

## **JAVA. Лабораторная работа №8**

### **Тема: Разработка графического пользовательского интерфейса (GUI) для редактирования списка записей**

Рассматривается вариант создания GUI на основе библиотеки Swing входящей в стандартную поставку языка Java.

В предлагаемом варианте для создания GUI используется GUI дизайнер Windows Builder.

Пример выполнения лабораторной работы:

Необходимо разработать GUI для ввода, просмотра и редактирования записей по одной. Каждая запись содержит следующие поля:

- 1 Название предмета;
- 2 ФИО преподавателя;
- 3 Количество лабораторных работ по предмету;
- 4 Количество успешно сданных лабораторных работ.

Необходимо обеспечить GUI для выполнения следующих операций:

- 1 Просмотр одной записи в один момент времени;
- 2 Переход к следующей записи (если она существует);
- 3 Переход к предыдущей записи (если она существует);
- 4 Удаление текущей записи;
- 5 Редактирование текущей записи;
- 6 Создание новой записи.

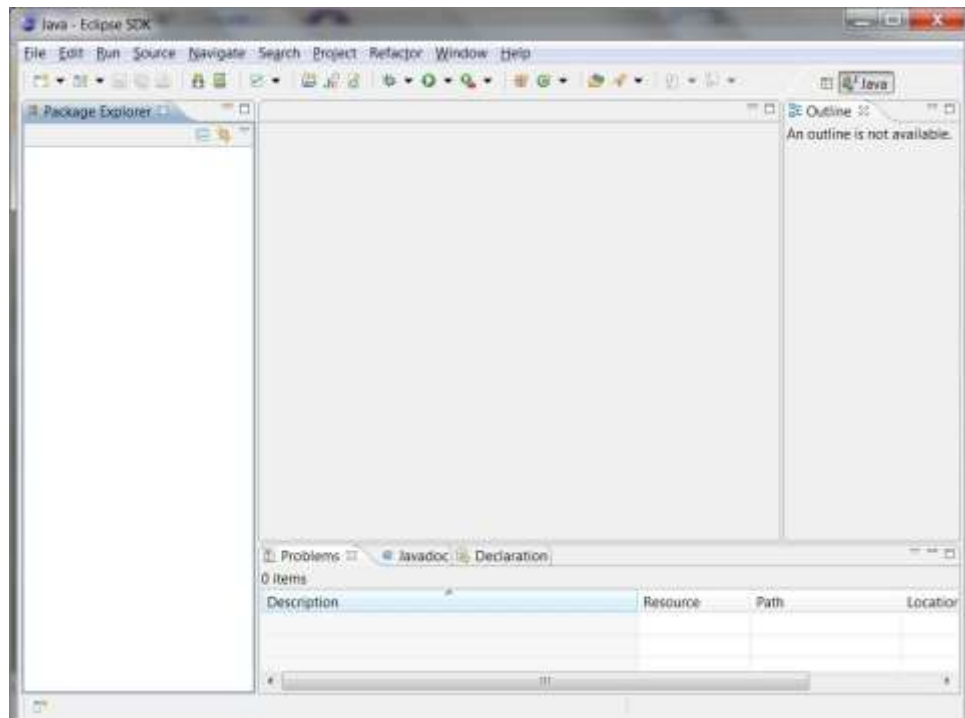
Пройдем поэтапно все этапы создания приложения с указанным функционалом.

На компьютере должен быть установлен Eclipse и GUI дизайнер Windows Builder.

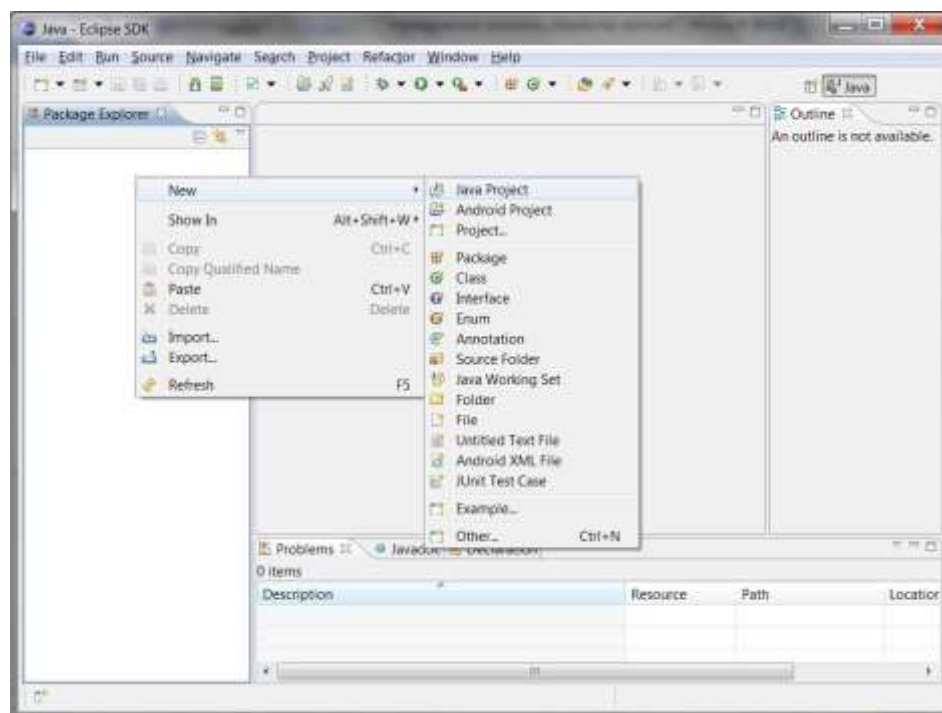
Шаги:

Создание приложения Swing.

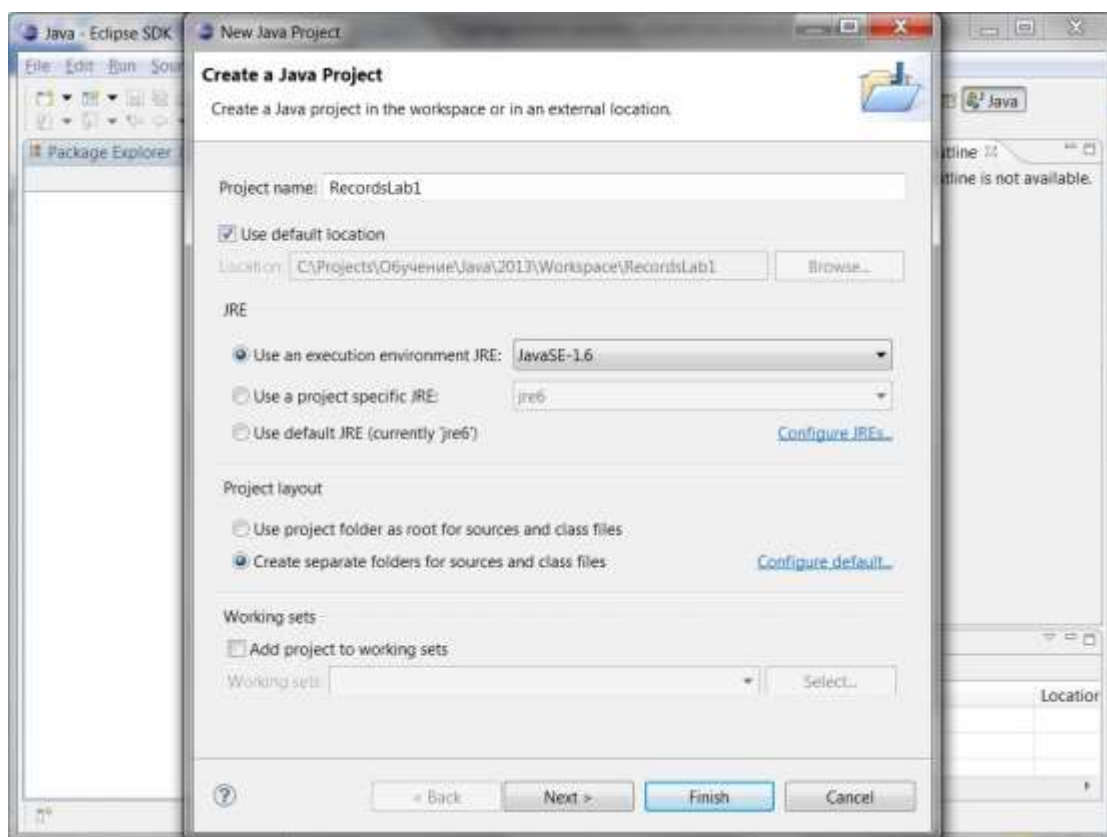
1 Запустить Eclipse. В итоге будет открыто окно примерно такого вида:



