

Федеральное государственное бюджетное образовательное учреждение высшего  
образования

Казанский государственный энергетический университет

Кафедра информационных технологий и интеллектуальных систем

## *Работа с формой*

Методические указания для выполнения лабораторных работ

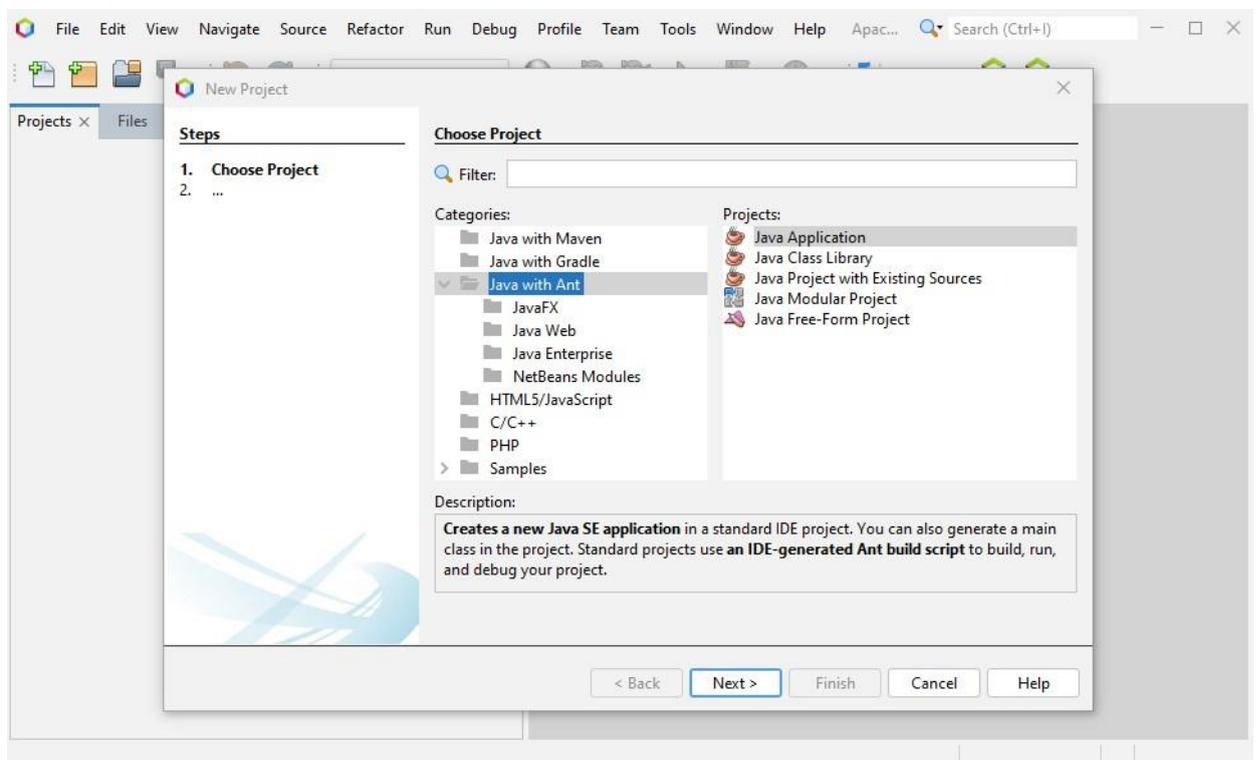


**Казань, 2024**

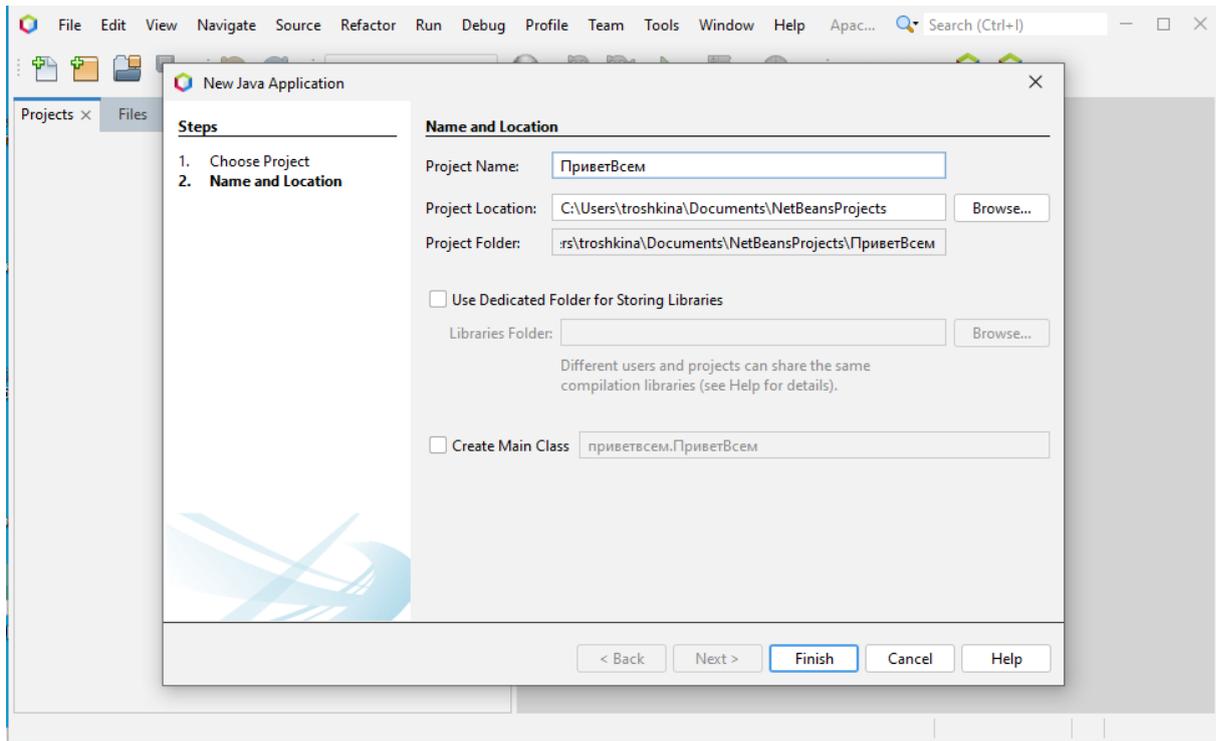
## Лабораторная работа. Элементы работы с формами

