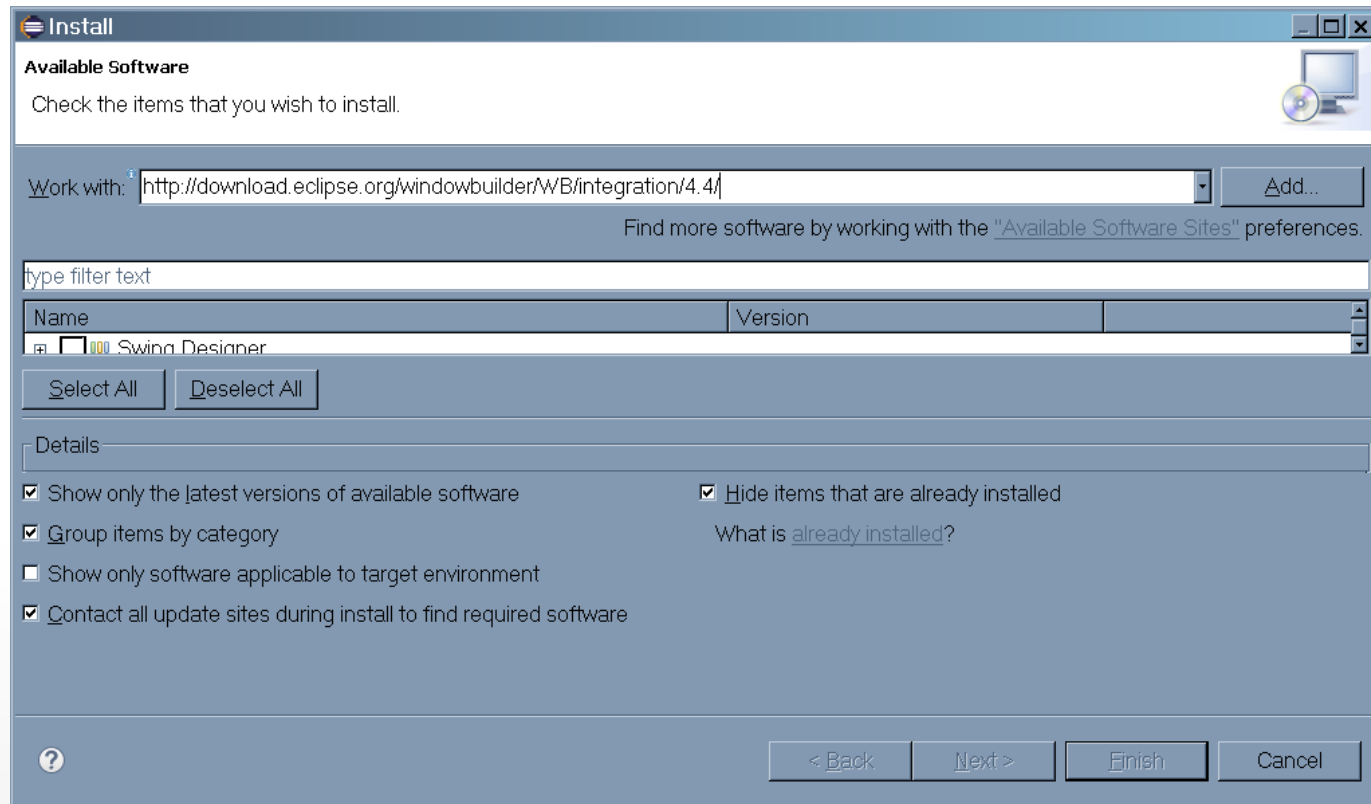


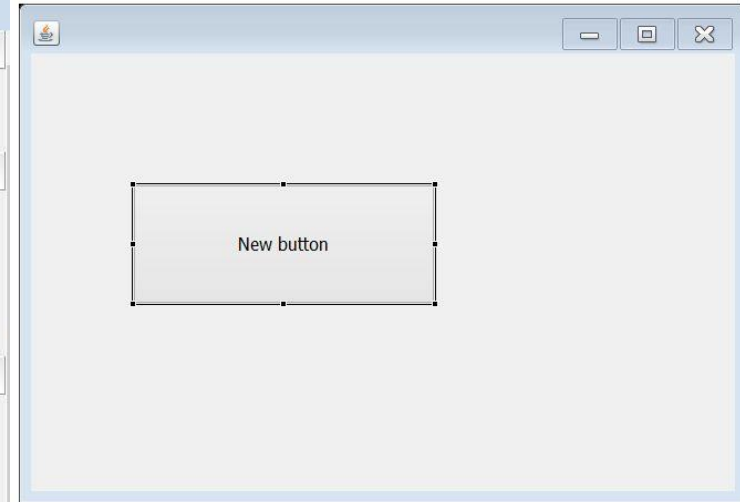
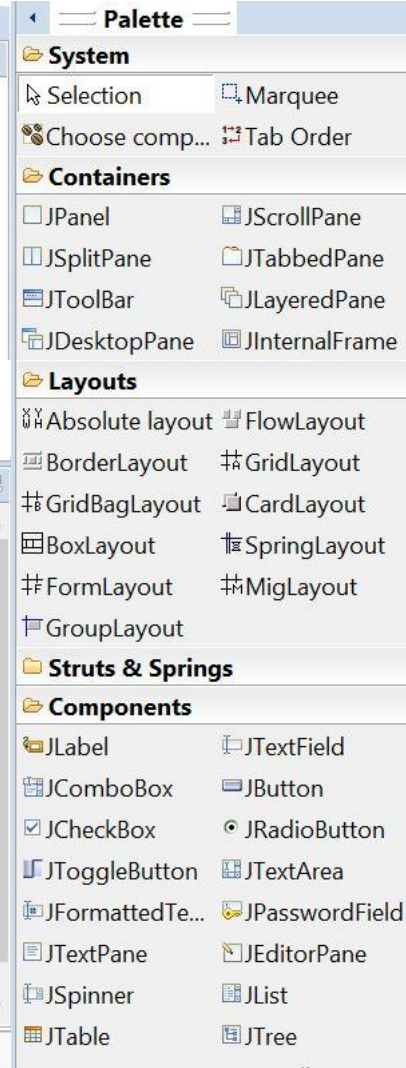
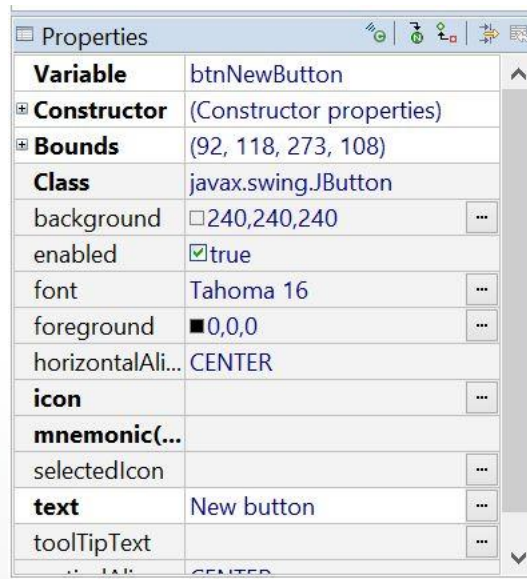
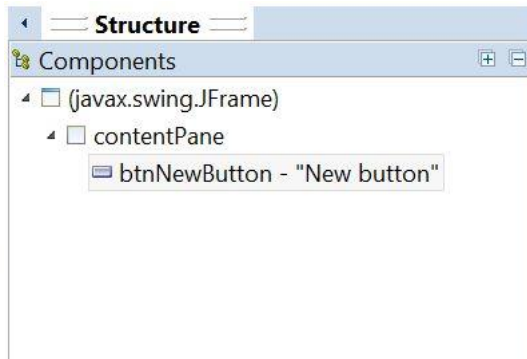
Библиотека Swing

Swing — библиотека для создания графического интерфейса для программ на языке Java. Она содержит ряд графических компонентов (англ. Swing widgets), таких как кнопки, поля ввода, таблицы и т. д.

