



Списъчно поле jComboBox



Общи сведения

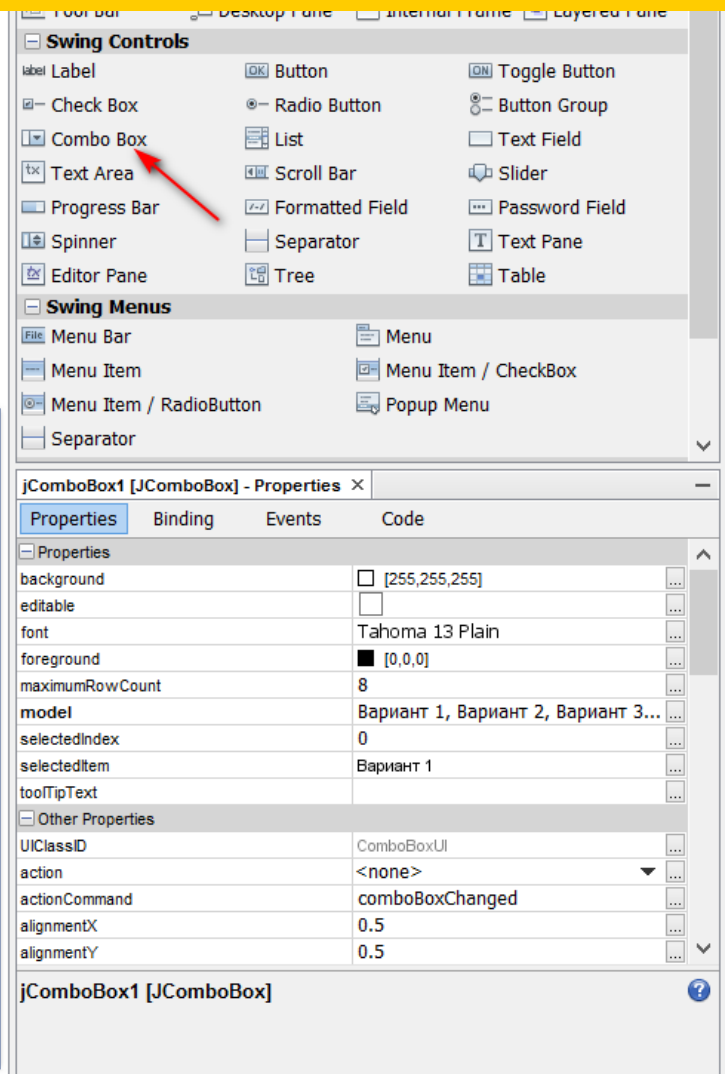
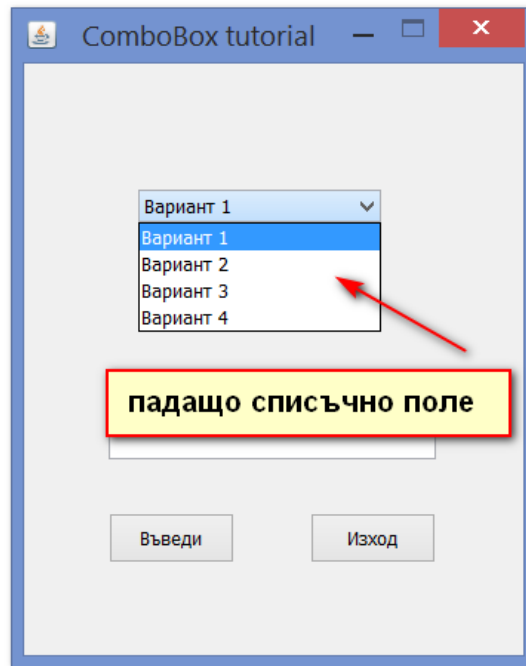
- ❗ Падащото списъчно поле `jComboBox` представлява обект, който съчетава бутон или редактируемо поле и падащ списък.
- ❗ Потребителят може да избере стойност от падащия списък, който се появява по искане на потребителя. Можем да създадем редактируемо падащо списъчно поле, в което има поле за редактиране, където потребителят може да въведе стойност.