**Цель работы:** научиться основам разработки оконных приложений в NetBeans.

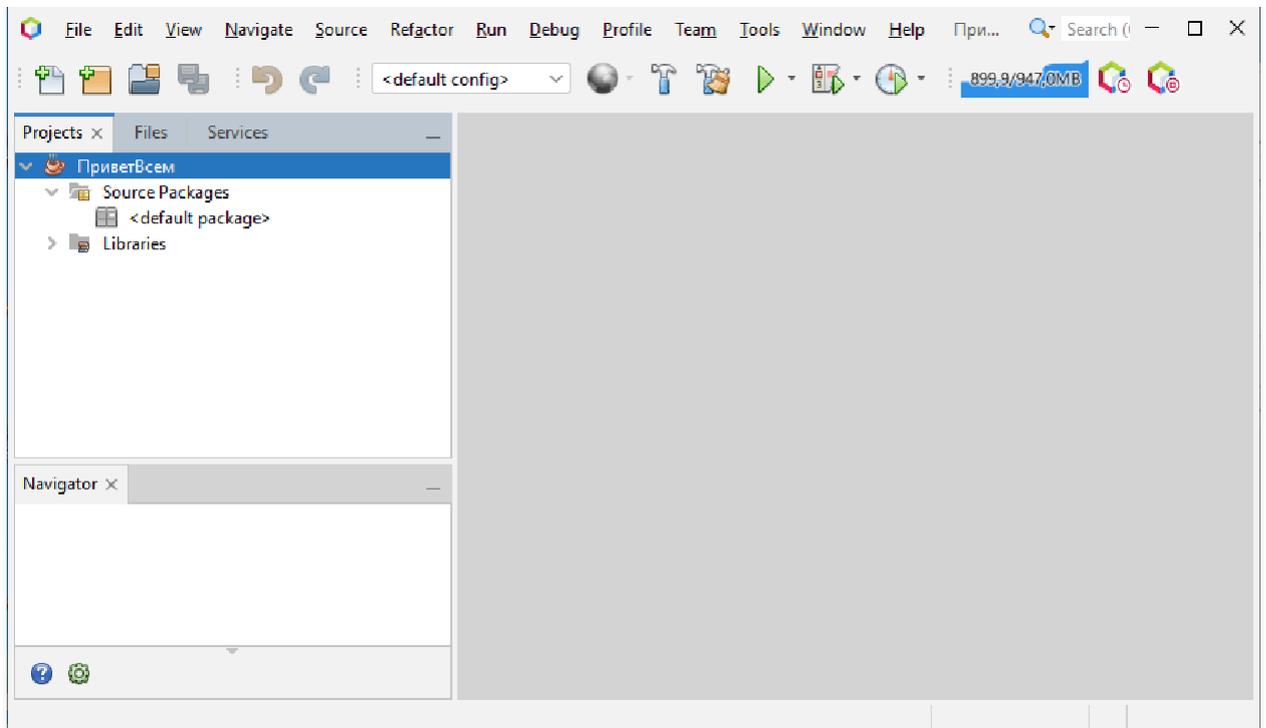
Для разработки оконных приложений для рабочего стола на платформе Java можно использовать среду разработки NetBeans. Запустите NetBeans и создайте проект на основе категории **Java with Ant**, тип проекта: **Java Application**:



Назовите проект, например, **ПриветВсем** и снимите выбор с бинарных переключателей: **Use Dedicated Folder for Storing Libraries** (Использовать специальную папку для хранения библиотек) и **Create Main class** (Создать класс Main):

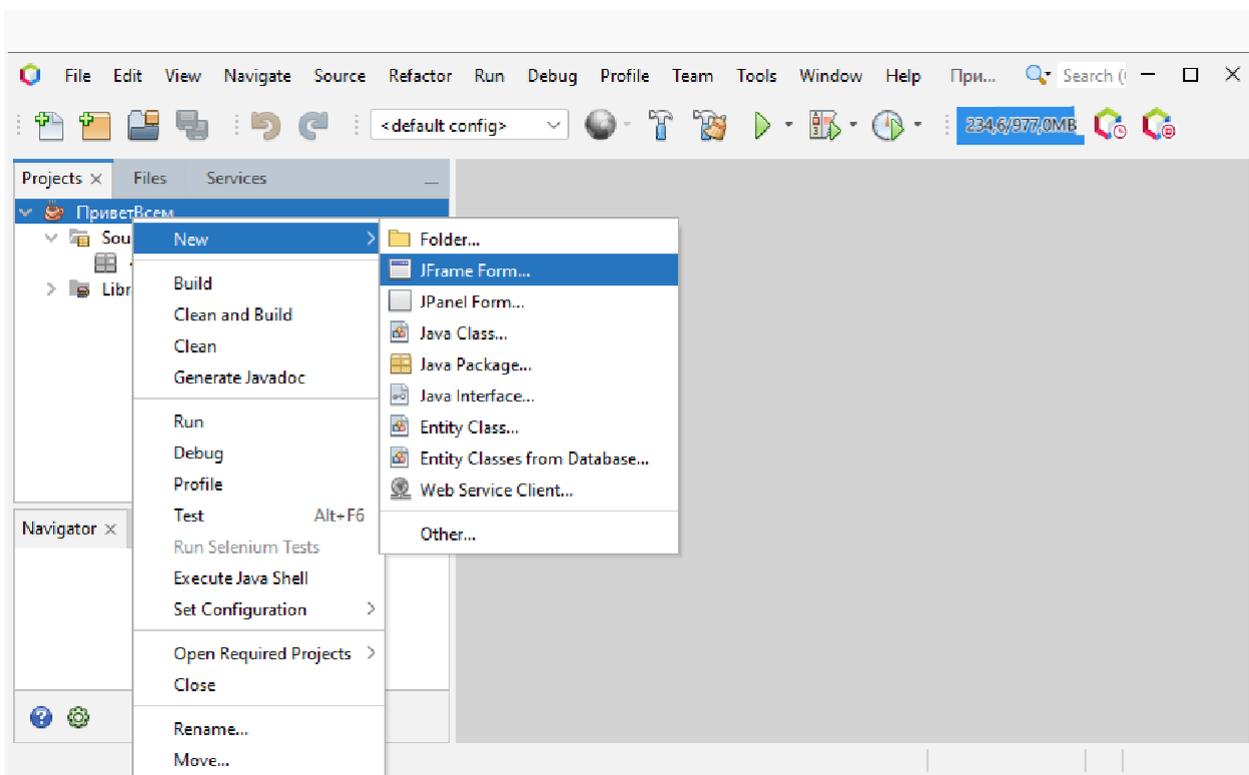


В результате будет создан пустой проект с пакетом «по умолчанию». В папке **Source Packages** создана папка **<default package>**:

