

## Создание оконных приложений в среде программирования Java

*Андрюенко Иван Сергеевич*

*Приамурский государственный университет имени Шолом-Алейхема*

*студент*

*Глаголев Владимир Александрович*

*Приамурский государственный университет имени Шолом-Алейхема*

*К.г.н., доцент, доцент кафедры информационных систем, математики и правовой информатики*

### Аннотация

В статье проведено проектирование и разработка программы в среде программирования Java, отражены основные элементы управления и выполнена практическая реализация.

**Ключевые слова:** java, программа, интерфейс.

## Creating windowed applications in the Java programming environment

*Andrienko Ivan Sergeevich*

*Sholom-Aleichem Priamursky State University*

*Student*

*Glagolev Vladimir Aleksandrovich*

*Sholom-Aleichem Priamursky State University*

*Ph.D, Associate Professor, Associate Professor of the Department of Information Systems, Mathematics and Legal Informatics*

### Abstract

The article describes the design and development of the program in the Java programming environment, reflects the main controls and makes a practical implementation.

**Keywords:** java, programm, interface.

Java — это язык программирования общего назначения, который следует парадигме объектно-ориентированного программирования и подходу «Написать один раз и использовать везде». Java используется для настольных, сетевых, мобильных и корпоративных приложений. Java не так дружелюбен к новичкам, однако довольно прост для любого разработчика с базовым пониманием фреймворков, пакетов, классов и объектов. Язык прост, типизирован и предсказуем, что позволяет учиться логически размышлять в правильном направлении [1].

Также важным преимуществом Java является независимость от аппаратной части и ОС. Важно лишь наличие исполняющей среды и JVM. А компьютерная архитектура в целом значения не имеет. Байт-код легко интерпретируется на любой машине. Подход доказал свою состоятельность во многом благодаря динамической компиляции. Кроссплатформенностью отличается также интерфейс, реализованный в системных библиотеках [2].

Цель данной статьи рассмотреть особенности разработки оконного приложения в среде программирования Java.

Что бы подробно изучить данную тематику в работе создана программа, которая будет работать с данными. А именно сможет сохранить вводимые в неё данные в отдельном текстовом документе, который пользователь потом сможет с легкостью открыть и прочитать. Все это будет сопровождаться графическим интерфейсом. Данная программа будет обладать кроссплатформенностью – подходить на любые операционные системы.

Для создания графического интерфейса будем использовать класс JFrame и Swing. Для этого мы должны импортировать все необходимые классы (см. рис.1).

```
1 package com.company;
2
3 import java.awt.*;
4 import java.awt.event.*;
5 import javax.swing.*;
```

Рисунок 1 – Импорт необходимых классов

Далее создадим основные элементы управления. Это будут кнопки, поля ввода данных и флажки. Для этого надо унаследовать функции от класса JFrame. В итоге получим следующее (см. рис.2).

```
1 package com.company;
2
3 import java.awt.*;
4 import java.awt.event.*;
5 import javax.swing.*;
6
7 class app extends JFrame {
8     private JButton button = new JButton( text: "Ок");
9     private JTextField input = new JTextField( text: "", columns: 1);
10    private JLabel label = new JLabel( text: "Ввести:");
11    private JCheckBox check = new JCheckBox( text: "Сохранить", selected: true);
12    private JTextField way = new JTextField( text: "Куда сохранять:", columns: 10);
```

Рисунок 2 – Добавление элементов управления

Чтобы данные элементы управления «кнопки» выводились в оконном приложении необходимо написать определенные функции для их вывода.

Так же сразу укажем сколько пикселей в ширину и высоту будет наше приложение, и где по координатам оно будет появляться на нашем экране (см. рис.3).

```
1 package com.company;
2
3 import java.awt.*;
4 import java.awt.event.*;
5 import javax.swing.*;
6
7 class app extends JFrame {
8     private JButton button = new JButton( text: "Ок");
9     private JTextField input = new JTextField( text: "", columns: 1);
10    private JLabel label = new JLabel( text: "Ввести:");
11    private JCheckBox check = new JCheckBox( text: "Сохранить", selected: true);
12    private JTextField way = new JTextField( text: "Куда сохранять:", columns: 10);
13    private JTextField name = new JTextField( text: "Название файла:", columns: 6);
14
15    public app() {
16        super( title: "Приложение");
17        this.setBounds( x: 150, y: 150, width: 650, height: 600);
18        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
19
20        Container container = this.getContentPane();
21        container.setLayout(new GridLayout( rows: 6, cols: 0, hgap: 2, vgap: 2));
22        container.add(label);
23        container.add(input);
24
25        ButtonGroup group = new ButtonGroup();
26
27        container.add(check);
28        button.addActionListener(new ButtonEventListener());
29        container.add(button);
30        container.add(way);
31        container.add(name);
32    }
```