<http://download.eclipse.org/windowbuilder/WB/integration/4.4/>



Библиотека Swing



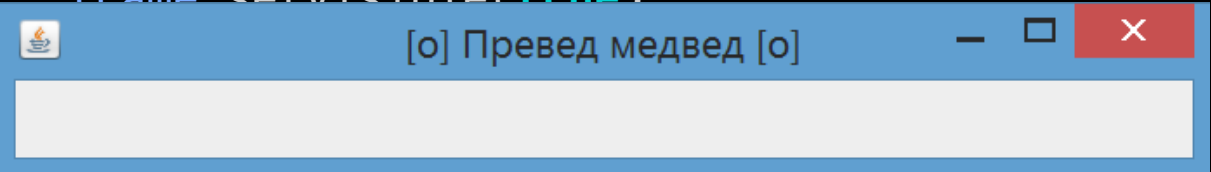
Библиотека Swing

Объект **JFrame** – это объект главного окна программы, на котором будут отображаться остальные компоненты, которые вы хотите показать.

```
import javax.swing.*;

public static void main(String[] args) {
    EventQueue.invokeLater(new Runnable() {
        public void run() {
            try {
                TestWindowBuilder frame = new TestWindowBuilder();
                frame.setVisible(true);
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    });
}

private static void createAndShowGUI() {
    // Создаём фрейм(окно) с заголовком
    JFrame frame = new JFrame("[o] Превед медвед [o]");
    // Размер окна
    frame.setSize(700, 100);
    // Показываем окно
    frame.setVisible(true);
}
```



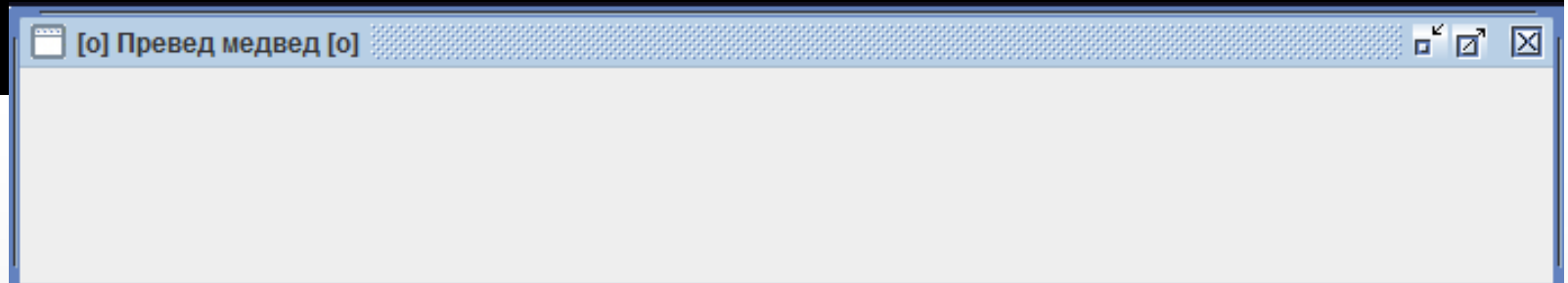
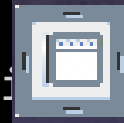
Библиотека Swing

```
// дефолтный стиль ('Look and Feel')
JFrame.setDefaultLookAndFeelDecorated(true);

JFrame frame = new JFrame("[o] Превед медвед [o]");
frame.setSize(700, 100);

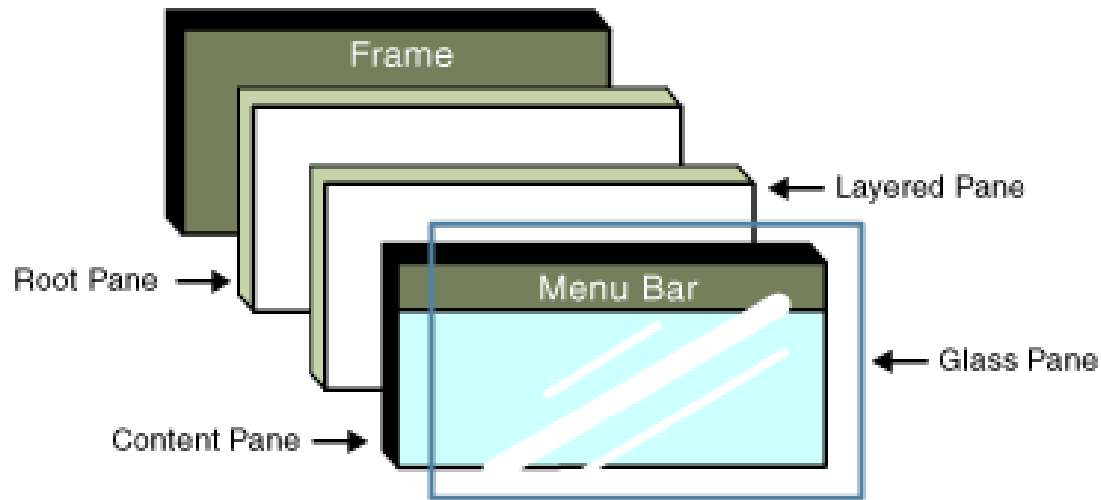
// Контрольно убеждаемся что при закрытии фрейма программа
// закроется
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

// Автоматически упаковываем(подгоняем размер) все элементы
// так, чтобы они поместились во фрейме
frame.pack();
```



Библиотека Swing

JPanel является контейнером, который используется для хранения «виджетов»(элементов управления).



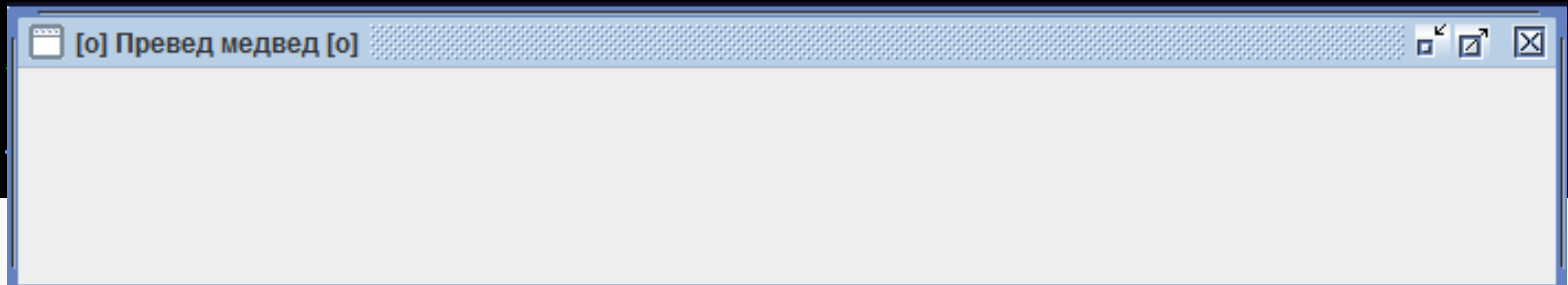
From Sun Microsystems(tm) Website

Панели выкладываются на контейнер верхнего уровня – фрейм. Все элементы управления располагаются на ContentPane

Библиотека Swing

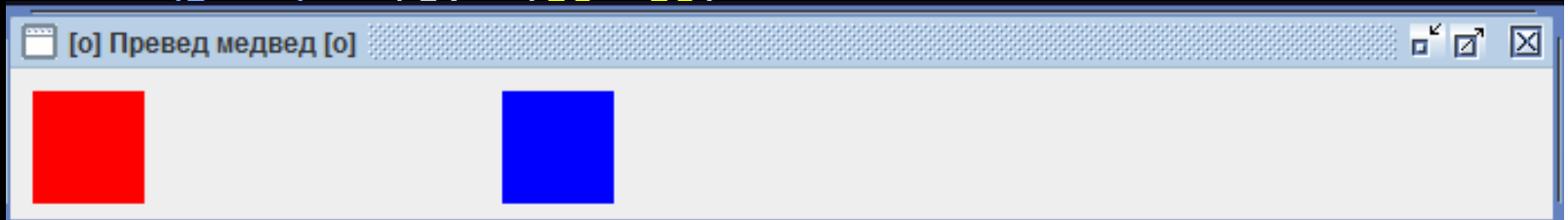
```
public static JPanel createContentPane() {  
    // на панель выкладываются элементы управления  
    JPanel thePanel = new JPanel();  
  
    // панели с содержимым(content) должны быть прозрачными  
    thePanel.setOpaque(true);  
    return thePanel;  
}
```

```
private static void createAndShowGUI() {  
  
    JFrame.setDefaultLookAndFeelDecorated(true);  
    JFrame frame = new JFrame("[o] Превед медвед [o]");
```