Създаване

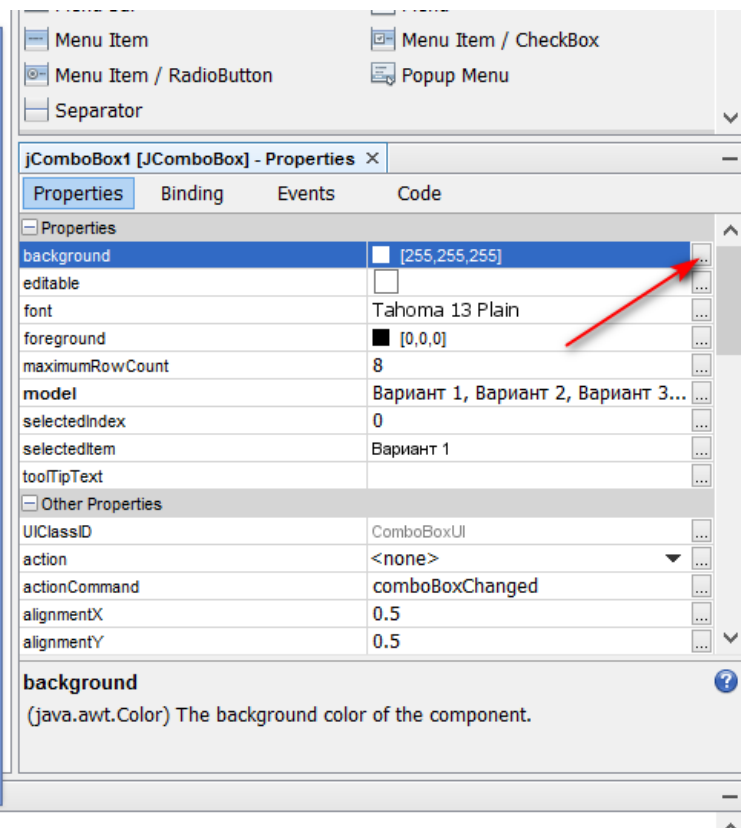
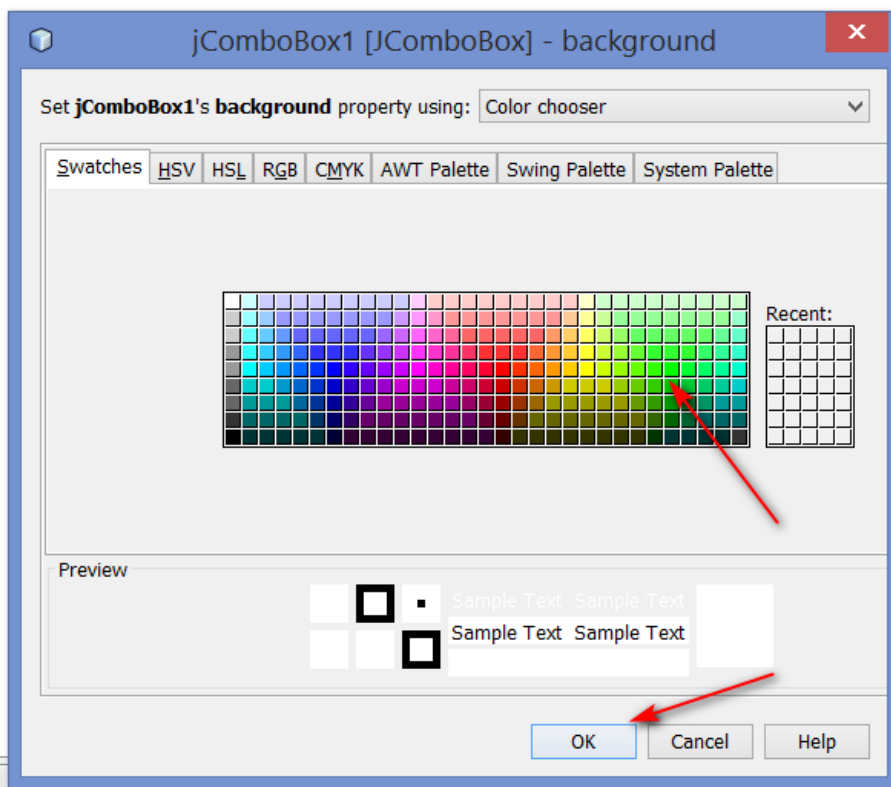
SOCIETY
ROBOTIC





background

🔧 Задаване на цвят на фона на **jComboBox**;



editable

- При състояние **True** , падащото списъчно поле става редактируемо;