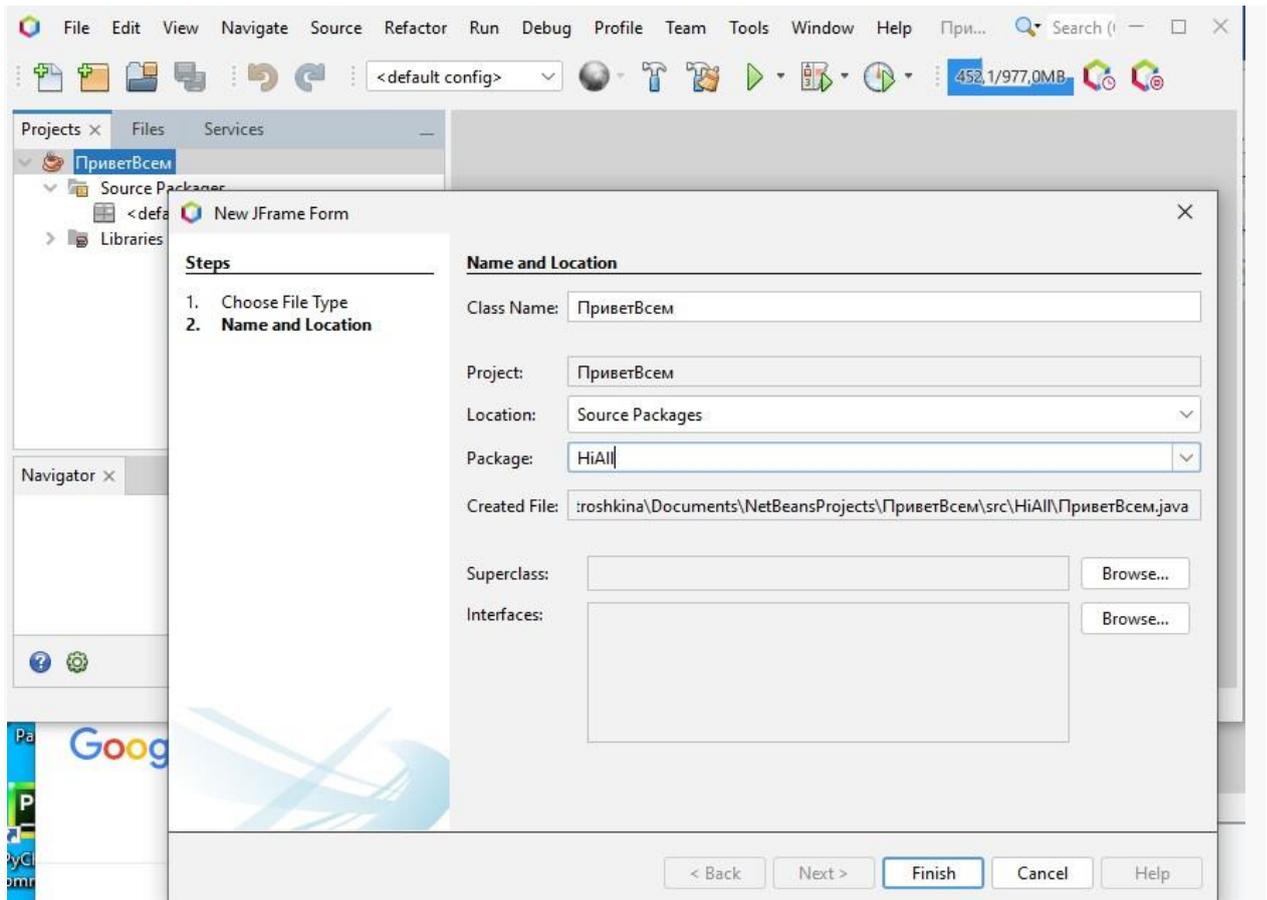


Теперь создадим оконную форму как главный модуль (как точку входа в наше приложение). Для этого создаваемый модуль должен иметь такое же имя, как

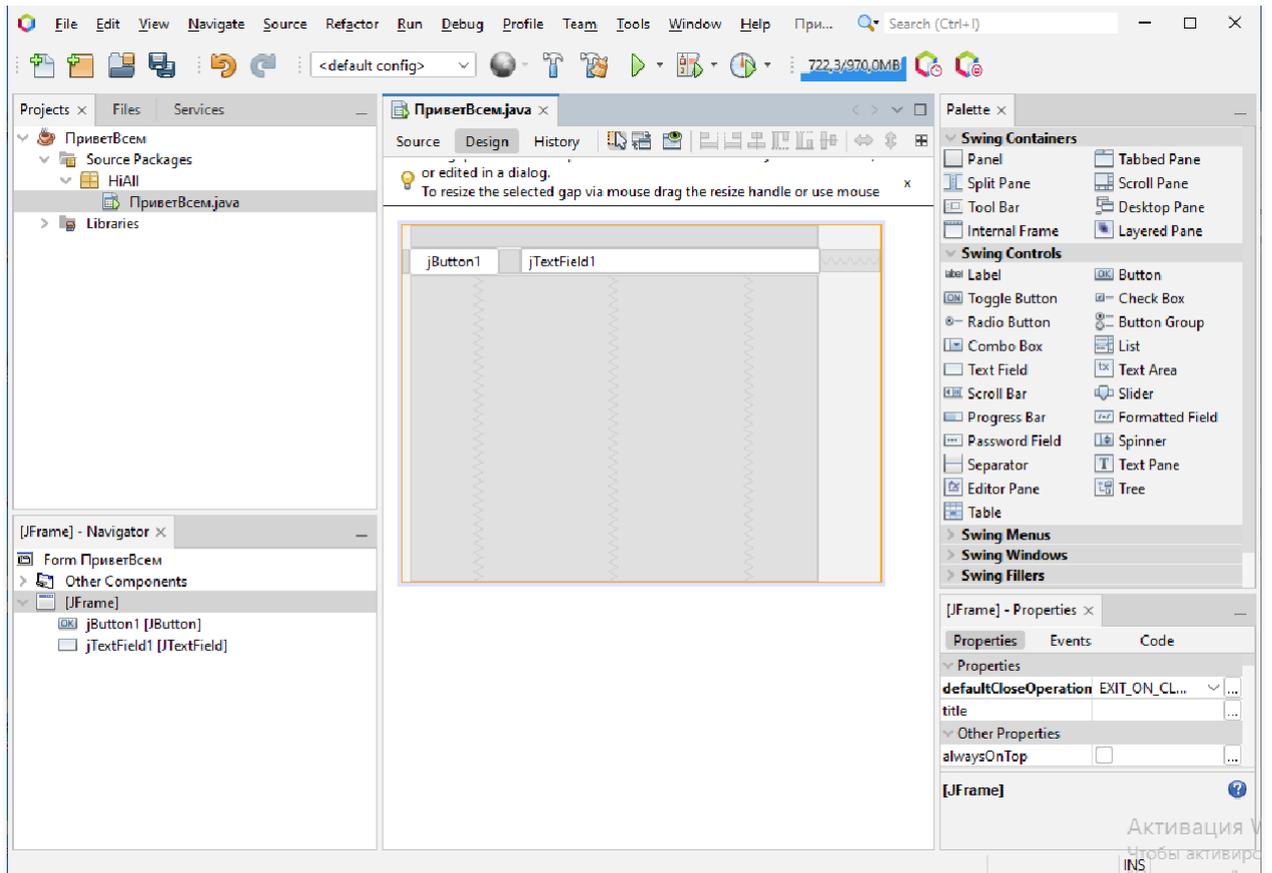
и проект – «ПриветВсем». Добавим в проект форму в виде **оконного контейнера пакета swing – JFrameForm**:



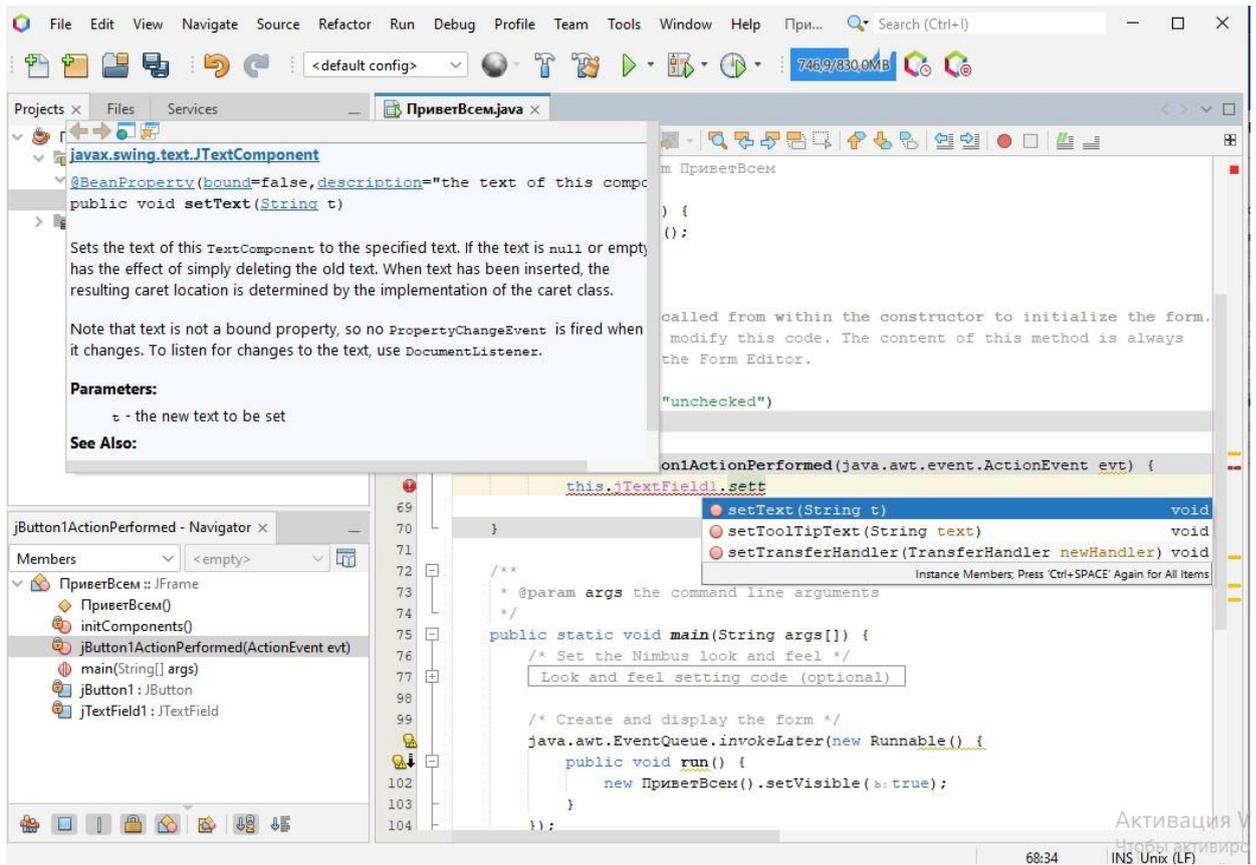
Назовем форму «ПриветВсем» и зададим имя пакету, например – «HiAll»:



Создайте форму и разместите на ней элементы управления: кнопку – «**Button**» и текстовое поле – «**Text Field**»:



На форме при этом, «по умолчанию» создаются объекты **jButton1** и **jTextField1**. Зададим обработчик нажатия на кнопку. Здесь можно, в принципе, использовать события кнопки, генерируемые мышью: `MouseClicked`, `MousePressed`, но надежнее использовать событие **ActionPerformed** (действие выполнено), поскольку «нажать» на кнопку можно как мышью, так и при помощи клавиатуры. Создайте обработчик **ActionPerformed** и добавьте в него код для вывода в текстовое поле сообщения:

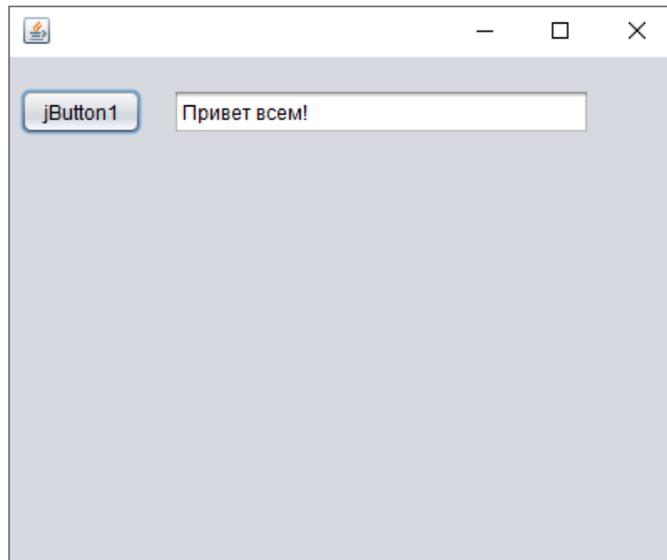


```

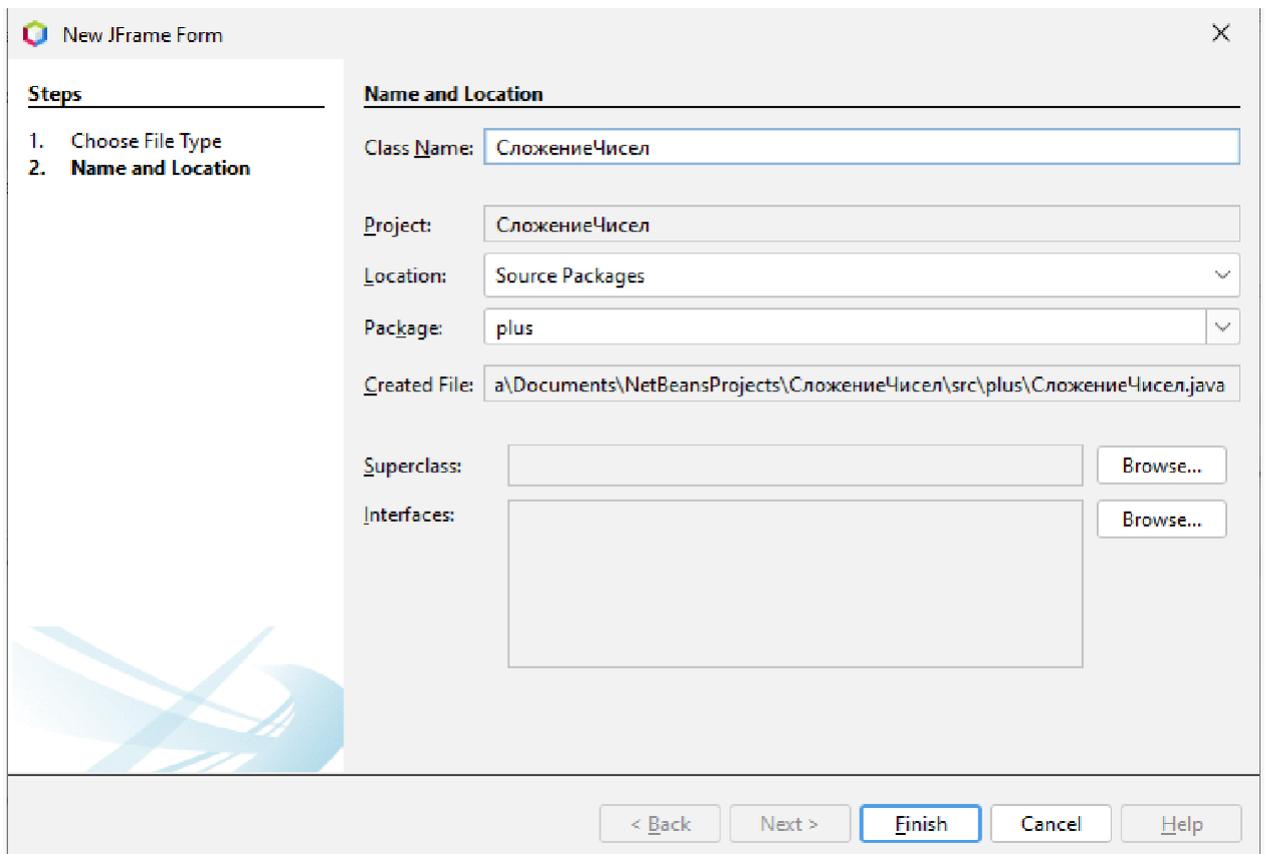
25     @SuppressWarnings ("unchecked")
26     Generated Code
66
67     private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
68         this.jTextField1.setText (t: "Привет всем!");
69     }
70
71