Библиотека Swing

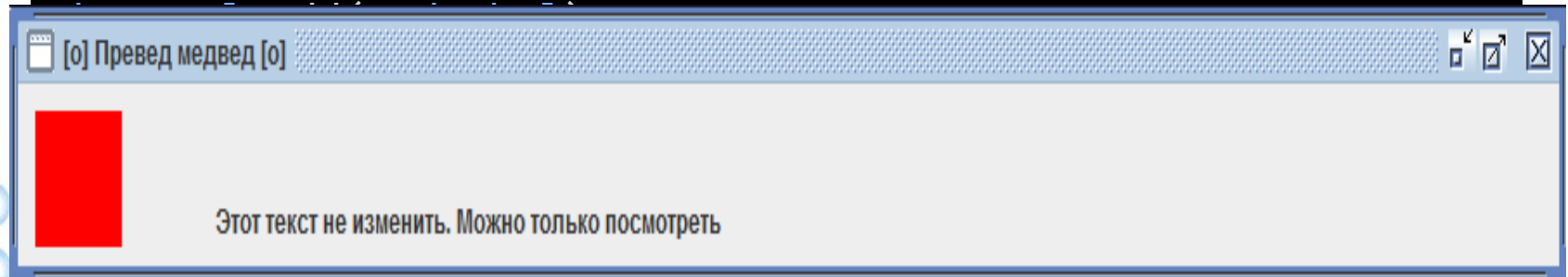
```
public static JPanel createExtContentPane() {  
  
    JPanel thePanel = new JPanel();  
  
    // Менеджер компоновки ставим Null чтобы вручную всё  
    // расставить  
    thePanel.setLayout(null);  
  
    // Ещё одна панель с размером, цветом и расположением  
    // добавляем её на предыдущую  
    JPanel redPanel = new JPanel();  
    redPanel.setBackground(Color.red);  
    redPanel.setLocation(10, 10);  
    redPanel.setSize(50, 50);  
    thePanel.add(redPanel);  
  
    JPanel bluePanel = new JPanel();  
    bluePanel.setBackground(Color.blue);  
    bluePanel.setLocation(220, 10);  
    bluePanel.setSize(50, 50);  
    thePanel.add(bluePanel);  
}
```



```
bluePanel.setBackground(Color.blue);  
bluePanel.setLocation(220, 10);  
bluePanel.setSize(50, 50);  
thePanel.add(bluePanel);
```

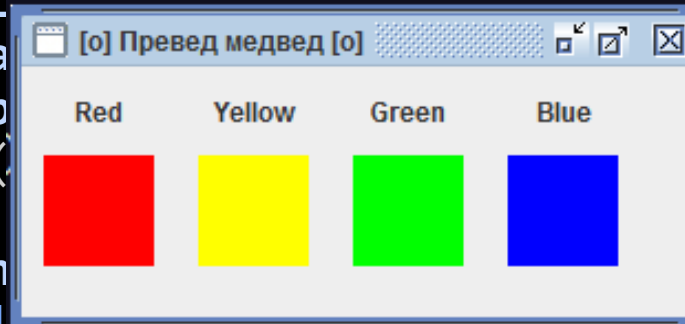

Библиотека Swing

```
// текстовая метка на панели  
JLabel redLabel = new JLabel("Этот текст не изменить. Можно  
только посмотреть");  
redLabel.setLocation(10, 30);  
redLabel.setSize(500, 40);  
redLabel.setHorizontalAlignment(0);
```



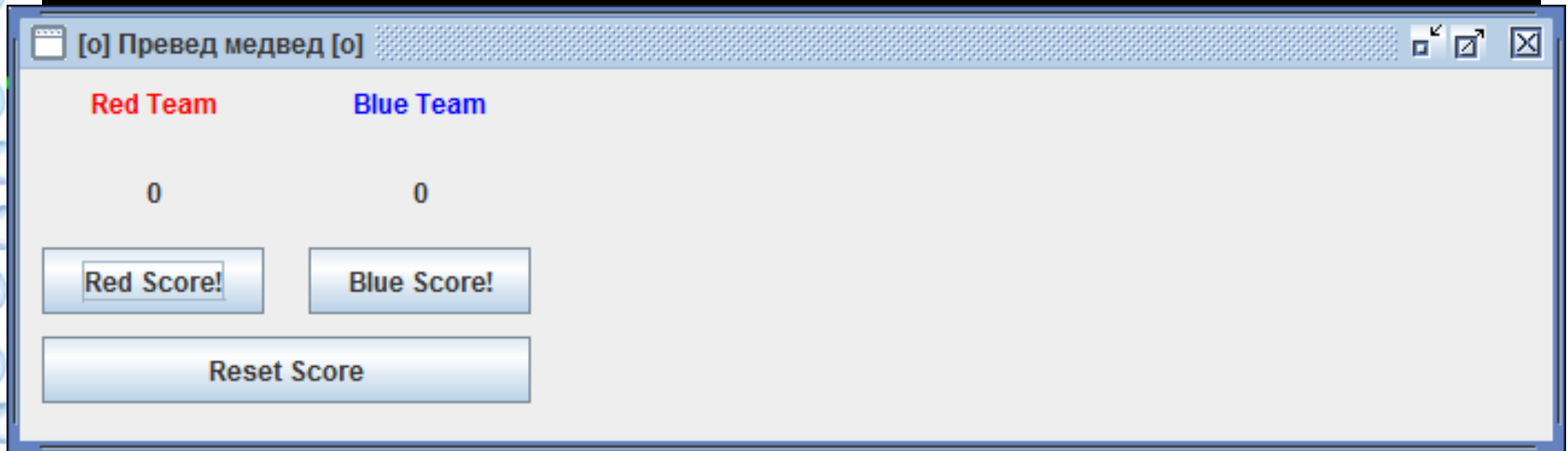
Библиотека Swing

```
// 1
JPanel textPanel = new JPanel();
textPanel.setSize(260, 30);
thePanel.add(textPanel);
// 1.1
JLabel redLabel = new JLabel("Red");
textPanel.add(redLabel);
JLabel yellowLabel = new JLabel("Yellow");
JLabel greenLabel = new JLabel("Green");
JLabel blueLabel = new JLabel("Blue");
// 2 JPanel panelForPanels = new JPanel();
thePanel.add(panelForPanels);
// 2.1
JPanel redPanel = new JPanel();
panelForPanels.add(redPanel);
JPanel yellowPanel = new JPanel();
JPanel greenPanel = new JPanel();
JPanel bluePanel = new JPanel();
...
```



Библиотека Swing

```
JPanel thePanel = new JPanel();  
// Метки  
JPanel titlePanel = new JPanel();  
thePanel.add(titlePanel);  
JLabel redLabel = new JLabel("Red Team");  
titlePanel.add(redLabel);  
...
```



```
redButton.setLocation(0, 0);  
redButton.setSize(100, 30);  
buttonPanel.add(redButton);  
buttonPanel.add(blueButton);  
...
```



Библиотека Swing

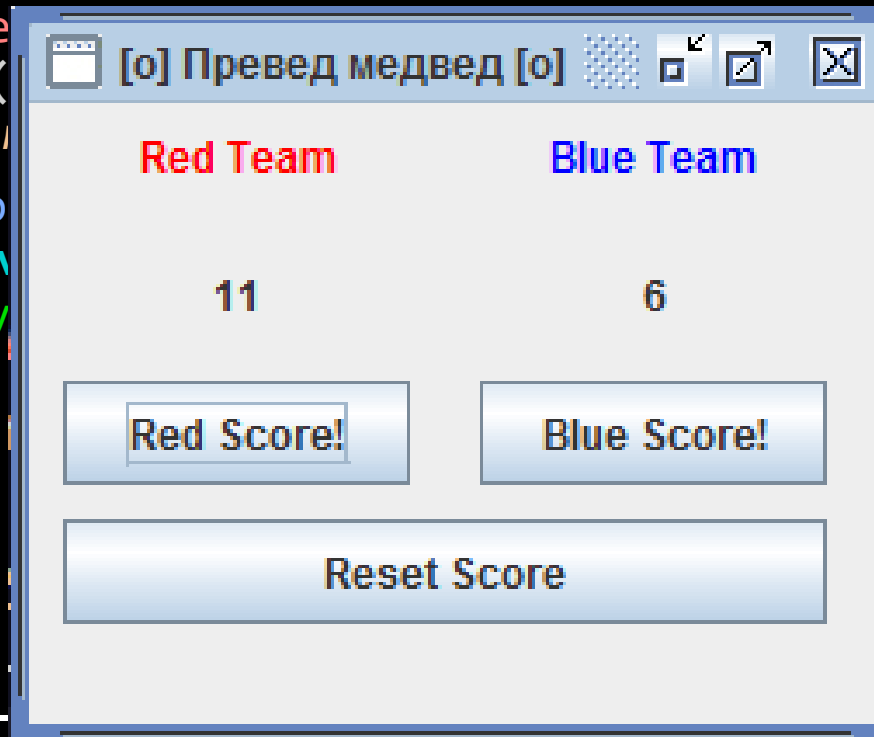
Слушатели(**Listeners**) событий делают ваш интерфейс интерактивным. Когда пользователь делает что-нибудь в GUI (изменяет размер, нажимает кнопку мыши или вводит текст в текстовое поле, происходит определённое "событие". Слушатель реагирует на возникновение события и позволяет выполнять код в ответ на него.