The image shows a WPF application window titled "ComboBox tutorial" with a ComboBox containing the text "Нов елемент". A red arrow points to the text, and a yellow box with a red border contains the text "полето е редактируемо". To the right, the Visual Studio Properties window for the ComboBox is open, showing the "editable" property checked. A red arrow points to the checkmark. Below the properties window, the definition for the "editable" property is shown: "(boolean) If true, the user can type a new value in the combo box."

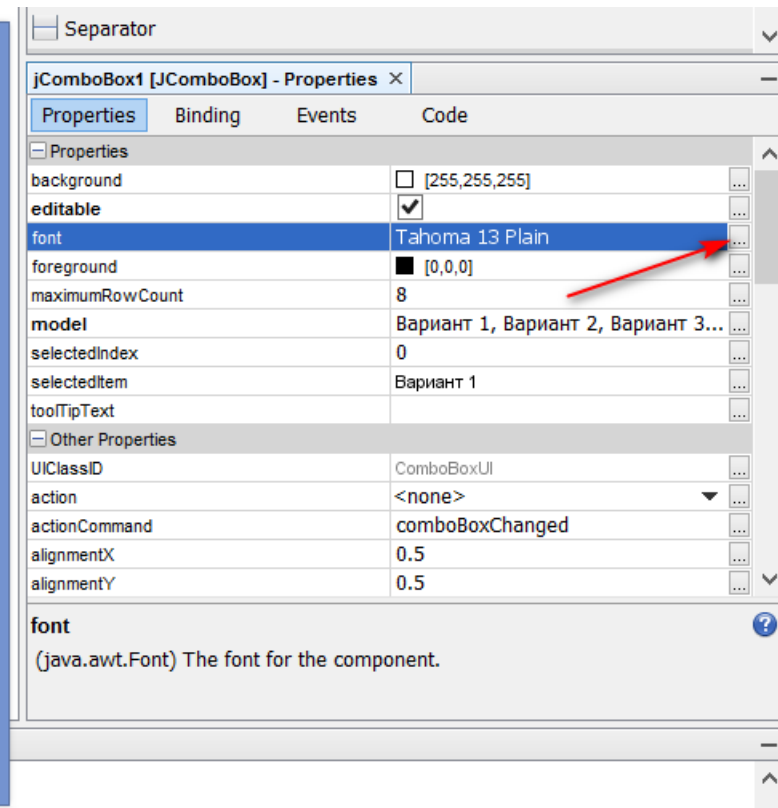
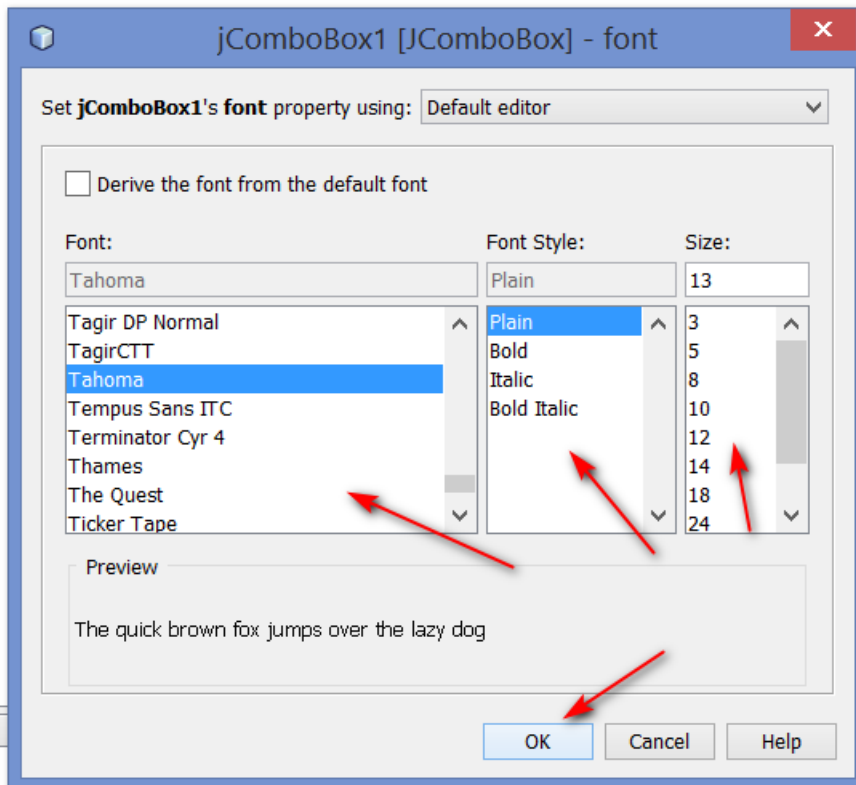
Property	Value
background	[255,255,255]
editable	<input checked="" type="checkbox"/>
font	Tahoma 13 Plain
foreground	[0,0,0]
maximumRowCount	8
model	Вариант 1, Вариант 2, Вариант 3...
selectedIndex	0
selectedItem	Вариант 1
toolTipText	
UIClassID	ComboBoxUI
action	<none>
actionCommand	comboBoxChanged
alignmentX	0.5
alignmentY	0.5