```

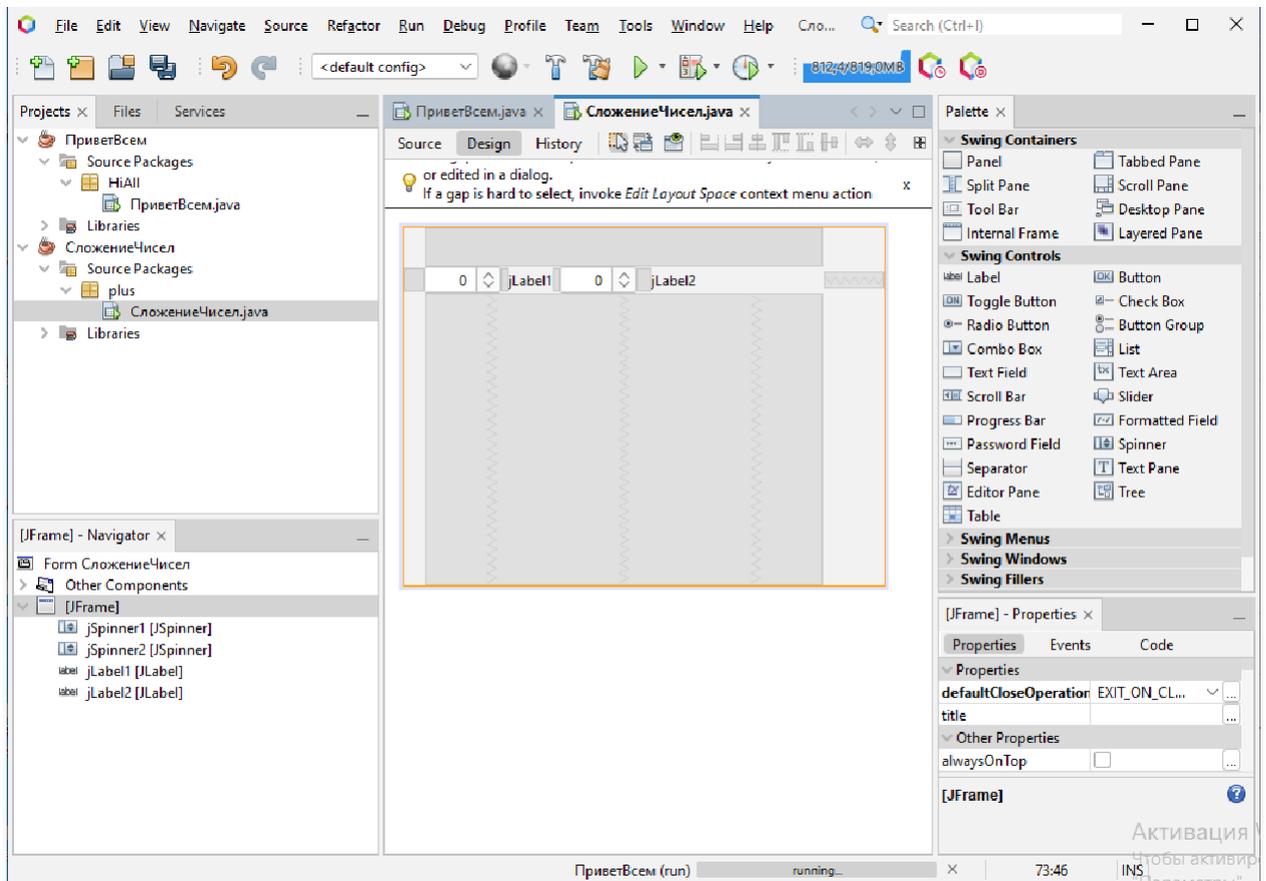
Запустите приложение и убедитесь, что при нажатии кнопки в текстовое поле выводится сообщение.



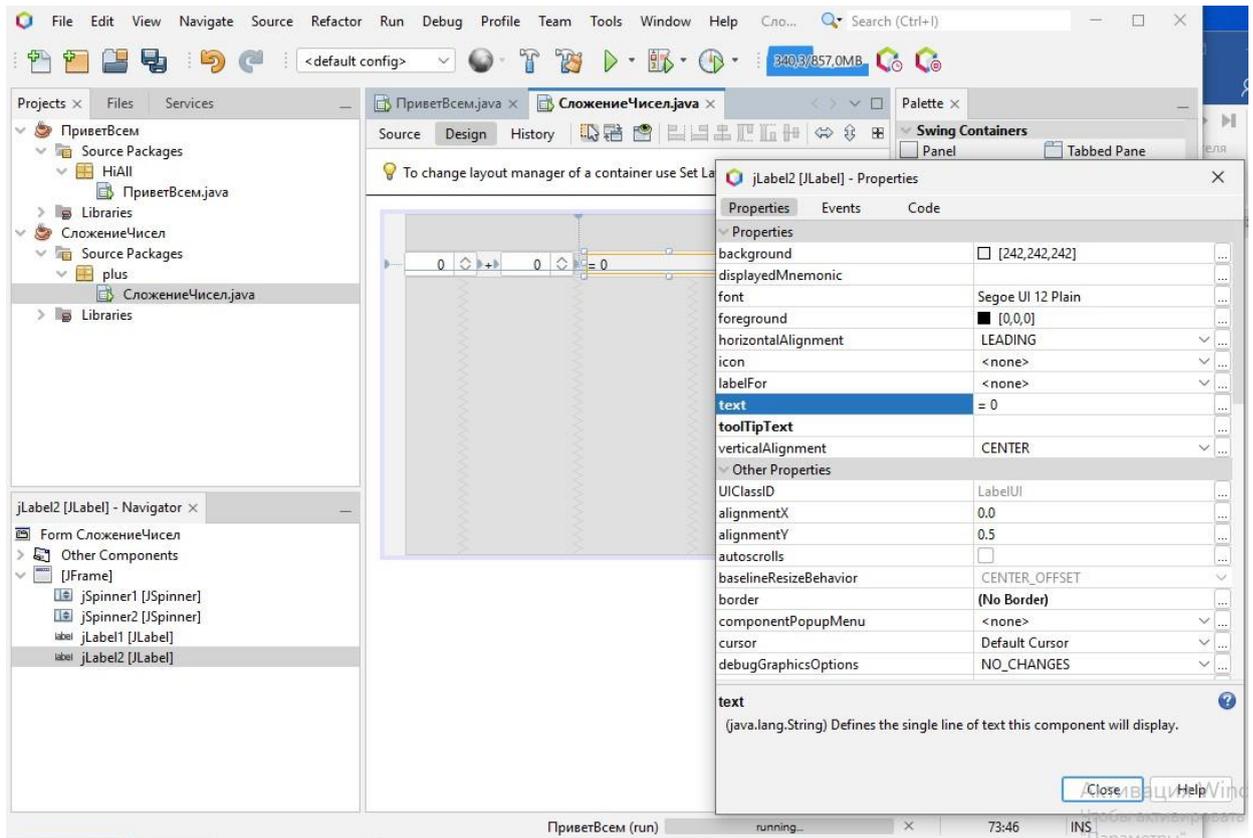
Создадим второе приложение – простейший калькулятор для сложения двух чисел. Назовите проект, например, «СложениеЧисел». При создании формы не забудьте указать для нее название, совпадающее с именем проекта:



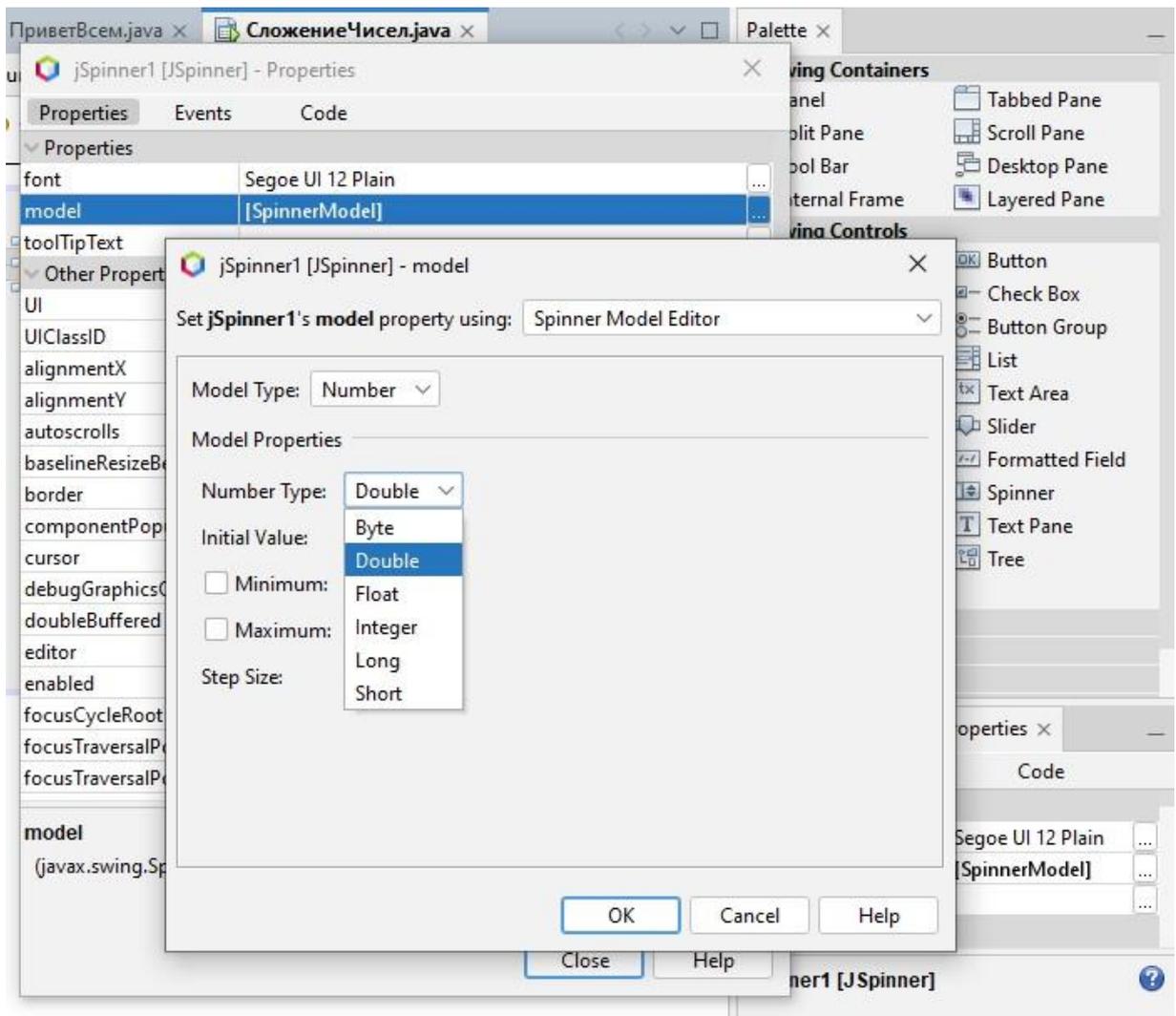
На форме разместим два объекта **Spinner** (для ввода слагаемых) и два объекта **Label** (один для отображения знака операции «плюс», другой – для отображения результата):



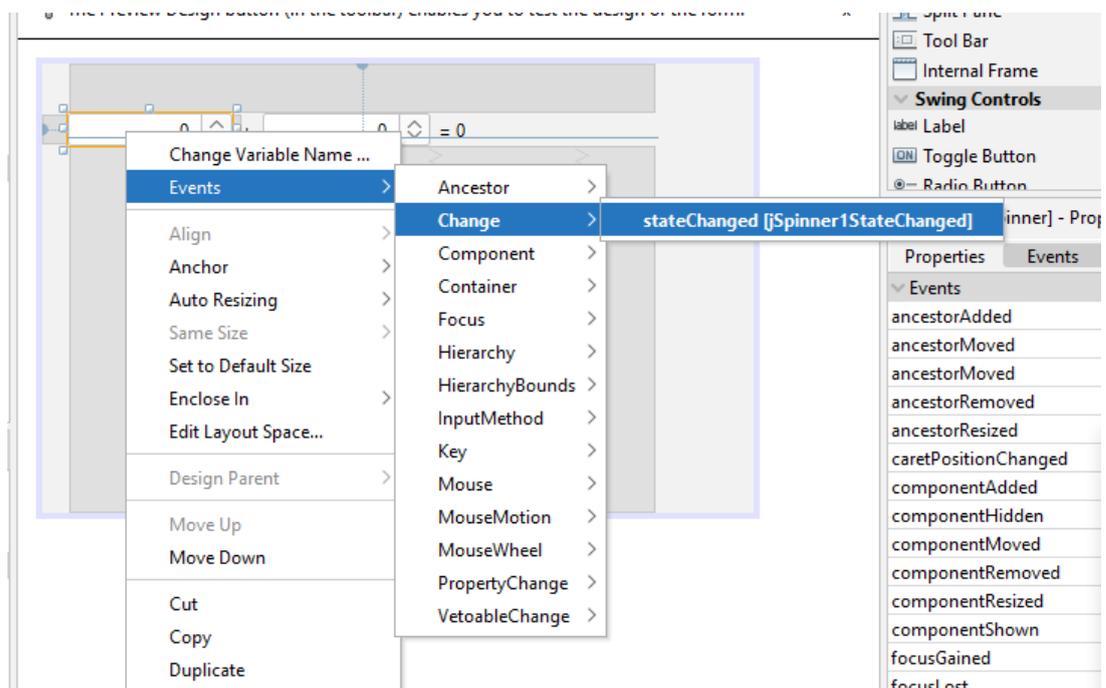
Далее «подпишем» наши надписи – **Label**:



И настроим спиннеры для ввода в них числовых значений типа double. Для этого используем свойство **model**:

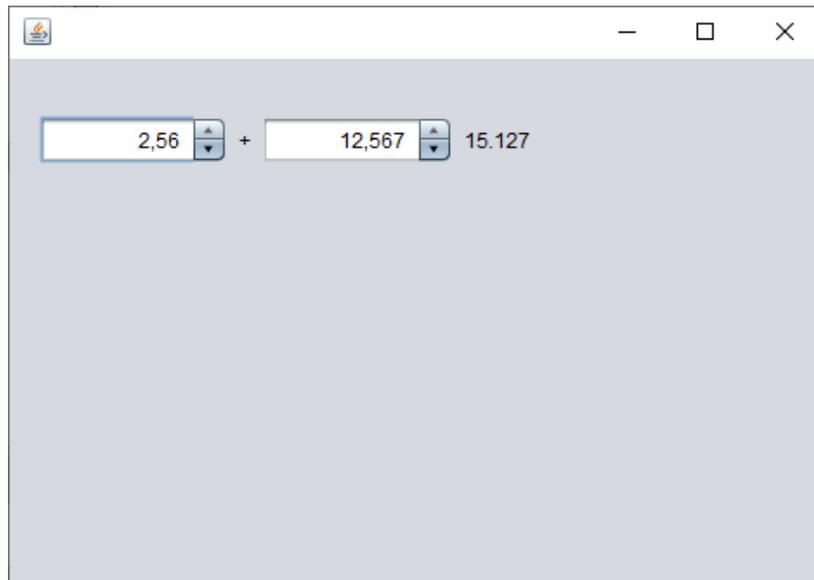


Установим числовой тип нашей модели данных для каждого спиннера **Number** и зададим свойство **Number Type** для ввода числовых значений двойной точности **Double**:

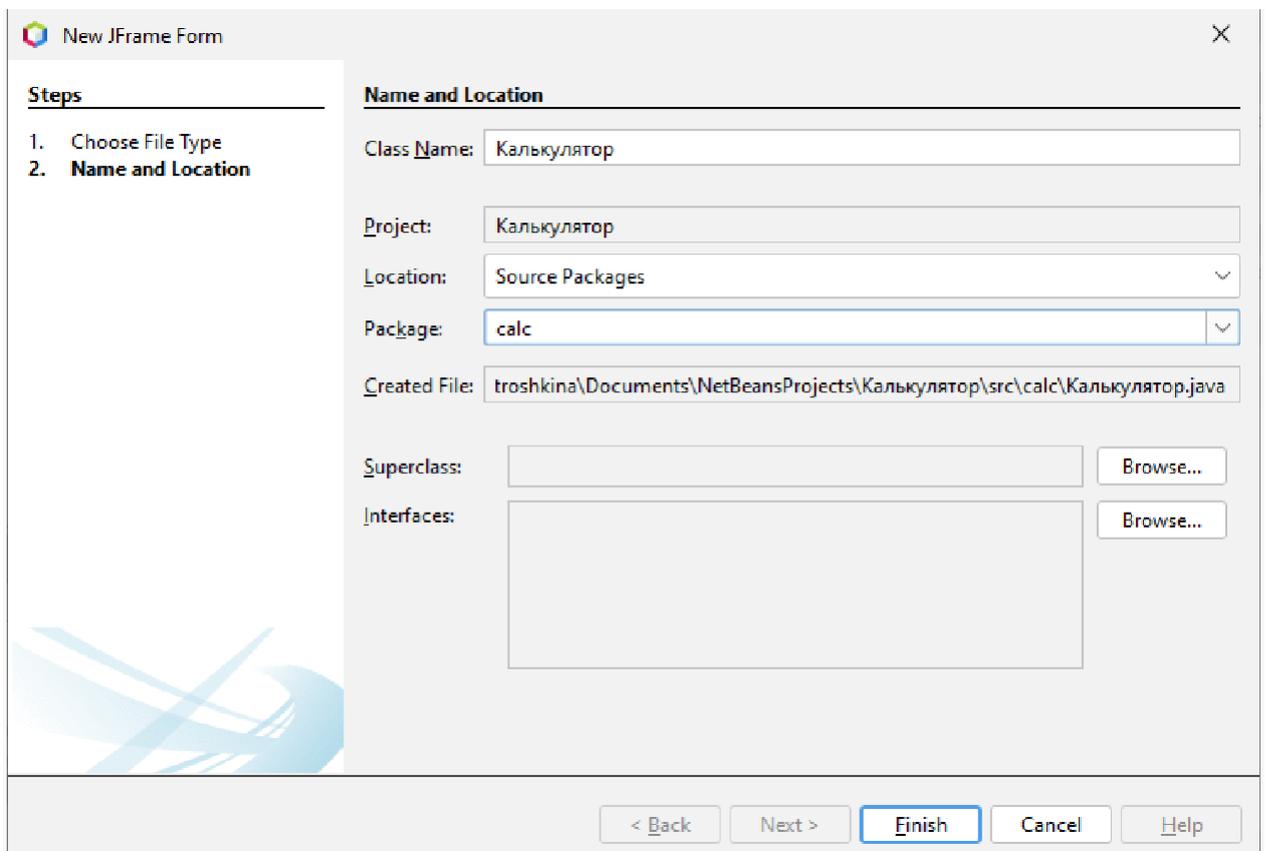


Для выполнения сложения используем событие спиннера **stateChanged**:

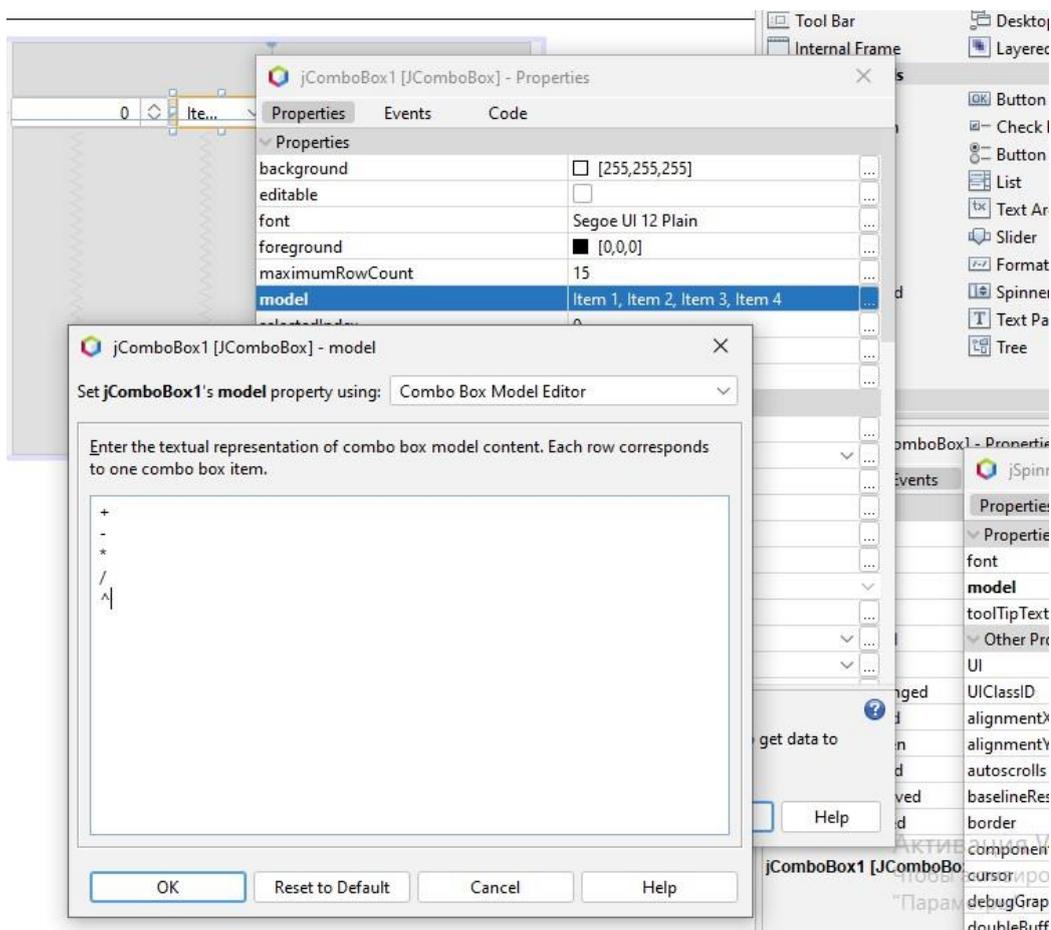
```
36 + | Generated Code |
35 |
36 | private void jSpinner1StateChanged(javax.swing.event.ChangeEvent evt) {
37 |     Double z = (Double) this.jSpinner1.getValue() + (Double) this.jSpinner2.getValue();
38 |     this.jLabel2.setText(text: z.toString());
39 | }
30 |
31 | private void jSpinner2StateChanged(javax.swing.event.ChangeEvent evt) {
32 |     Double z = (Double) this.jSpinner1.getValue() + (Double) this.jSpinner2.getValue();
33 |     this.jLabel2.setText(text: z.toString());
34 | }
35 |
36 | /**
```