Слушатель у JButton это объект типа ActionListener. Он может работать(слушать) с кнопками, текстовыми полями и пунктами меню.

Библиотека Swing

```
static ActionListener listener = new ActionListener() {  
    @Override  
    public void actionPerformed(ActionEvent e) {
```

```
        Object  
        if (  
        btnNewButto  
        public \br/>        // y  
    }  
});
```



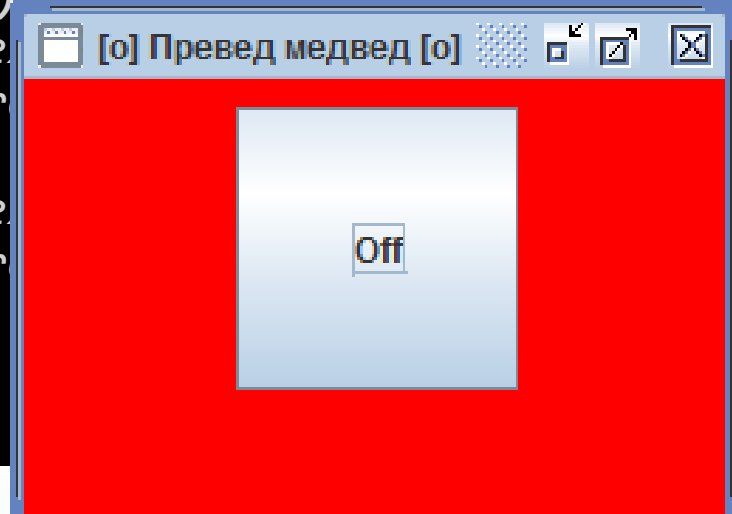
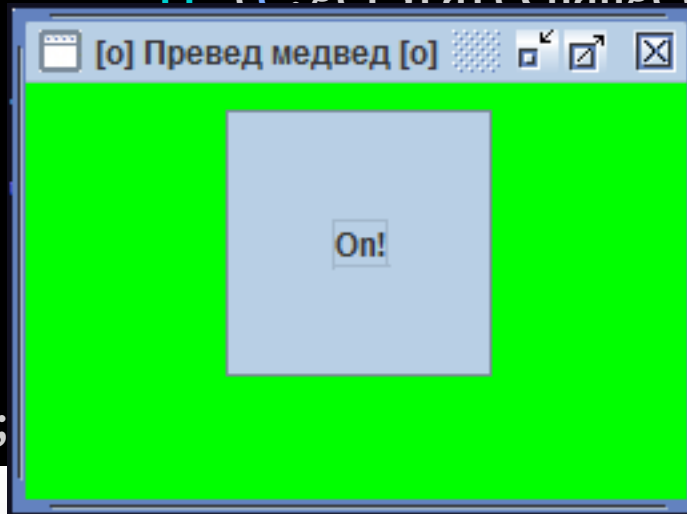
```
        redButton.addActionListener(listener);  
        blueButton.addActionListener(listener);  
        resetButton.addActionListener(listener);
```

```
1;  
tener() {  
    } {  
  
Amount);  
reAmount);
```

Библиотека Swing

```
toggleButton = new JToggleButton("Off");
toggleButton.setLocation(75, 10);
toggleButton.setSize(100, 100);
toggleButton.addItemListener(itemListener);
thePanel.add(toggleButton);
```

```
static ItemListener itemListener = new ItemListener() {
    public void itemStateChanged(ItemEvent e) {
        if (e.getStateChange() == ItemEvent.SELECTED) {
```



```
};
```

Библиотека Swing

Текстовые поля(**JTextField**) позволяют пользователю вводить текст в вашу программу

```
usernameField = new JTextField(8);  
usernameField.setLocation(0, 0);  
usernameField.setSize(100, 30);  
panelForTextFields.add(usernameField);  
  
loginField = new JTextField(8);
```

Two overlapping Java Swing windows are shown. The top window has a title bar with "[o] Превед медвед [o]" and a title "Авторизация". The bottom window also has a title bar with "[o] Превед медвед [o]" and a title "Авторизируемся....". Inside the bottom window, there are two text input fields. The first is labeled "Логин" and contains the text "Вася". To its right, the word "Верно!" is displayed in green. The second is labeled "Пароль" and contains the text "Пупкин". To its right, the word "Верно!" is also displayed in green. At the bottom center of the bottom window is a button labeled "Логин".

Библиотека Swing

Специальное текстовое поле (**JPasswordField**) позволяет пользователю вводить пароль без отображения на экране

```
LoginField2 = new JPasswordField(8);
LoginField2.setEchoChar('+');
panelForTextFields.add(LoginField2);
...
char[] answer = { 'П', 'у', 'п', 'к', 'и', 'н' };
char[] input = LoginField2.getPassword();
if (Arrays.equals(input, answer)) {
    JOptionPane.showMessageDialog(Color.green);
}
```

[o] Превед медвед [o]

Авторизация

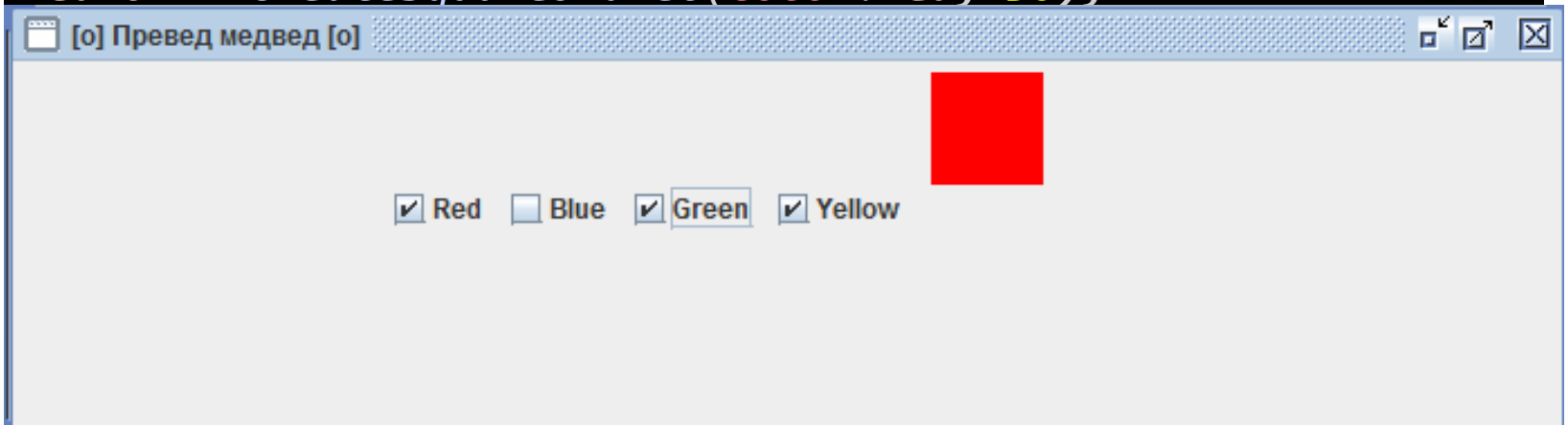
Логин Неверно!

Пароль Верно!