editable
(boolean) If true, the user can type a new value in the combo box.



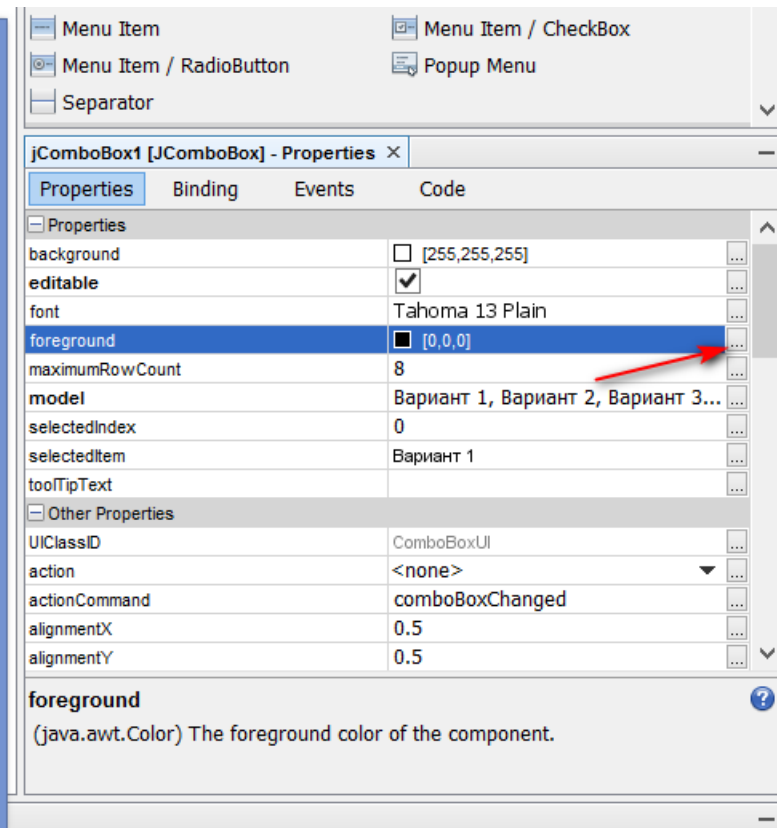
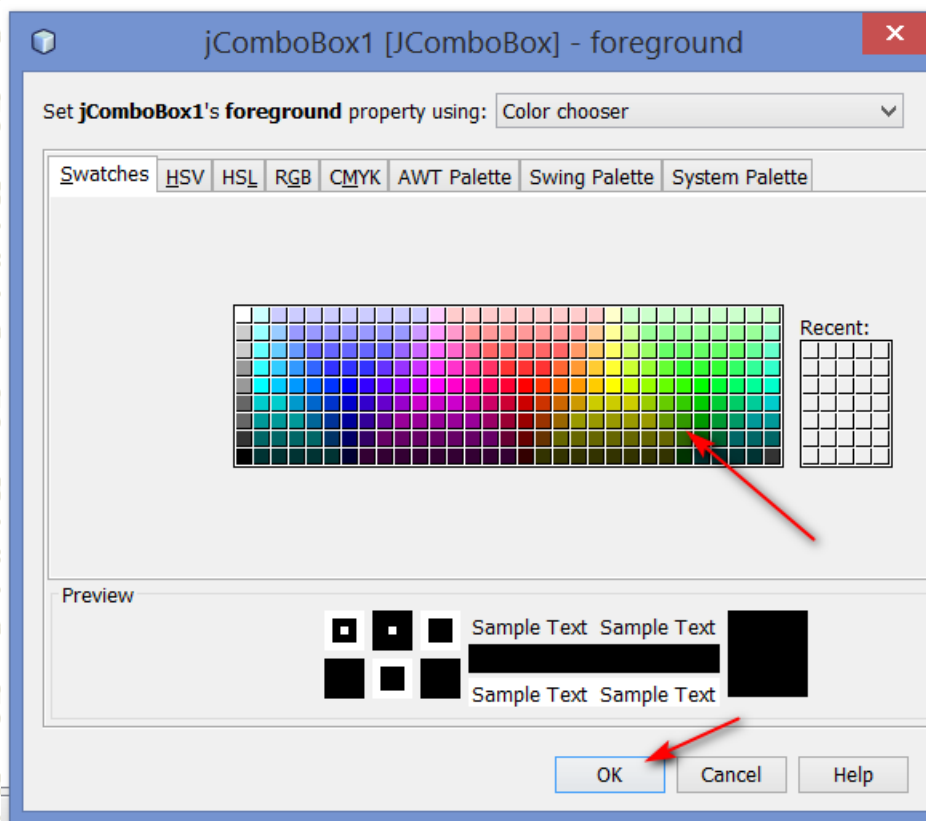
Font