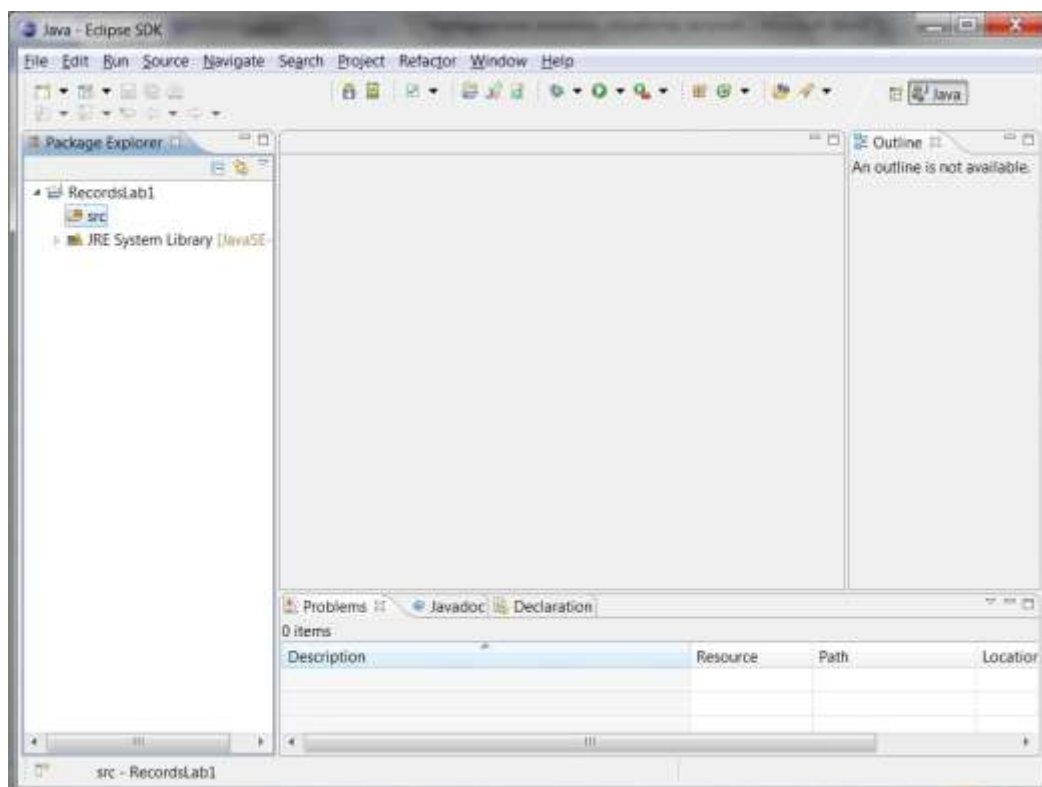
2 Нужно создать новый проект. Для этого нужно в окне Package Explorer кликнуть правой кнопкой мыши, после чего выбрать пункт меню New, а в выпавшем новом меню выбрать Java Project:



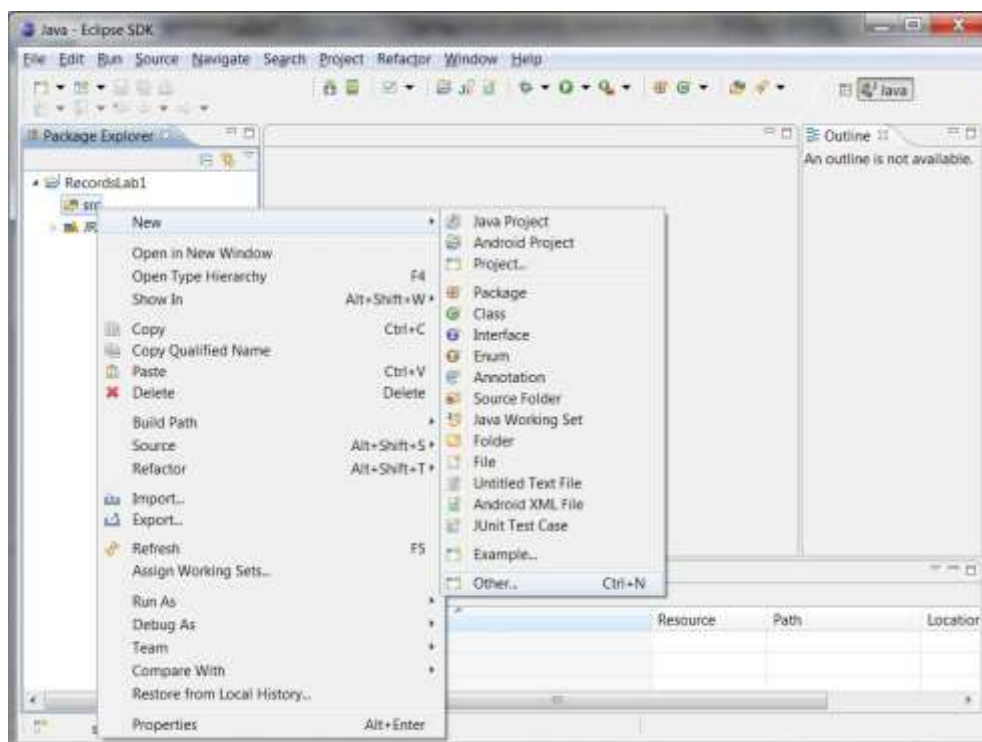
3 В открывшемся окне New Java Project нужно ввести имя проекта – в примере это RecordsLab1. После чего нажать кнопку Finish:



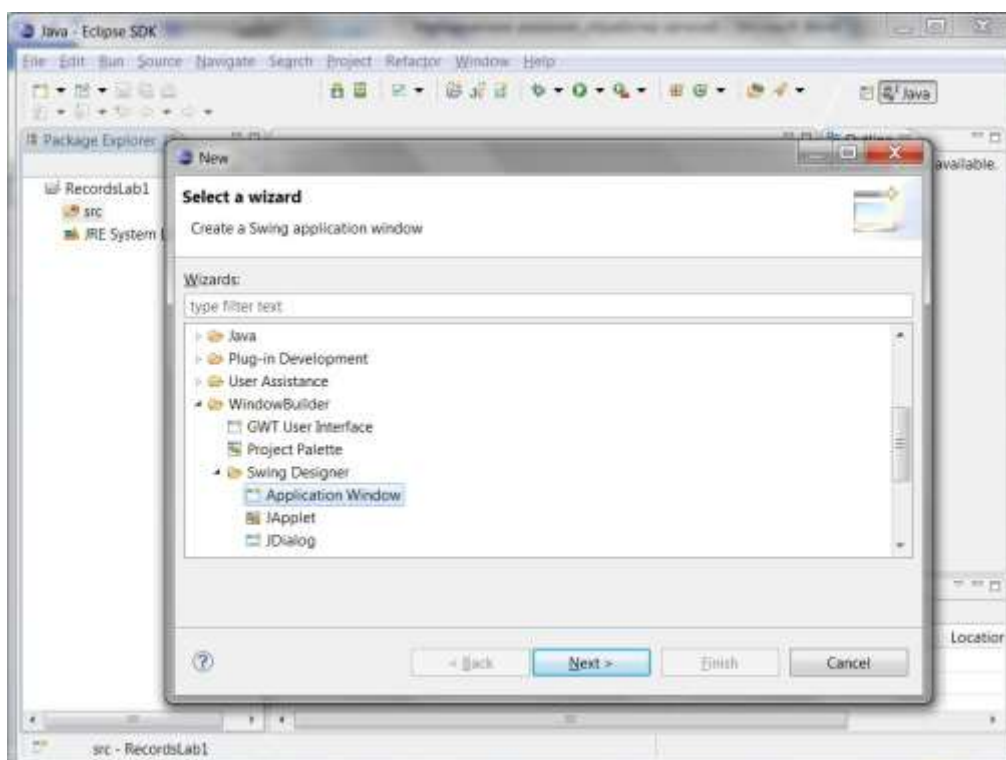
4 В итоге в окне Package Explorer появится созданный проект:



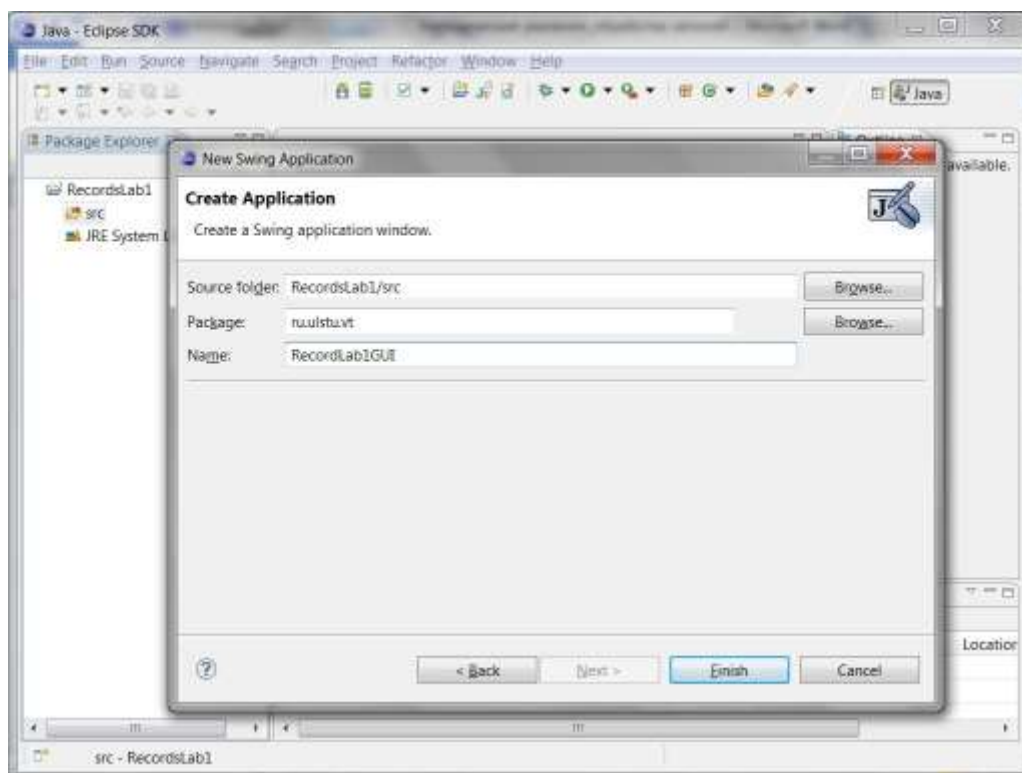
5 Теперь нужно создать форму для нашего проекта. Для этого нужно нажать правую кнопку мыши во вновь созданном проекте, после чего выбрать пункт меню New, а в выпавшем новом меню выбрать Other:



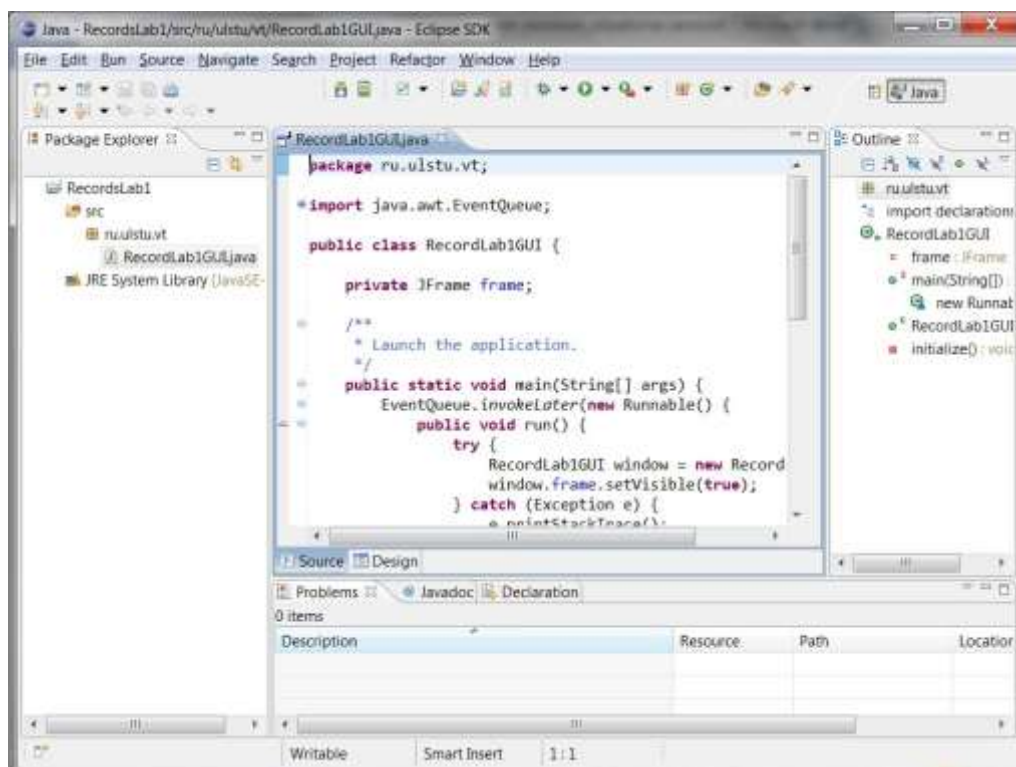
6 В открывшемся окне New нужно выбрать группу WindowBuilder, в ней группу Swing Designer и в ней пункт Application Window. После чего, для создания проекта, нужно нажать кнопку Next:



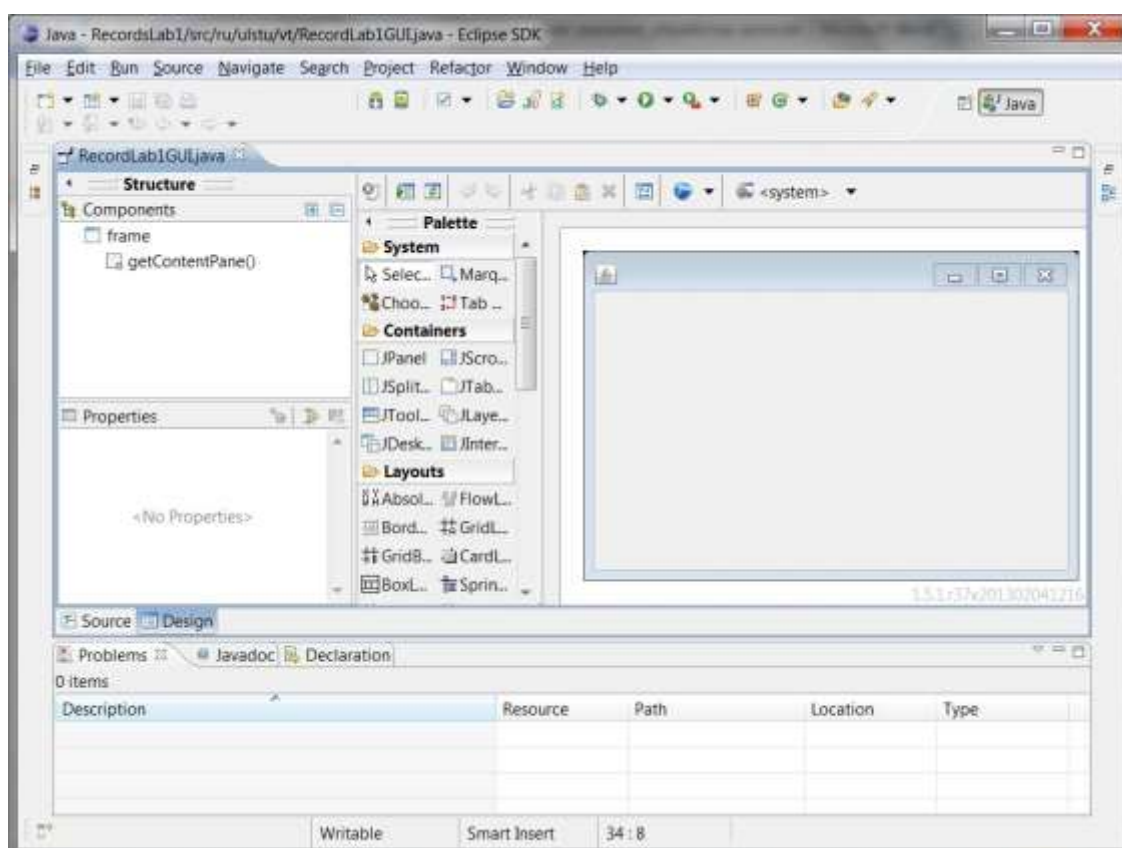
7 В появившемся окне New Swing Application нужно ввести имя пакета (в примере это ru.ulstu.vt) и имя класса окна приложения (в примере это RecordLab1GUI), после чего нажать кнопку Finish:



8 В итоге будет создан класс RecordLab1GUI и Eclipse примет такой вид:



9 Чтобы переключить Eclipse в режим визуального редактирования формы нужно нажать закладку Design:



Изучим сгенерированный код:

```
package ru.ulstu.vt;
import java.awt.EventQueue;
public class RecordLab1GUI {
    private JFrame frame;
    /**
     * Launch the application.
     */
    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    RecordLab1GUI window = new RecordLab1GUI(); window.frame.setVisible(true);
                } catch (Exception e) { e.printStackTrace();
                }
            }
        });
    }
    /**
     * Create the application.
     */
    public RecordLab1GUI() { initialize();
    }
```

```

/**
 * Initialize the contents of the frame.
 */
private void initialize() {
    frame = new JFrame();
    frame.setBounds(100, 100, 450, 300);
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
}
}