Библиотека Swing

Переключатель(чекбокс, флажок) **JCheckBox** - виджет, который позволяет выбрать(отметить) элемент

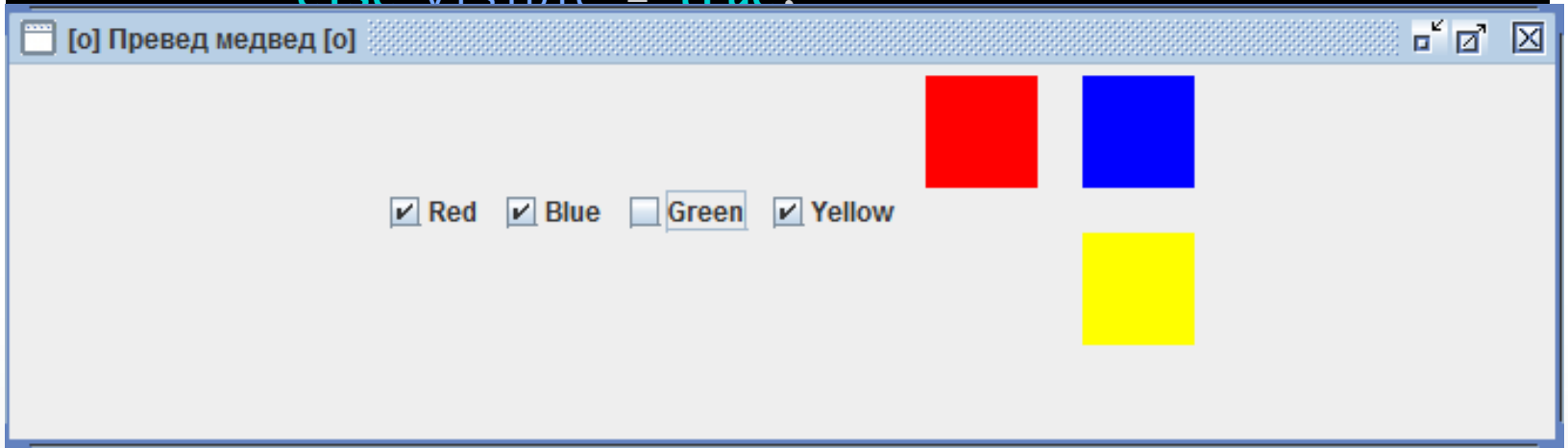
```
redCB = new JCheckBox("Red");  
redCB.setSelected(true);  
checkBoxPanel.add(redCB);  
  
blueCB = new JCheckBox("Blue");  
...  
redBox = createSquareJPanel(Color.red, 50);
```



Библиотека Swing

```
redCB.addItemListener(itemListener2);  
blueCB.addItemListener(itemListener2);  
...
```

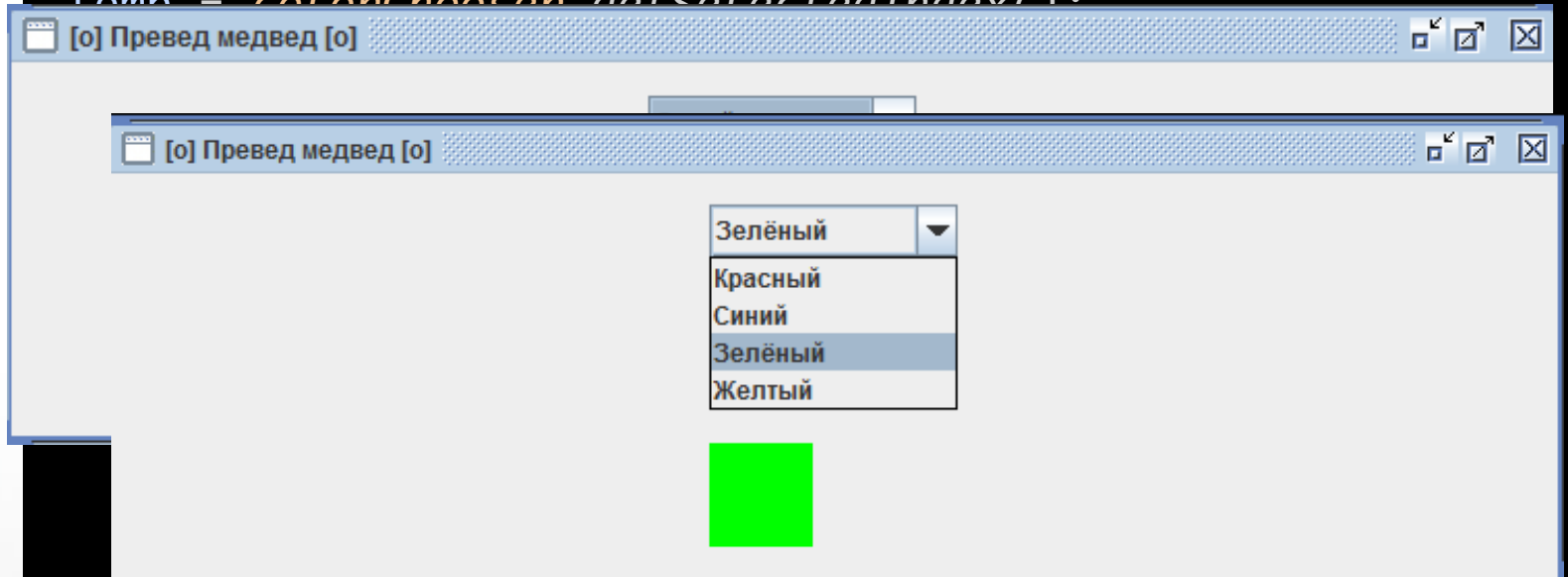
```
static ItemListener itemListener2 = new ItemListener() {  
    public void itemStateChanged(ItemEvent e) {  
        boolean visible = false;  
        if (e.getStateChange() == ItemEvent.DESELECTED)  
            visible = false;  
        else visible = true;  
    }  
};
```



Библиотека Swing

Поле со списком(**JComboBox**) является очень удобным раскрывающимся списком, которое позволяет выбрать один вариант из списка. JComboBox использует ActionListener.

```
String colors[] = {"Красный", "Синий", "Зелёный", "Желтый"};  
colorChooser = new JComboBox(colors);  
colorChooser.setSelectedIndex(1);  
colorChooser.addActionListener(listener);  
...  
temp = colorChooser.getSelectedIndex();
```



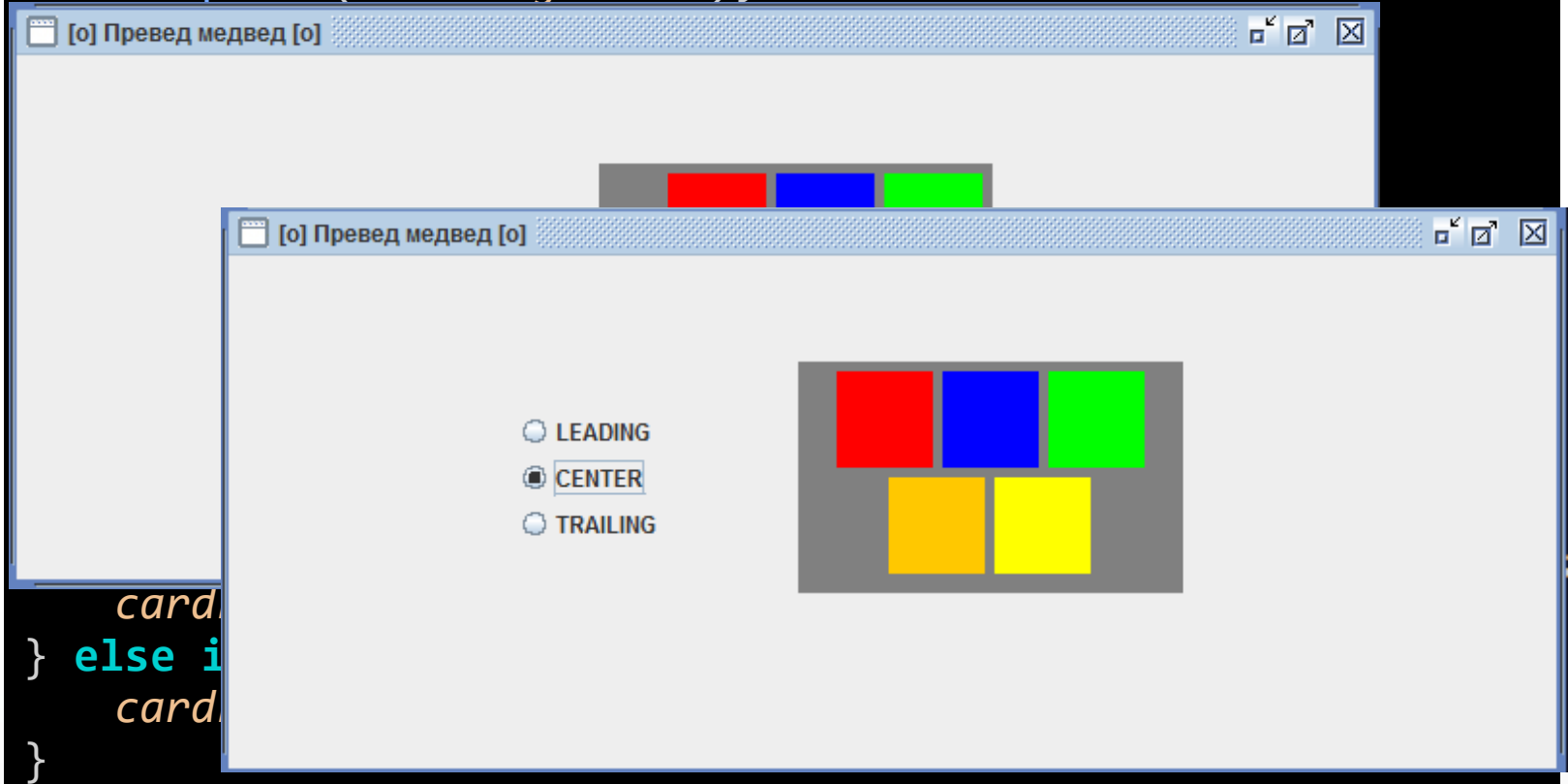
Библиотека Swing

Радио кнопки (кнопки с единичным выбором, **JRadioButton**) – элемент интерфейса, позволяющий выбрать один вариант из нескольких.

```
String names[] = {"LEADING", "CENTER", "TRAILING"};
...
LeadingButton = new JRadioButton(names[0]);
LeadingButton.addActionListener(listener);
LeadingButton.setSelected(true);
centerButton = new JRadioButton(names[1]);
centerButton.addActionListener(listener);
trailingButton = new JRadioButton(names[2]);
trailingButton.addActionListener(listener);
```

