II",
    "Програмиране II", "Механика", "Материалознание",
    "Техническо документиране II", "Чужд език II", "Физическа култура",
    "Учебна практика"};

    public Choice() {
        super("Избор");
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        Container container = this.getContentPane();
```

```
// Панел с два бутона за избор
panelSemester = new JPanel();

winter = new JCheckBox("Зимен семестър");
winter.addItemListener(this);
summer = new JCheckBox("Летен семестър");
summer.addItemListener(this);
panelSemester.add(winter);
panelSemester.add(summer);

container.add(panelSemester, BorderLayout.PAGE_START);

// Текстова област с плъзгачи
area = new JTextArea(5, 10);
JScrollPane scroll = new JScrollPane(area,
    JScrollPane.VERTICAL_SCROLLBAR_ALWAYS,
    JScrollPane.HORIZONTAL_SCROLLBAR_ALWAYS);
container.add(scroll, BorderLayout.CENTER);

this.setSize(350, 300);
this.setVisible(true);
}
```

```
// Извиква се автоматично при промяна на състоянието на бутон за избор
public void itemStateChanged(ItemEvent e) {
    area.setText(""); // Изчиства текстовата област

    if(winter.isSelected()) { // Избран е Зимен семестър
        for (int i = 0; i < coursesWinter.length; i++)
            area.append(coursesWinter[i] + "\n");
        if (summer.isSelected()) { // и Летен семестър
            for (int i = 0; i < coursesSummer.length; i++)
                area.append(coursesSummer[i] + "\n");
        }
    }
    else if (summer.isSelected()) { // Избран е Летен семестър
        for (int i = 0; i < coursesSummer.length; i++)
            area.append(coursesSummer[i] + "\n");
    }
}
```

```
public static void main(String[] args) {
    SwingUtilities.invokeLater(new Runnable() {
        public void run() {
            new Choice();
        }
    });
}
```

2. Радио бутон

Клас `JRadioButton` (`javax.swing`)

Радио бутон, които може да бъде селектиран (състояние „включено“ (`true`)) или деселектиран („изключено“ (`false`)). Радио бутоните се добавят в група, създадена чрез `ButtonGroup`, в която само един радио бутон може да се селектира. Групата се поставя в панел и чрез метода му `setBorder` се установява рамка около радио бутоните.

```
public JRadioButton(String text)
public JRadioButton(String text, boolean selected)
text – текст на радио бутона (по подразбиране null);
selected – състояние на радио бутона (по подразбиране false).
```

`public void setSelected(boolean b)`

Установява състоянието на бутона.

Събития

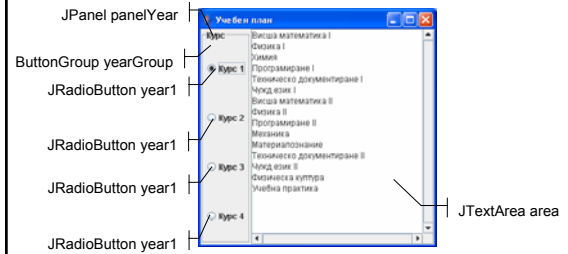
1. Събития с действие

- клас **ActionEvent**
- интерфейс **ActionListener**
 - метод **actionPerformed**
- регистриране на компонентата за събитието **ActionEvent** чрез метода **addActionListener**

2. Събития с избор на елемент

- клас **ItemEvent**
- интерфейс **ItemListener**
 - метод **itemStateChanged**
- регистриране на компонентата за събитието **ItemEvent** чрез метода **addItemListener**

Пример: Приложение, което изобразява учебните дисциплини, които се изучават в различните курсове при избор чрез радио бутон. Използва събитието **ActionEvent** за **JRadioButton**.