```

Как видно код минимален. Создан класс **class** RecordLab1GUI, внутри которого есть главное окно приложения **private** JFrame frame;

В статическом методе main в отдельном потоке создается экземпляр класса RecordLab1GUI, после чего он делается видимым.

```

public static void main(String[] args) {
    EventQueue.invokeLater(new Runnable() {
        public void run() {
            try {
                RecordLab1GUI window = new RecordLab1GUI(); window.frame.setVisible(true);
            } catch (Exception e) { e.printStackTrace(); }
        }
    });
}

```

При создании экземпляра класса вызывается конструктор

```

public RecordLab1GUI() {
    initialize();
}

```

Который в свою очередь вызывает метод initialize(); Который в минимальной версии имеет вид:

```

private void initialize() {
    frame = new JFrame();
    frame.setBounds(100, 100, 450, 300);
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
}

```

В этой версии во время инициализации создается JFrame

```
frame = new JFrame();
```

задаются размеры окна

```
frame.setBounds(100, 100, 450, 300);
```

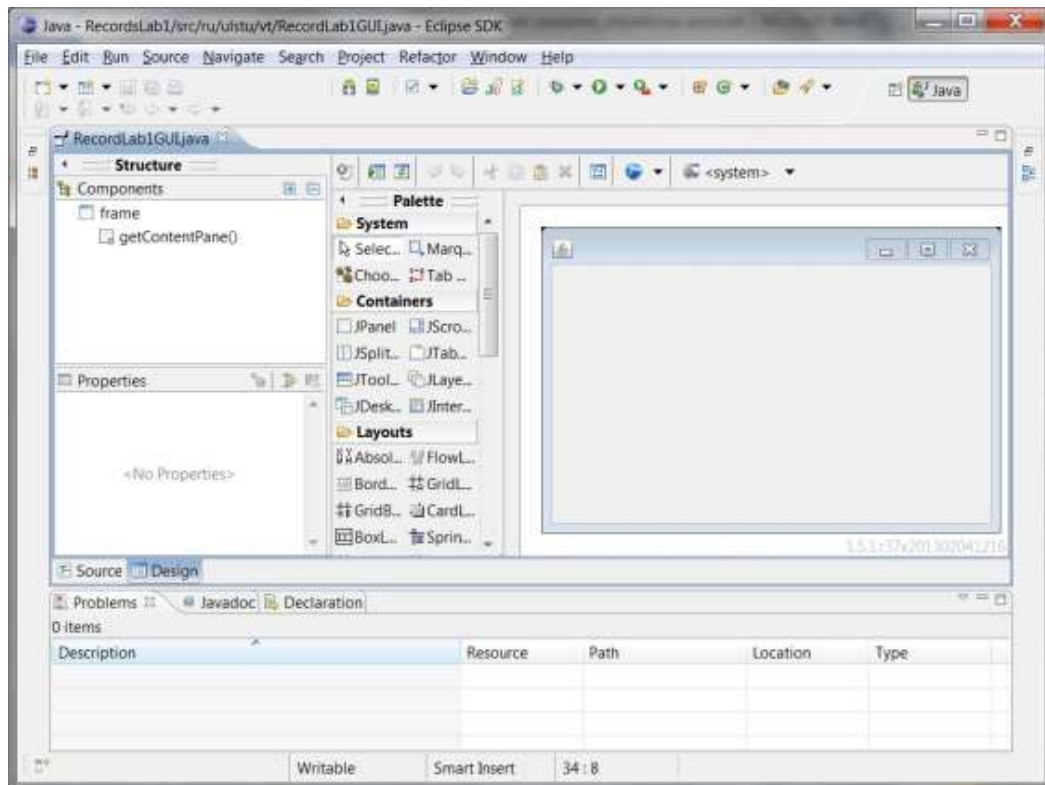
и задается дефолтное поведение окна при нажатии кнопки «Заккрыть окно»

```
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

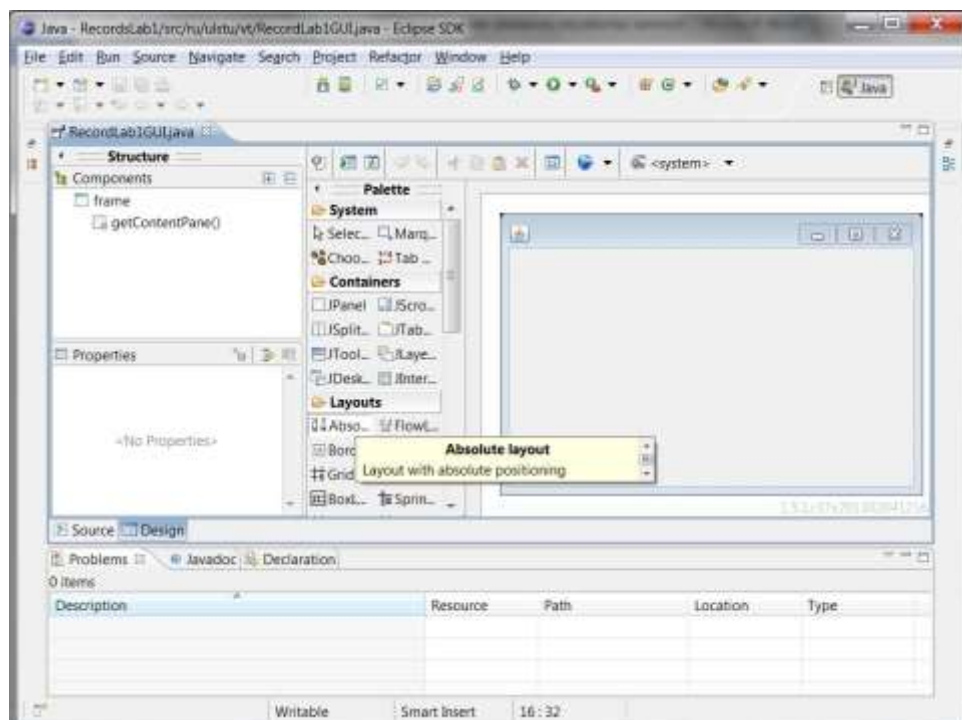


## 1 Замена менеджера размещения на абсолютное (ручное) размещение.

Мы будем добавлять элементы в режиме визуального редактирования. Поэтому переключим Eclipse в этот режим:

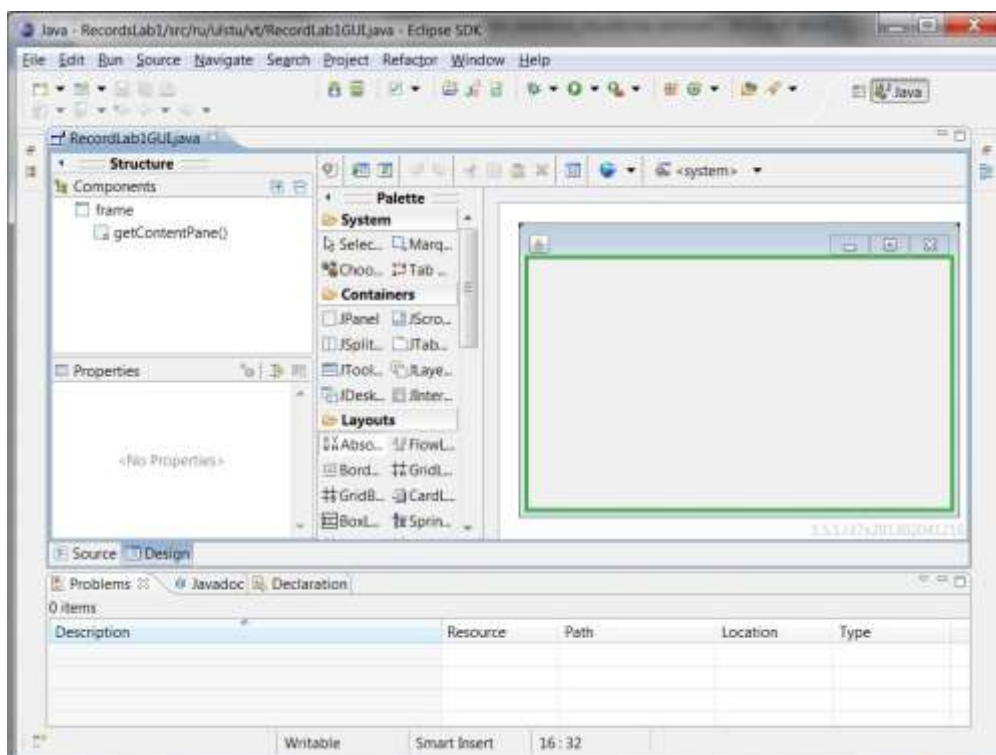