Библиотека Swing

```
ButtonGroup btnGroup = new ButtonGroup();  
btnGroup.add(LeadingButton);  
btnGroup.add(centerButton);  
btnGroup.add(trailingButton);
```



```
card  
} else i  
card  
}
```



Библиотека Swing

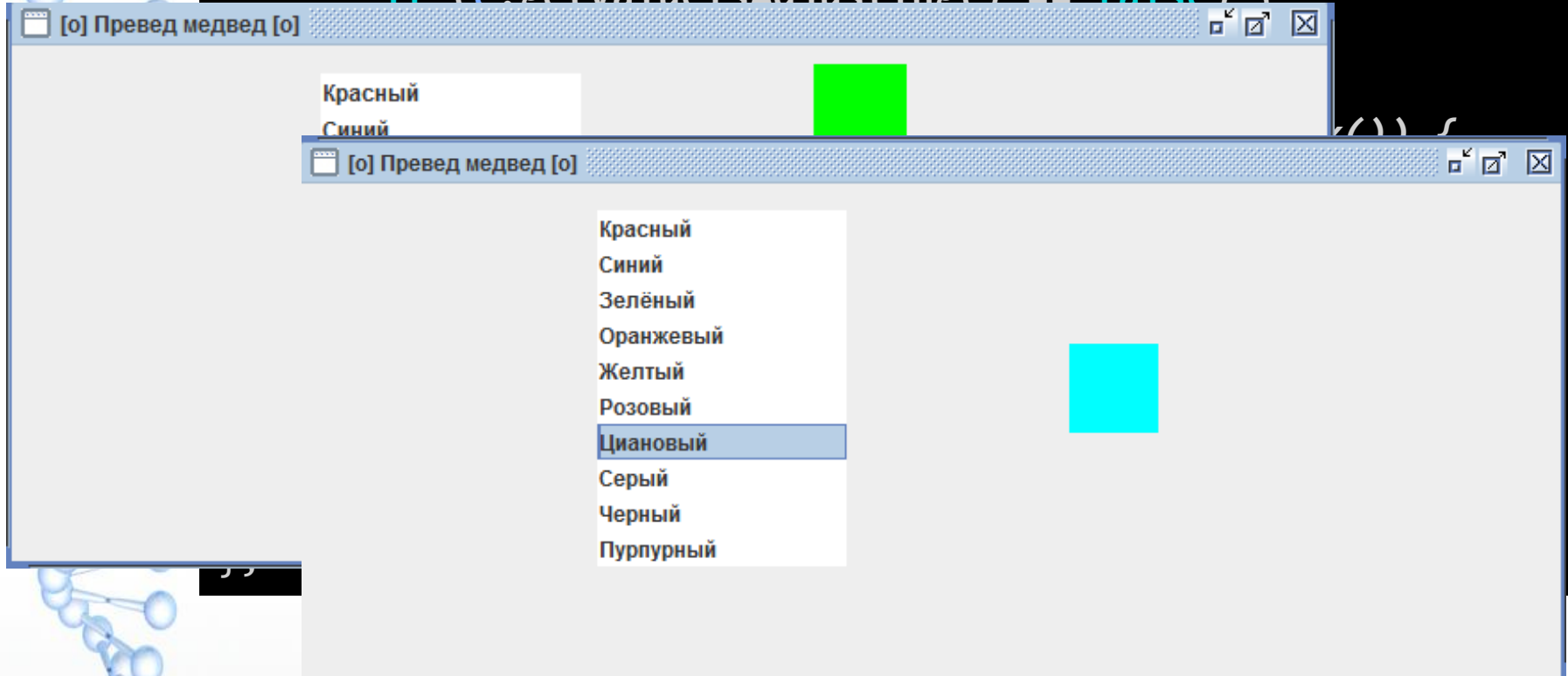
Список (**JList**) отображает элементы в виде списка с возможностью выбора. JList использует слушатель `ListSelectionListener`.

```
String names[] = {"Красный", "Синий", "Зелёный", "Оранжевый",  
"Желтый", "Розовый", "Циановый", "Серый", "Черный", "Пурпурный"}
```

```
boxes = new JList(names);  
boxes.setVisibleRowCount(10);  
boxes.setFixedCellHeight(20);  
boxes.setFixedCellWidth(140);  
boxes.setSelectionMode(ListSelectionModel.SINGLE_SELECTION)  
boxes.addListSelectionListener(listListener);
```

Библиотека Swing

```
static ListSelectionListener listListener = new  
ListSelectionListener() {  
    public void valueChanged(ListSelectionEvent e) {  
        if (e.getValueIsAdjusting() == false) {
```

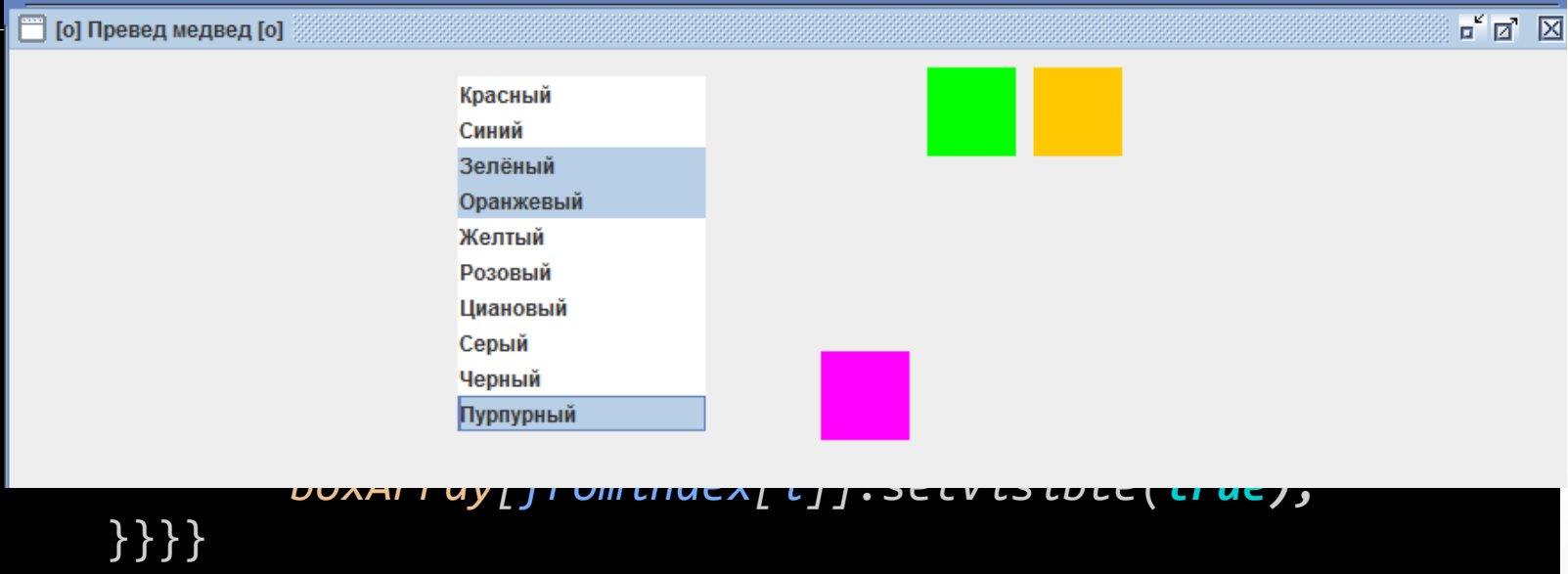


Библиотека Swing

- **SINGLE_SELECTION** - позволяет выбрать только один элемент в списке
- **SINGLE_INTERVAL_SELECTION** - позволяет выбрать более чем один пункт если они находятся рядом в списке (Shift + Mouse).
- **MULTIPLE_INTERVAL_SELECTION** - позволяет выбрать несколько элементов (Ctrl + Mouse)

```
boxes.setSelectionMode
```

```
(ListSelectionMode MULTIPLE_INTERVAL_SELECTION):
```