```
import javax.swing.*;
import javax.swing.border.*;
import java.awt.event.*;
import java.awt.*;

public class RadioButtons extends JFrame implements ActionListener {
    private JPanel panelYear;
    private ButtonGroup yearGroup;
    private JRadioButton year1;
    private JRadioButton year2;
    private JRadioButton year3;
    private JRadioButton year4;
    private JTextArea area;

    private String[] coursesYear1 = {"Висша математика I", "Физика I",
    "Химия", "Програмиране I", "Техническо документиране I",
    "Чужд език I", "Висша математика II", "Физика II", "Програмиране II",
    "Механика", "Материалознание", "Техническо документиране II",
    "Чужд език II", "Физическа култура", "Учебна практика"};

```

```
private String[] coursesYear2 = {"Висша математика III",
    "Теоретична електротехника I", "Машинни елементи и механизми",
    "Икономика", "Техническа безопасност", "Чужд език III",
    "Теоретична електротехника II", "Електрически измервания",
    "Електроника", "Цифрова и микропроцесорна техника",
    "Електротехнически материали", "Физическа култура"};
private String[] coursesYear3 = {"Електрически машини I",
    "Електрически апарати I", "Техника на високите напрежения",
    "Телекомуникации", "Осветителна и инсталационна техника",
    "Енергийна техника", "Електрически машини II",
    "Електрически апарати II", "Преобразователна техника",
    "Проектиране на електрически машини", "Мениджмънт"};
private String[] coursesYear4 = {"Електрозадвижване",
    "Електроенергетика", "Комутационна техника",
    "Технологии в електротехниката и електрониката",
    "Екология и възобновяеми източници на енергия",
    "Електрически машини с електронно управление",
    "Числено моделиране на процеси и полета",
    "Електрически микромашини",
    "Изпитване и андеждност на електрически машини",
    "Свободно избираема дисциплина",
    "Свободно избираема дисциплина"};

```

```
public RadioButtons() {
    super("Учебен план");
    this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    Container container = this.getContentPane();

    yearGroup = new ButtonGroup();
    year1 = new JRadioButton("Курс 1");
    year1.setActionCommand("Курс 1");
    year1.setSelected(false);
    year1.addActionListener(this);
    year2 = new JRadioButton("Курс 2");
    year2.setActionCommand("Курс 2");
    year2.setSelected(false);
    year2.addActionListener(this);
    year3 = new JRadioButton("Курс 3");
    year3.setActionCommand("Курс 3");
    year3.setSelected(false);
    year3.addActionListener(this);
    year4 = new JRadioButton("Курс 4");
    year4.setActionCommand("Курс 4");
    year4.setSelected(false);
    year4.addActionListener(this);

```

```
yearGroup.add(year1);
yearGroup.add(year2);
yearGroup.add(year3);
yearGroup.add(year4);

panelYear = new JPanel(new GridLayout(0, 1));
panelYear.add(year1);
panelYear.add(year2);
panelYear.add(year3);
panelYear.add(year4);
panelYear.setBorder(new TitledBorder(new EtchedBorder(), "Курс"));
container.add(panelYear, BorderLayout.LINE_START);
area = new JTextArea(5, 10);
JScrollPane scroll = new JScrollPane(area,
    JScrollPane.VERTICAL_SCROLLBAR_ALWAYS,
    JScrollPane.HORIZONTAL_SCROLLBAR_ALWAYS);
container.add(scroll, BorderLayout.CENTER);
this.setSize(350, 350);
this.setVisible(true);
}

```

```
// Добавя дисциплини в текстовата област
private void appendCourses (String[] courses) {
    for (int i = 0; i < courses.length; i++)
        area.append(courses[i] + "\n");
}

// Извиква се автоматично при натискане на радио бутон
public void actionPerformed(ActionEvent e) {
    String command = e.getActionCommand();
    area.setText("");
    if(command.equals("Курс 1") && year1.isSelected())
        appendCourses(coursesYear1);
    else if (command.equals("Курс 2") && year2.isSelected())
        appendCourses(coursesYear2);
    else if (command.equals("Курс 3") && year3.isSelected())
        appendCourses(coursesYear3);
    else if (command.equals("Курс 4") && year4.isSelected())
        appendCourses(coursesYear4);
}
}
```