⚙️ Настройки на шрифта;



foreground

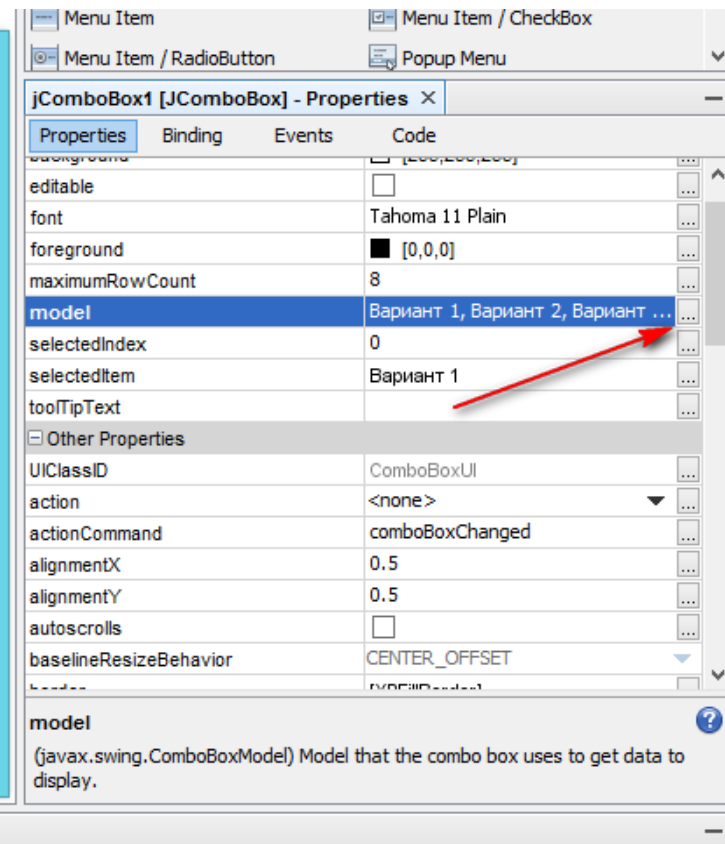
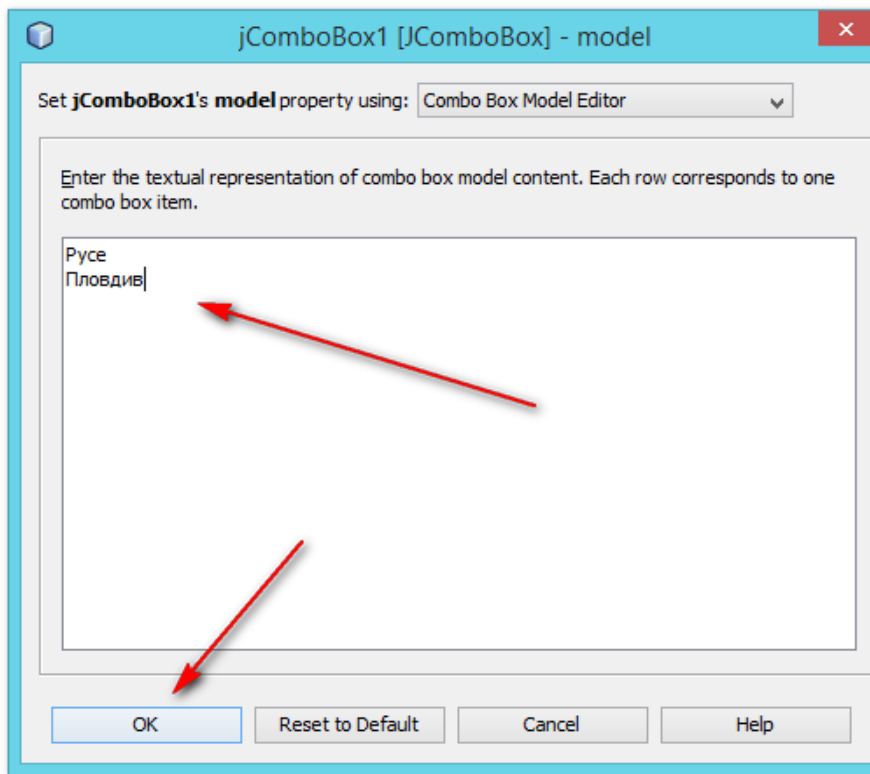
Задава цвят на текста в **jComboBox**;





model

⚡ Задава елементите от списъка на **jComboBox**;



SOCIETY
ROBOTIC





Добавяне на нов елемент с код при стартиране

Използваме метода `addItem` за да добавим елемент. Поставяме кода във функцията за създаване на формата.

```
/**  
 * Creates new form combobox_tutor  
 */  
public combobox_tutor() {  
    initComponents();  
    jComboBox1.addItem("Варна");  
    jComboBox1.addItem("София");  
}
```





Индекс на избран елемент

- Индексът на избрания елемент получаваме като използваме метода `getSelectedIndex()`.

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
    jComboBox1.addItem(jTextField1.getText ());  
    jComboBox1.getSelectedIndex ();  
}
```

- Резултатът е число тип `int`, което показва поредния номер на избрания елемент;





Получаване на елемент

Използваме метода `getSelectedItem()`, за да получим текста на избрания елемент.

```
jComboBox1.getSelectedItem().toString()
```

Използваме метода `toString()`, за да преобразуваме в стринг избрания елемент





Пример

```
public class combobox_tutor extends javax.swing.JFrame {
```

```
    /**  
     * Creates new form combobox_tutor  
     */
```

```
    public combobox_tutor() {  
        initComponents();  
        jComboBox1.addItem("Варна");  
        jComboBox1.addItem("София");  
    }
```

```
    /**  
     * This method is called from within the constructor to initialize the form.  
     * WARNING: Do NOT modify this code. The content of this method is always  
     * regenerated by the Form Editor.  
     */
```

```
    @SuppressWarnings("unchecked")
```

```
    Generated Code
```

```
    private void jComboBox1ActionPerformed(java.awt.event.ActionEvent evt) {  
        jLabel1.setText(jComboBox1.getSelectedItem().toString());  
    }
```

```
    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {  
        System.exit(0);  
    }
```

```
    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
        jComboBox1.addItem(jTextField1.getText());  
    }
```