1 На панели Palette выберем инструмент Absolute layout. Он нужен нам для того, чтобы создавать форму с фиксированным размещением элементов. Чтобы выбрать инструмент нужно сделать клик левой кнопкой мыши по инструменту:

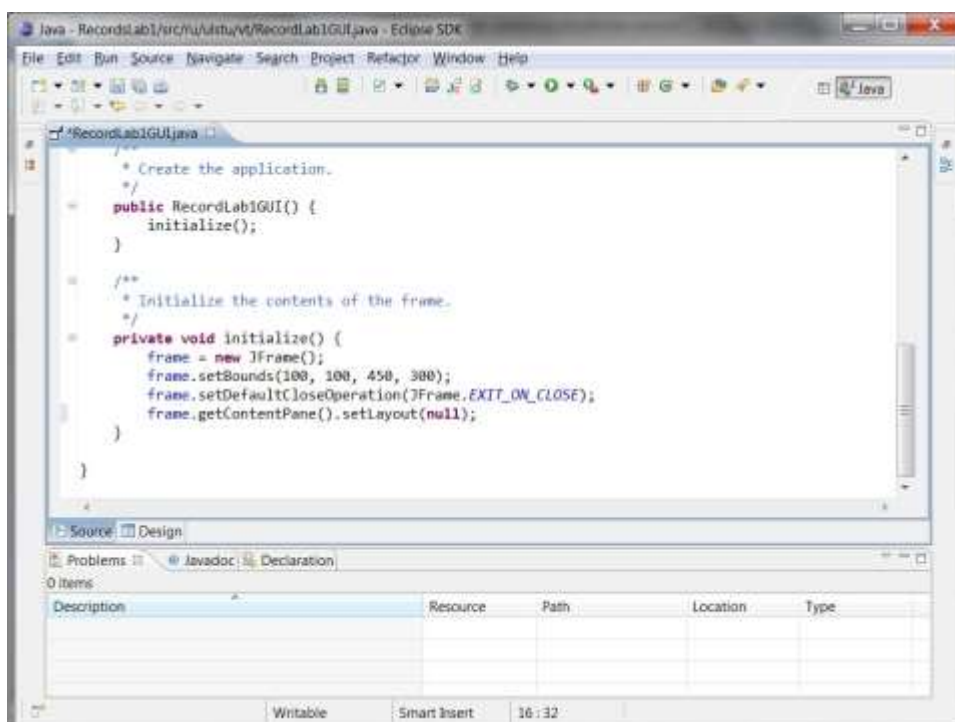




2 Теперь нужно курсор мыши переместить в редактируемое окно. При этом рабочая область окна, к которой будет применен новый layout, выделиться зеленой рамкой:



3 После того как будет произведен клик левой кнопкой мыши по рабочей области окна, зеленая рамка исчезнет. Если переключиться в режим редактирования кода и изучить код функции initialize, то вы обнаружите, что к ранее сгенерированному коду добавилась еще одна строка:



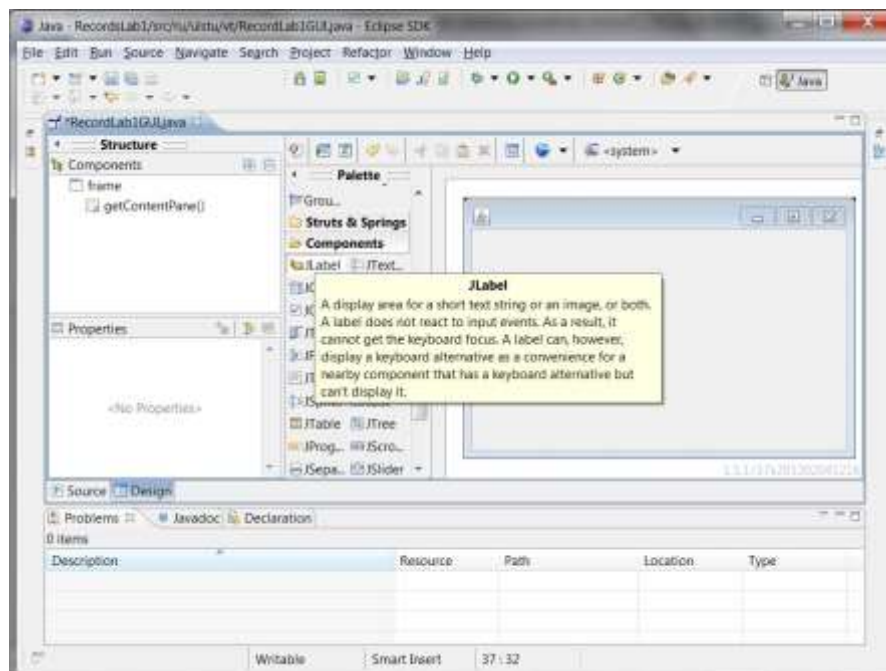
```
frame.getContentPane().setLayout(null);
```

означает, что теперь никакие layout (менеджеры размещения элементов) применяться не будут, а размещаемые элементы нужно будет размещать вручную.

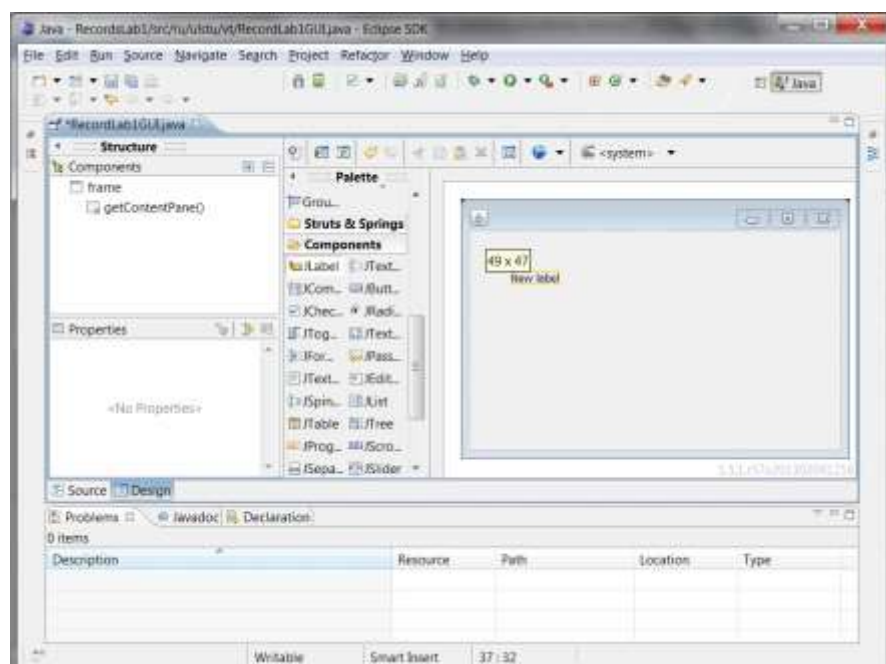
## 2 Добавление на форму метки.

Для каждого поля мы будем на форму добавлять 2 элемента – метку, с текстом, поясняющим смысл текста, и элемент редактирования поля с собственно текстом. Сначала добавим метку.

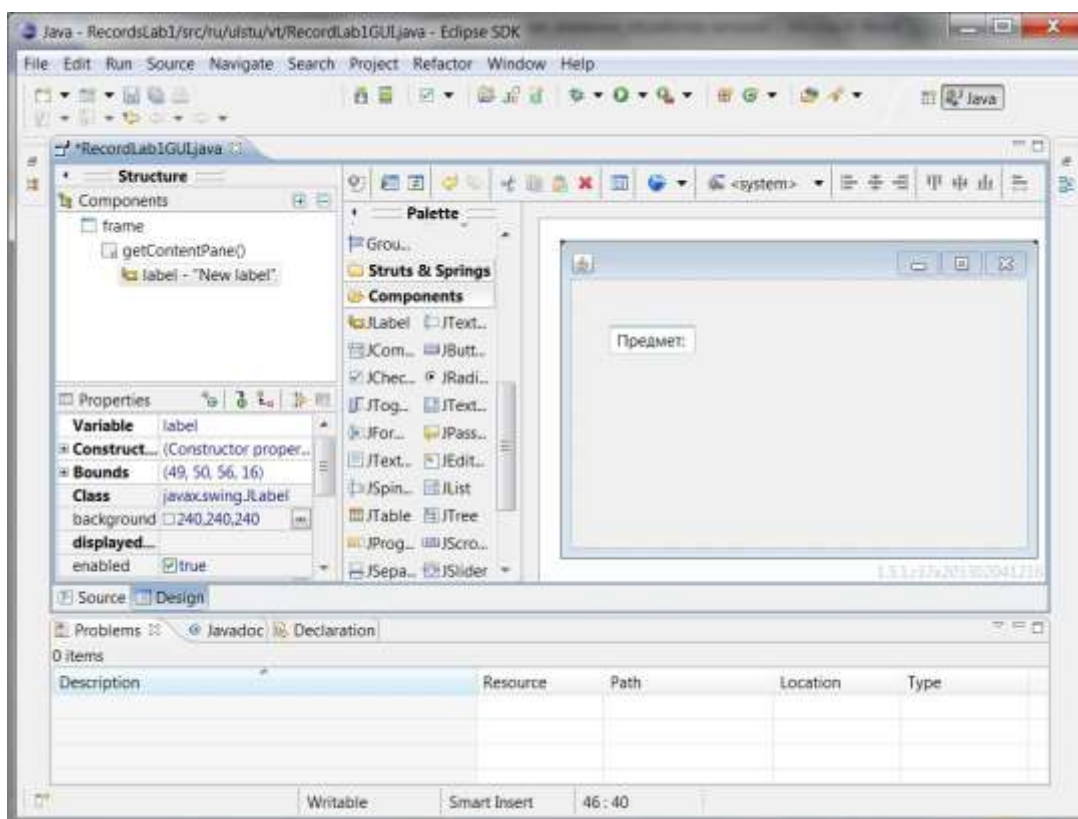
1 На панели Palette выберем компонент JLabel – для этого кликнем по нему левой кнопкой мыши:



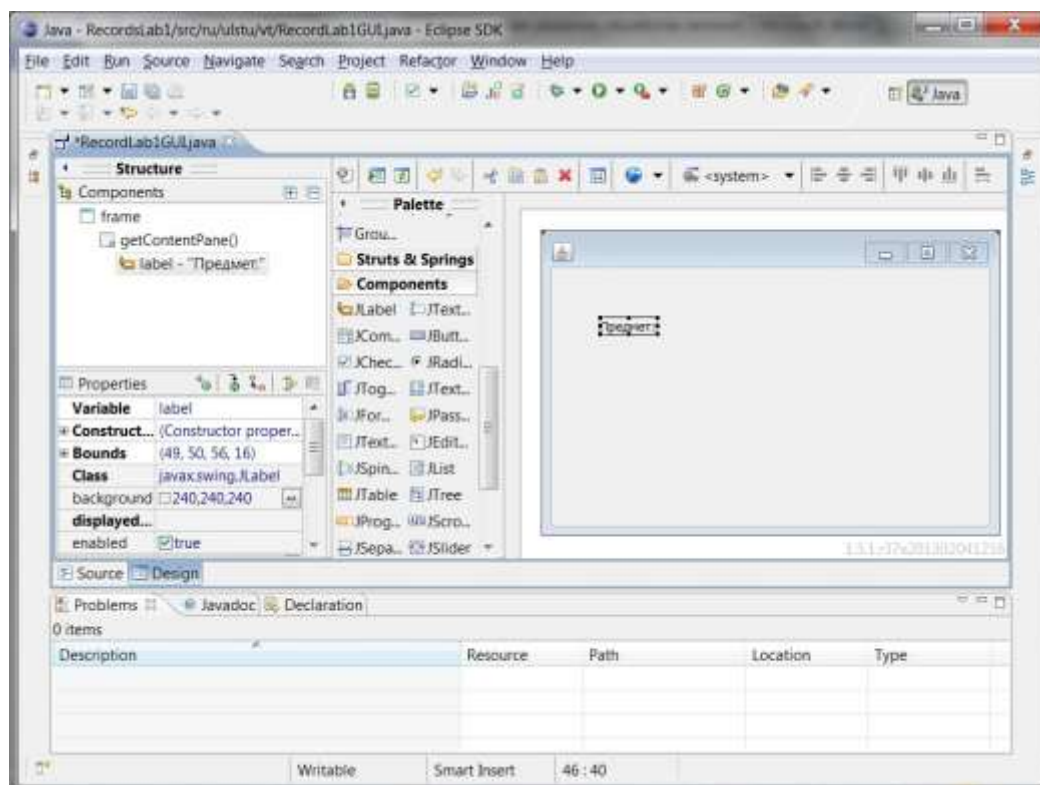
2 При помощи курсора мыши выберем место для метки на форме:



3 После фиксации элемента требуется ввести текст. В данном примере текстом будет «Предмет:»



4 После нажатия Enter текст зафиксировался и форма примет следующий вид:



```
private void initialize() {  
    frame = new JFrame();  
    frame.setBounds(100, 100, 450, 300);  
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
    frame.getContentPane().setLayout(null);  
    JLabel label = new JLabel("\u041F\u0440\u0435\u0434\u0435\u0434\u0442");  
    label.setBounds(49, 50, 56, 16);  
    frame.getContentPane().add(label);  
}
```

```
JLabel label = new JLabel("\u041F\u0440\u0435\u0434\u0435\u0435\u0442:");
label.setBounds(49, 50, 56, 16);
frame.getContentPane().add(label);
```

```
JLabel label = new JLabel("\u041F\u0440\u0435\u0434\u0435\u0434\u0442.");
```

Строка

Задаёт размещение и границы метки.

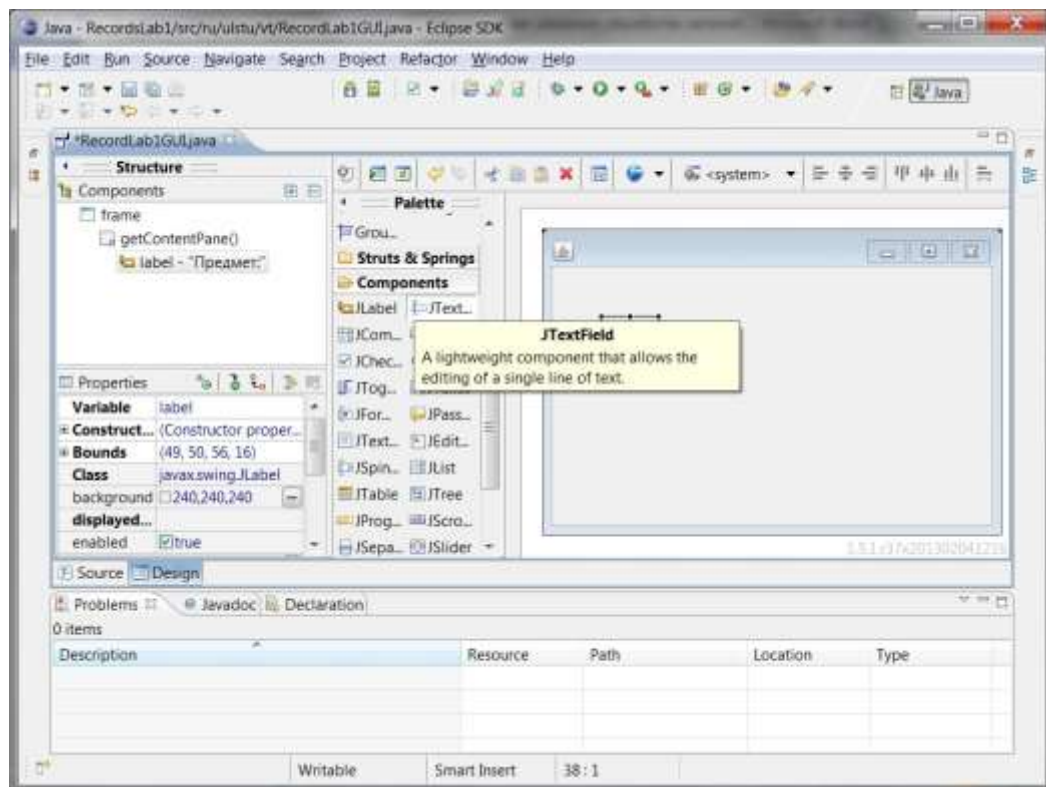
```
frame.getContentPane().add(label);
```