```
public static void main(String[] args) {
    SwingUtilities.invokeLater(new Runnable() {
        public void run() {
            new RadioButtons();
        }
    });
}
```

3. Комбиниран бутон за избор
Клас JComboBox (javax.swing)
Компонента, която комбинира бутон или поле с възможност за редакция и падащ списък. Позволява избор от списък или въвеждане на стойност в поле.
 public JComboBox()
Създава комбиниран бутон за избор.
 public void addItem(Object anObject)
Добавя елемент anObject към списъка с елементи на комбинирания бутон.
 public Object getSelectedItem()
Връща текущия избран елемент.
 public void setSelectedIndex(int anIndex)
Установява елементът с индекс anIndex.
 public void removeAllItems()
Премахва всички елементи от списъка.

Събития
1. Събития с действие
 – клас **ActionEvent**
 – интерфейс **ActionListener**
 – метод **actionPerformed**
 – регистриране на компонентата за събитието **ActionEvent** чрез метода **addActionListener**
2. Събития с избор на елемент
 – клас **ItemEvent**
 – интерфейс **ItemListener**
 – метод **itemStateChanged**
 – регистриране на компонентата за събитието **ItemEvent** чрез метода **addItemListener**

Пример: Приложение, което въвежда данни за имената на студент чрез текстови полета и факултет чрез комбиниран бутон за избор. Бутонът **Добави** добавя данните за студента в текстовата област, а **Изчисти** – изчиства компонентите. Използва събитието **ActionEvent** за **JButton**.

```
import javax.swing.*;
import java.awt.event.*;
import java.awt.*;

public class ComboBoxDemo extends JFrame implements ActionListener {
    private JPanel panelNames;
    private JTextField[] names;
    private JComboBox faculty;
    private String[] labels = {"Собствено име:", "Бащино име:", "Фамилно име:", "Факултет"};

    private int n;
    private JPanel panelButtons;
    private JButton addButton;
    private JButton clearButton;
    private JTextArea area;

    public ComboBoxDemo() {
        super("Записване на студенти");
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        Container container = this.getContentPane();
    }
}
```

```

n = labels.length;
names = new JTextField[n-1];
panelNames = new JPanel(new GridLayout(4,3));
for (int i = 0; i < n - 1; i++) {
    JLabel l = new JLabel(labels[i], JLabel.TRAILING);
    panelNames.add(l);
    names[i] = new JTextField(10);
    panelNames.add(names[i]);
}
JLabel l = new JLabel(labels[3], JLabel.TRAILING);
panelNames.add(l);
faculty = new JComboBox();
panelNames.add(faculty);
container.add(panelNames, BorderLayout.PAGE_START);
area = new JTextArea(5, 10);
JScrollPane scroll = new JScrollPane(area,
    JScrollPane.VERTICAL_SCROLLBAR_ALWAYS,
    JScrollPane.HORIZONTAL_SCROLLBAR_ALWAYS);
container.add(scroll, BorderLayout.CENTER);

```

```

panelButtons = new JPanel();
addButton = new JButton("Добави");
panelButtons.add(addButton);
addButton.addActionListener(this);
clearButton = new JButton("Изчисти");
panelButtons.add(clearButton);
clearButton.addActionListener(this);
container.add(panelButtons, BorderLayout.PAGE_END);
reset();
this.setSize(350, 300);
this.setVisible(true);
}

```

```

private void reset() { // Установява началното състояние
    for (int i = 0; i < n-1; i++) // на компонентите
        names[i].setText("");
    faculty.removeAllItems();
    faculty.addItem("ФФ");
    faculty.addItem("ЕФ");
    faculty.addItem("ФЕТТ");
    faculty.addItem("ФКТТ");
    faculty.addItem("ФКСУ");
    area.setText("");
}
public void actionPerformed(ActionEvent e) {
    if (e.getSource() == addButton) {
        area.append("Студент: ");
        for (int i = 0; i < n - 1; i++)
            area.append(names[i].getText() + " ");
        area.append("\nФакултет: " + faculty.getSelectedItem());
    }
    else if (e.getSource() == clearButton)
        reset();
}

```

```

public static void main(String[] args) {
    SwingUtilities.invokeLater(new Runnable() {
        public void run() {
            new ComboBoxDemo();
        }
    });
}
}

```