Рисунок 3 – Программный код для вывода кнопок

Первоначально программа показывает стандартное окошко с кнопками и полями ввода, при использовании которых ничего не будет происходить. Чтобы данную ситуацию исправить, необходимо добавить специальный класс под названием ButtonEventListener. Теперь с его помощью надо написать программный код чтобы, когда пользователь нажимал кнопку «Ок», далее выводилось сообщение о успешном сохранении файла. (см. рис.4)

```
34 class ButtonEventListener implements ActionListener {
35     public void actionPerformed(ActionEvent e) {
36         String message = "";
37         message += "Данные сохранены!\n";
38         message += "Текст: " + input.getText() + "\n";
39         JOptionPane.showMessageDialog( parentComponent: null,
40             message,
41             title: "Готово",
42             JOptionPane.PLAIN_MESSAGE);
43     }
44 }
```

Рисунок 4 – Добавление функционала кнопкам

Далее добавляем программный код, который будет создавать файл в указанном пользователем месте. Для этого вводим класс WriteToFile. Путь укажет сам пользователь уже в запущенной программе.

После окончания написания кода нужно компилировать программу. Запускается отдельное окно с формами для заполнения. (см. рис.5)

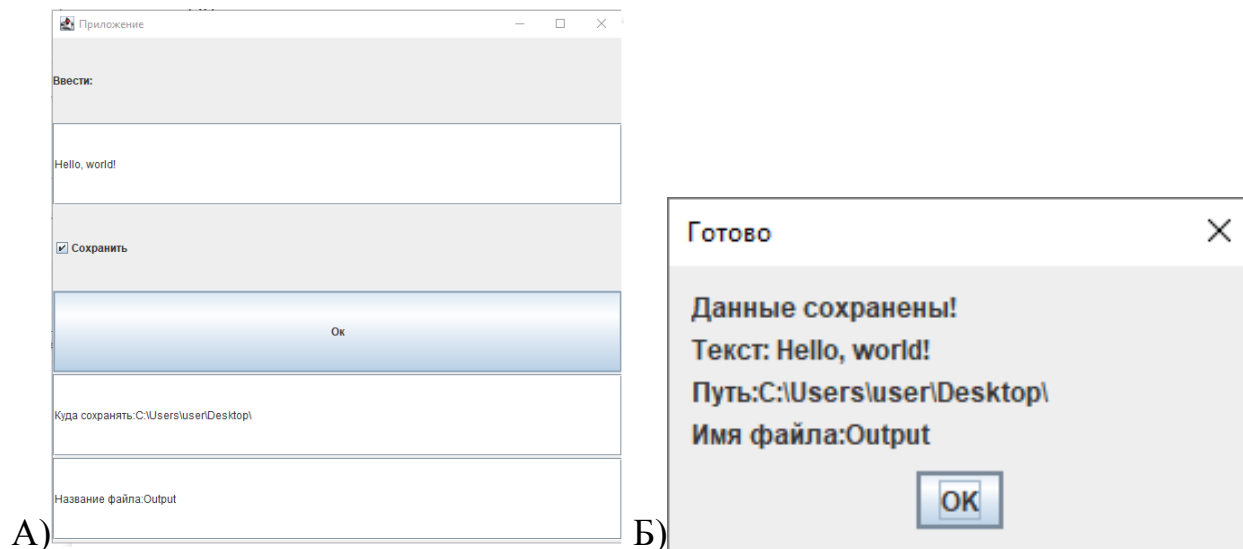


Рисунок 5 – Внешний вид программы (А) и кнопки «ОК» (Б)

Таким образом, в данной статье было описано создание оконного приложения на Java и рассмотрены особенности данного языка программирования. Данное приложение кроссплатформенно.

### Библиографический список

1. Шилдт Г. Java 8. Руководство для начинающих. М.: Вильямс, 2015. 720 с.
2. Портянкин И. А. Swing. Эффектные пользовательские интерфейсы. Питер: М.: Лори, 2005. 640 с.