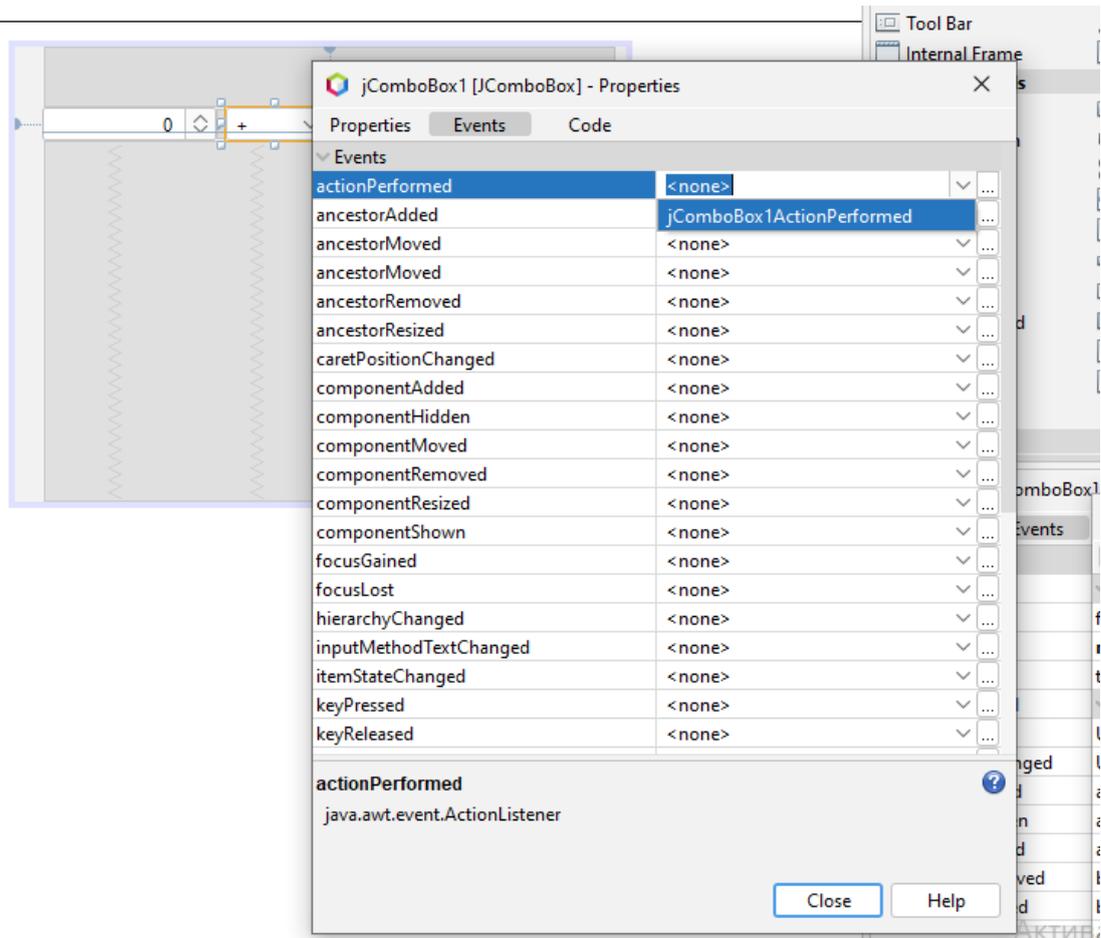
Разработайте самостоятельно приложение-калькулятор для вычисления всех арифметических операций. Для выбора знака операции примените выпадающий список.



Настройте содержание выпадающего списка при помощи его свойства model:



Для теста зададим событие – выполнение действия для выпадающего списка таким образом, чтобы при изменении операции в надписи отображалось сообщение о выполняемой операции:



Код обработчика:

```

25  @SuppressWarnings("unchecked")
26  Generated Code
78
79  @SuppressWarnings("StringEquality")
80  private void jComboBox1ActionPerformed(java.awt.event.ActionEvent evt) {
81      String s = "";
82      if (this.jComboBox1.getSelectedItem().toString() == "+")
83          s = "сложение";
84      else if (this.jComboBox1.getSelectedItem().toString() == "-")
85          s = "разность";
86      else if (this.jComboBox1.getSelectedItem().toString() == "*")
87          s = "умножение";
88      else if (this.jComboBox1.getSelectedItem().toString() == "/")
89          s = "деление";
90      else if (this.jComboBox1.getSelectedItem().toString() == "^")
91          s = "степень";
92      else
93          s = "";
94      this.jLabel1.setText(text: s);
95  }
96

```

Настройте код обработчика так, чтобы калькулятор вычислял арифметические операции с контролем возможных ошибок при делении на ноль и возведения отрицательного числа в дробную степень.

Назначение метода `clc()` для вычисления операций на события изменения состояния элементов управления:

```
private void jComboBox1ActionPerformed(java.awt.event.ActionEvent evt) {
    clc();
}
private void jSpinner1StateChanged(javax.swing.event.ChangeEvent evt) {
    clc(); // TODO add your handling code here:
}
private void jSpinner2StateChanged(javax.swing.event.ChangeEvent evt) {
    clc(); // TODO add your handling code here:
}
```

Метод для вычисления операций:

```
@SuppressWarnings({"UnusedAssignment", "StringEquality"})
private void clc()
{
    double z=0;
    double x = (double) this.jSpinner1.getValue();
    double y = (double) this.jSpinner2.getValue();
    if (this.jComboBox1.getSelectedItem().toString() == "+")
        z=x+y;
    else if (this.jComboBox1.getSelectedItem().toString() == "-")
        z=x-y;
    else if (this.jComboBox1.getSelectedItem().toString() == "*")
        z=x*y;
    else if (this.jComboBox1.getSelectedItem().toString() == "/")
    {
        if (y != 0)
            z=x/y;
        else
            z=0;
    }
    else if (this.jComboBox1.getSelectedItem().toString() == "^")
    {
        try
        {
            z = Math.pow(x,y);
        }
        catch (Exception e)
        {
            z = 0;
        }
    }
    else
        z=0;

    this.jLabel1.setText(" = " + String.valueOf(z));
}
```