Библиотека Swing

Для добавления и удаления элементов из списка можно воспользоваться объектом **ListModel**

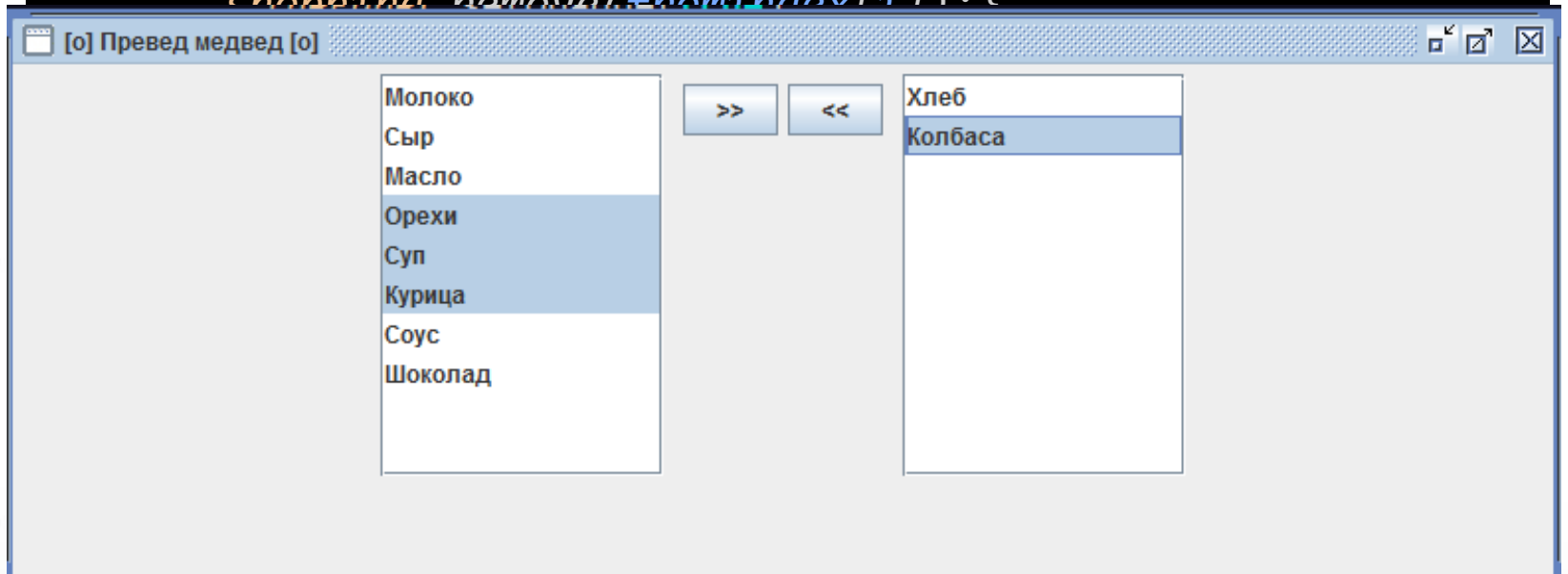
```
DefaultListModel shopping = new DefaultListModel();
DefaultListModel items = new DefaultListModel();

for (int i = 0; i < shoppingItems.length; i++) {
    shopping.addElement(shoppingItems[i]);
}

itemList = new JList(shopping);
itemList.setSelectionMode
    (ListSelectionModel.MULTIPLE_INTERVAL_SELECTION);
JScrollPane list1 = new JScrollPane(itemList);
```

Библиотека Swing

```
if(e.getSource() == buttonin){  
    int[] fromindex = itemList.getSelectedIndices();  
    Object[] from = itemList.getSelectedValues();  
    for(i = 0; i < from.length; i++){  
        items.addElement(from[i]); }  
    for(i = (fromindex.length-1); i >=0; i--){  
        shopping.remove(fromindex[i]); }  
}
```



Библиотека Swing

Область текста (**JTextArea**) - просто большой JTextField, который может отображать более одной строки текста

```
storyArea = new JTextArea(story, 5, 30);  
storyArea.setEditable(false);  
storyArea.setLineWrap(true);  
storyArea.setWrapStyleWord(true);
```

The image shows two screenshots of a Java Swing application window titled "[o] Превед медвед [o]".

Top Screenshot:

- Text Area:** Contains the text:
Когда я был маленьким мальчиком, я всегда мечтал
иметь странную Газету .
Я бы ездил на ней повсюду и заезжал к маме
Иногда я бы посещал Санузел
Я бы любил Газету и она бы любила меня.
- Form:**
 - Gender: ☒ Мальчик, ☐ Девочка
 - Locations: ☐ Офис, ☐ Банк, ☐ Супермаркет, ☐ Бассейн, ☒ Санузел
 - Vehicle/Item: A dropdown menu showing "Газету" and "Машину".

Bottom Screenshot:

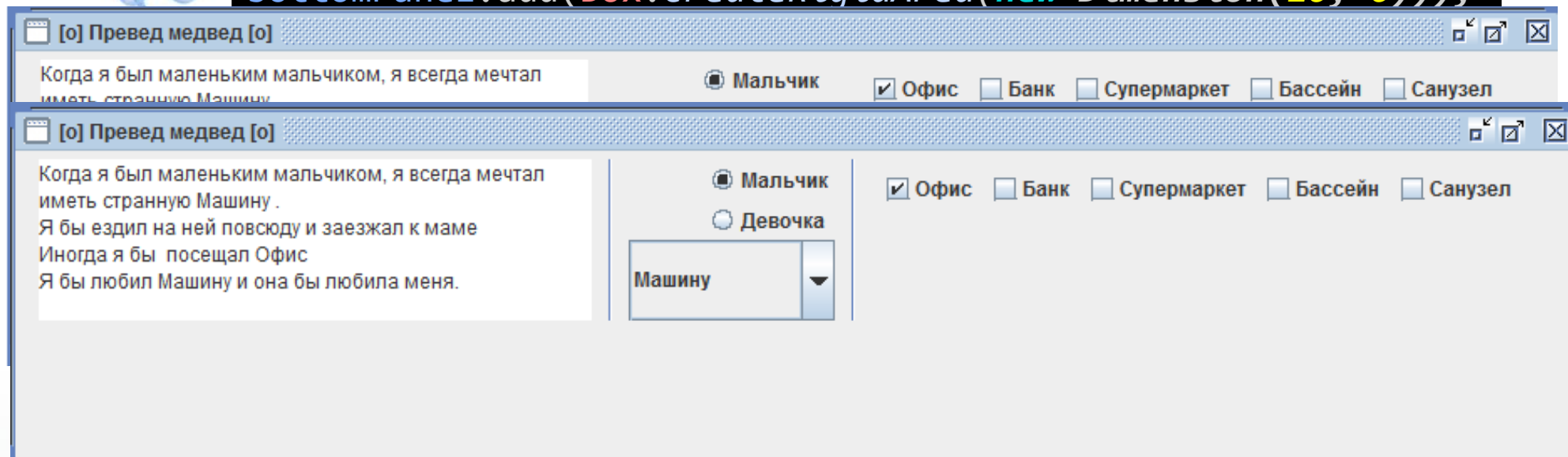
- Text Area:** Contains the text:
Когда я была маленькой девочкой, я всегда мечтала
иметь странную Машину .
Я бы ездила на ней повсюду и заезжала к маме
Иногда я бы посещала Офис Банк
Я бы любила Машину и она бы любила меня.
- Form:**
 - Gender: ☐ Мальчик, ☒ Девочка
 - Locations: ☒ Офис, ☒ Банк, ☐ Супермаркет, ☐ Бассейн, ☐ Санузел
 - Vehicle/Item: A dropdown menu showing "Машину", "Лошадь", "Барабашку", "Корову", and "Газету". "Барабашку" is selected.

Библиотека Swing

Разделитель (**JSeparator**) - виджет, который рисует линию, чтобы разделить другие виджеты в разметке.