```
frame.getContentPane().setLayout(null);
```

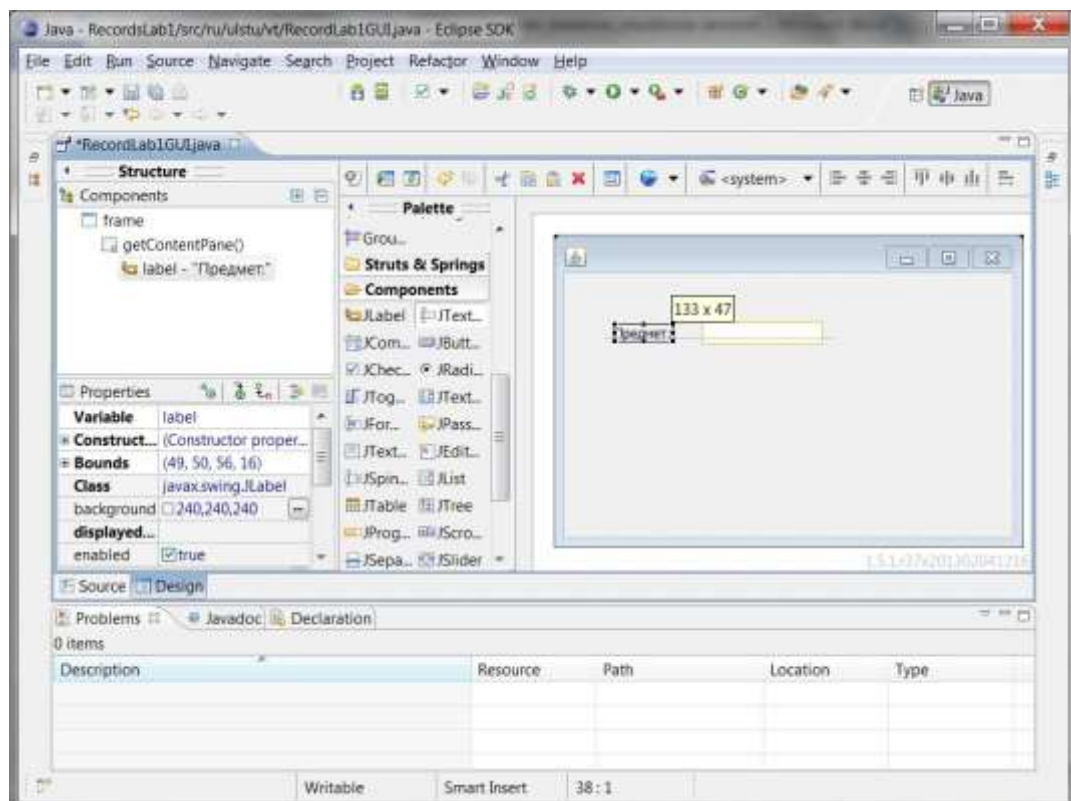
```
label.setBounds(49, 50, 56, 16);
```

### 3 Добавление на форму элемента редактирования поля.

1 На панели Palette выберем компонент JTextField – для этого кликнем по нему левой кнопкой мыши:



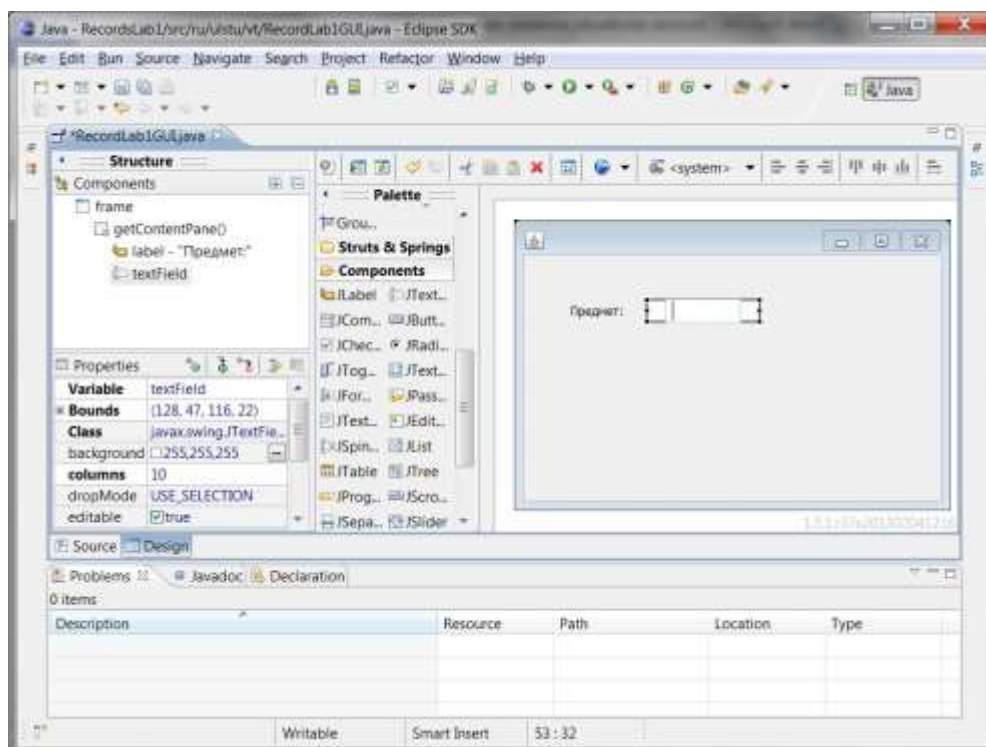
2 При помощи курсора мыши выберем место для элемента на форме:



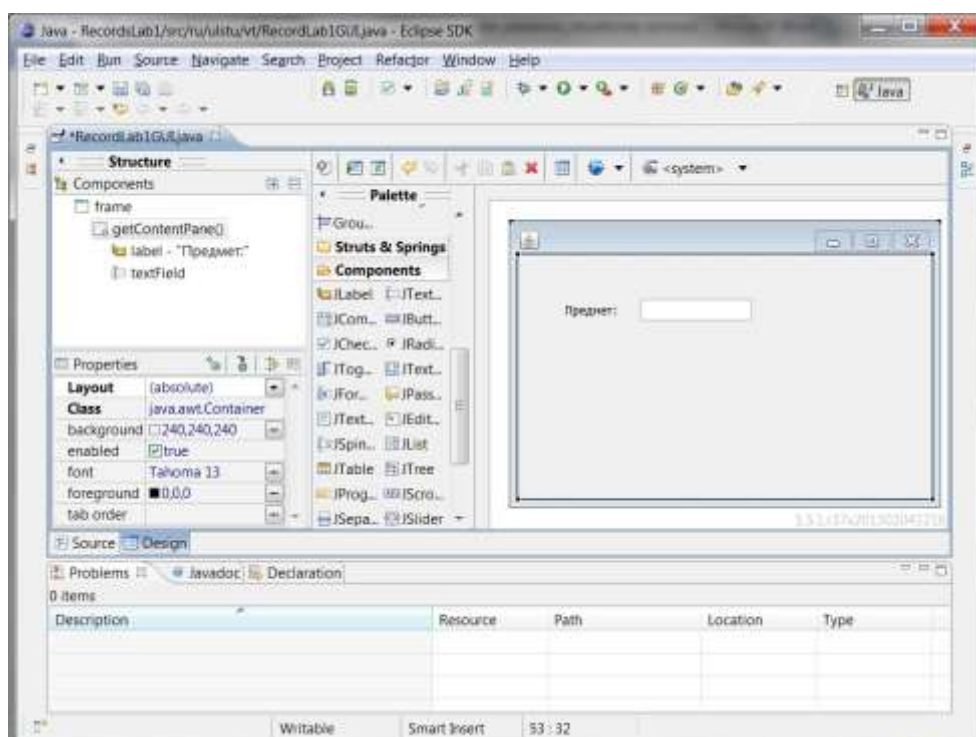


Обратите внимание на вертикальное выравнивание текста метки и текста добавляемого элемента – их нужно выравнивать так, как показано на рисунке – должна появиться линия подчеркивающая текст в метке.

3 После добавления элемента форма примет такой вид:

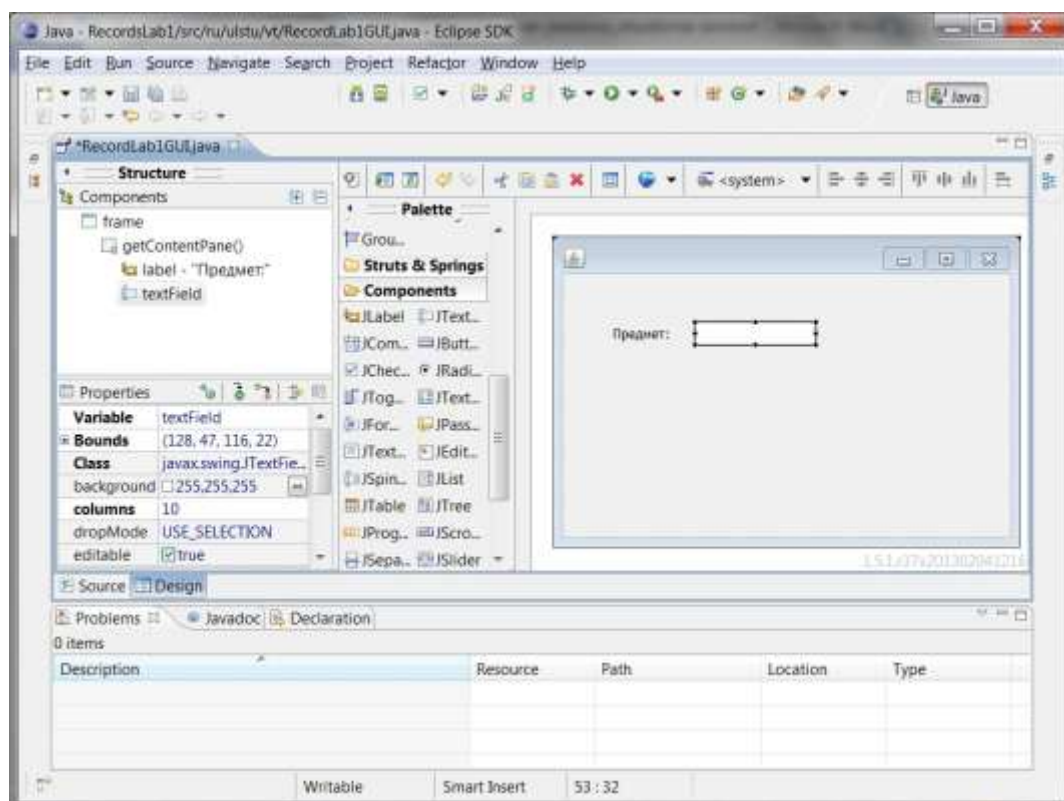