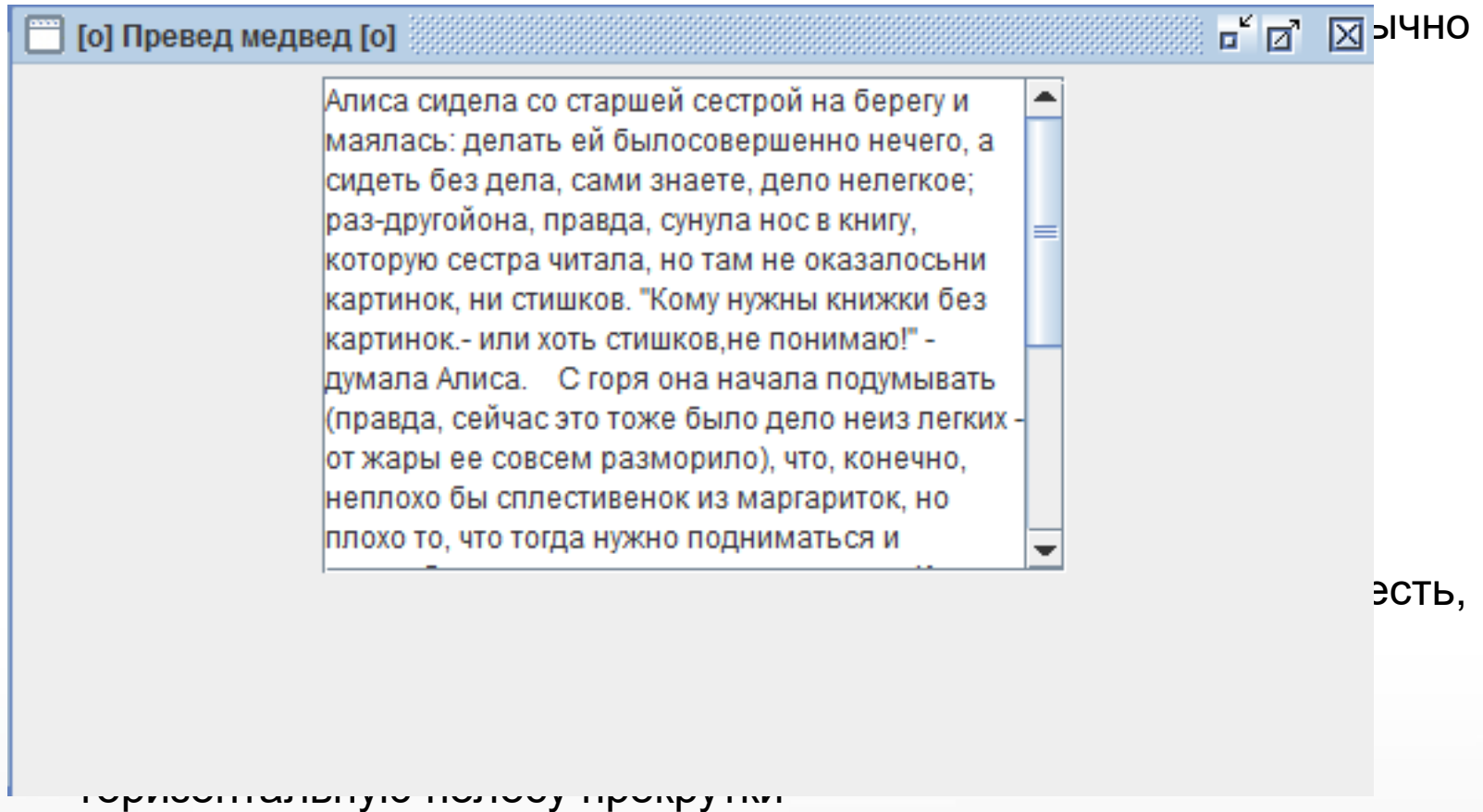
- `SwingConstants.VERTICAL`
- `SwingConstants.HORIZONTAL`

```
bottomPanel.add(Box.createRigidArea(new Dimension(10, 0)));  
bottomPanel.add(storyArea);  
bottomPanel.add(Box.createRigidArea(new Dimension(10, 0)));
```



Библиотека Swing

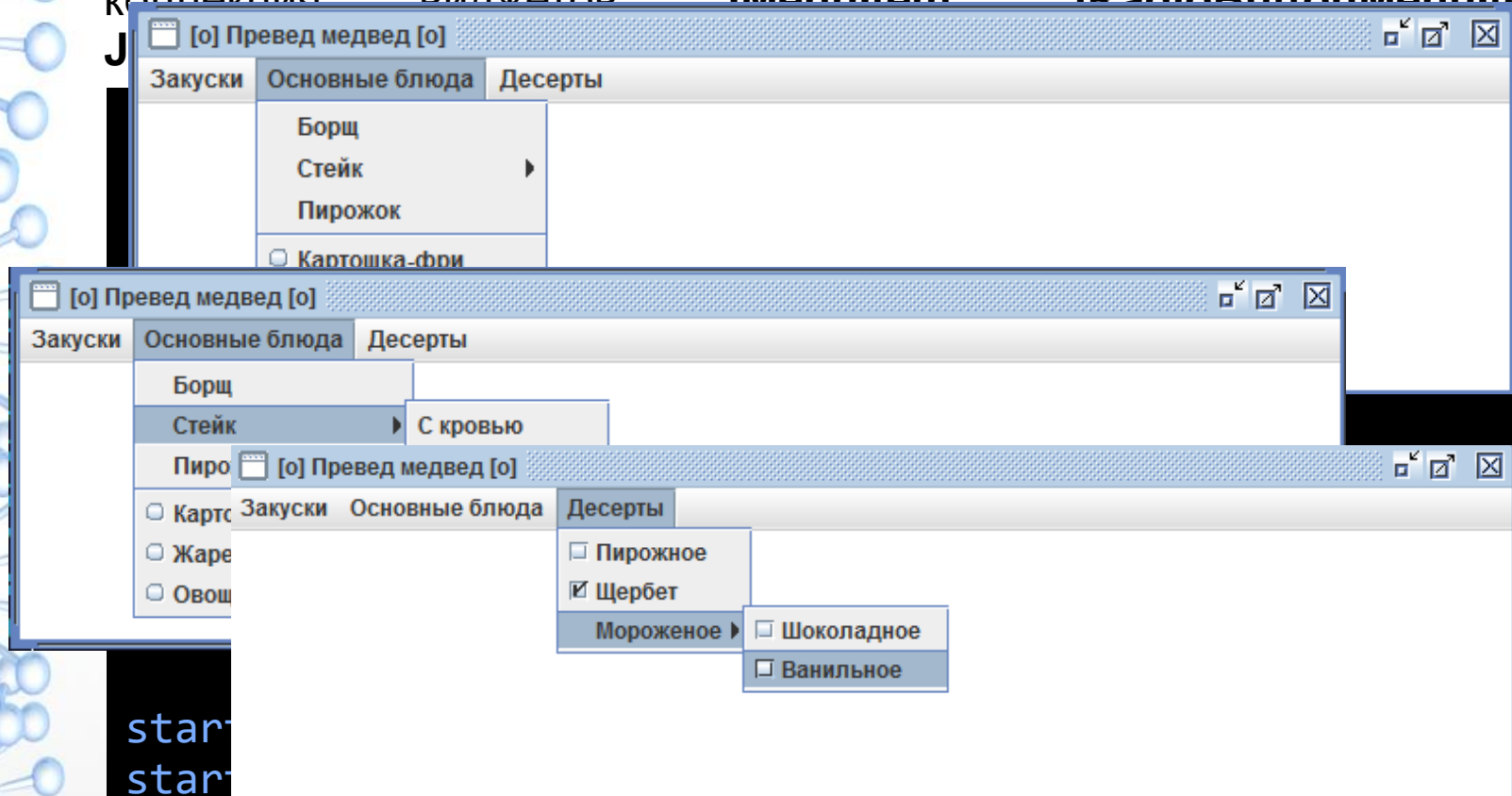
Область прокрутки (**JScrollPane**) используются в разработке GUI, чтобы ограничить виджет определенного размера на экране и дать возможность прокрутки вверх и вниз, влево и вправо, если



есть,

Библиотека Swing

Меню программы (**JMenuBar**), которое находится в верхней части окна программы. JMenuBar является виджетом верхнего уровня. Элементами его могут быть объектами **JMenu**. В меню может быть вложено другое JMenu(подменю). Каждое меню – это коллекция виджетов **JMenuItem**, **JRadioButtonMenuItem**, **JCheckBoxMenuItem**.



```
start  
start  
starter.add(salad);
```




Практика

Создать визуальное отображение ваших устройств в виде списка. Предусмотреть кнопки добавления нового компьютера и редактирования существующего. При попытке редактирования другого устройства выдавать сообщение об имени и мощности этого устройства. Проверять изменяемые параметры RAM для компьютеров и выдавать соответствующие предупреждения.