4 Чтобы изменить местоположение элемента и его размеры нужно выйти из режима ввода текста. Для этого можно нажать Esc, либо изменить фокус ввода, щелкнув в любой другой элемент. В итоге форма приобретет такой вид:

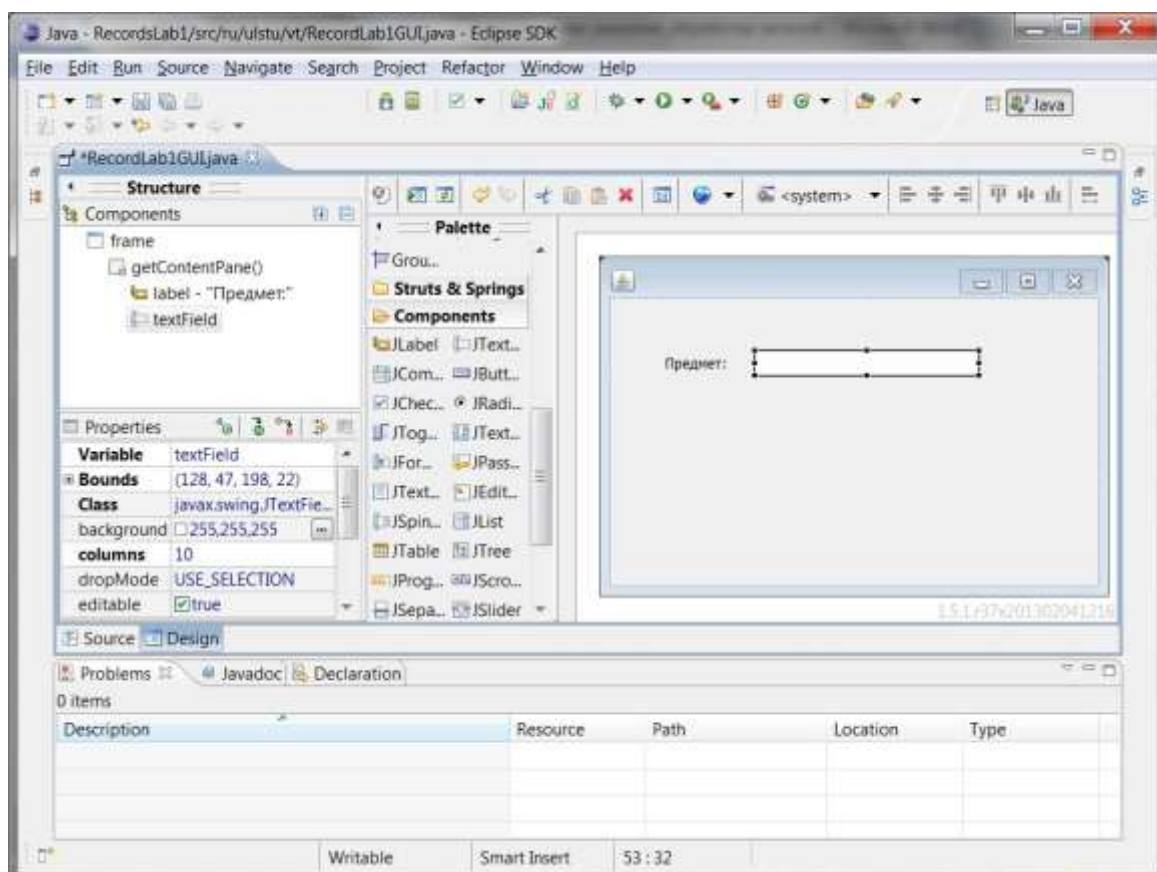




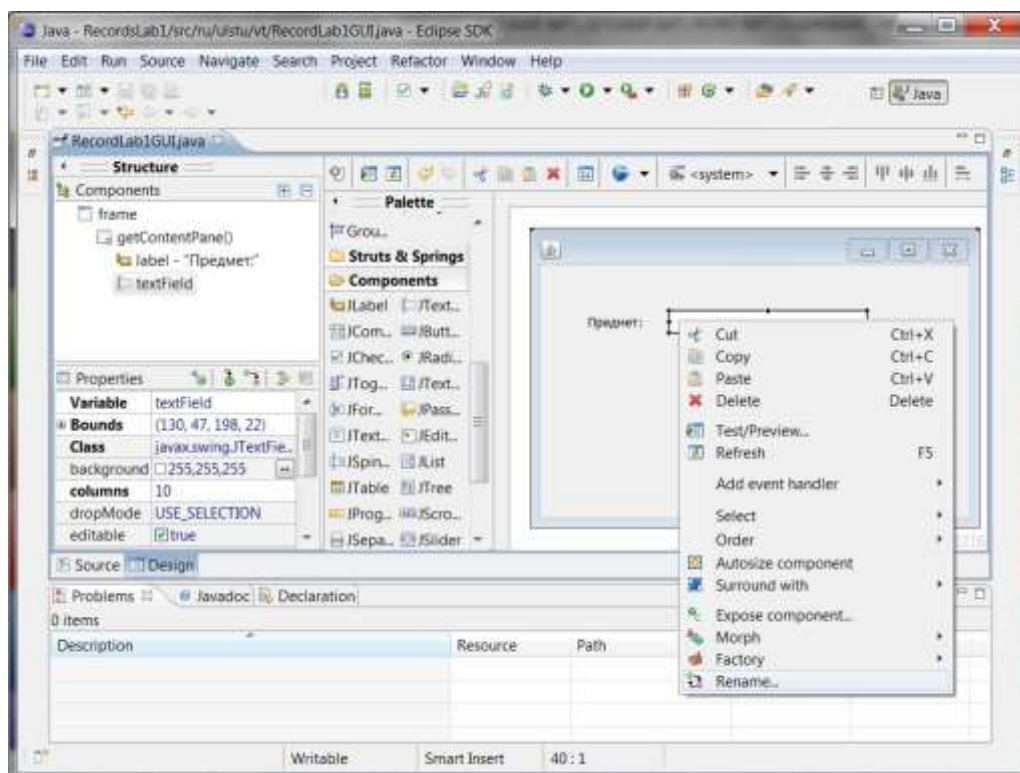
5 Чтобы изменить местоположение элемента нужно щелкнуть по нему левой кнопкой:



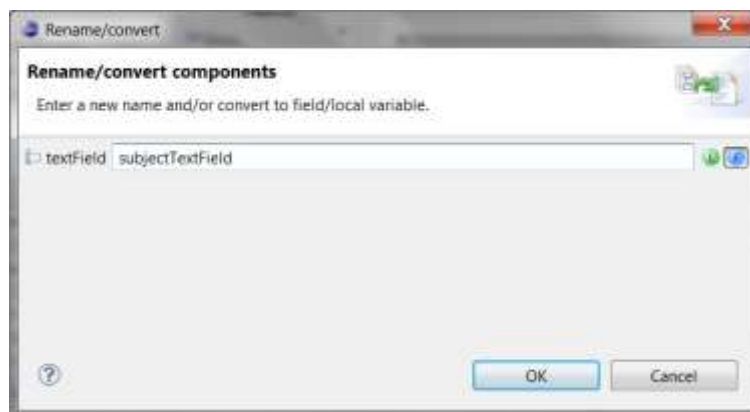
6 После этого можно изменять местоположение и размеры элемента:



7 Элементу редактирования нужно задать осмысленное имя, чтобы впоследствии в коде его можно было легко использовать. Для этого нужно кликнуть элемент правой кнопкой мыши и в меню выбрать пункт Rename:



8 В открывшемся окне вводим новое имя. В данном примере новое имя subjectTextField:



Посмотрим на изменения кода теперь.

Во-первых в классе RecordLab1GUI добавилось еще одно приватное поле subjectTextField:

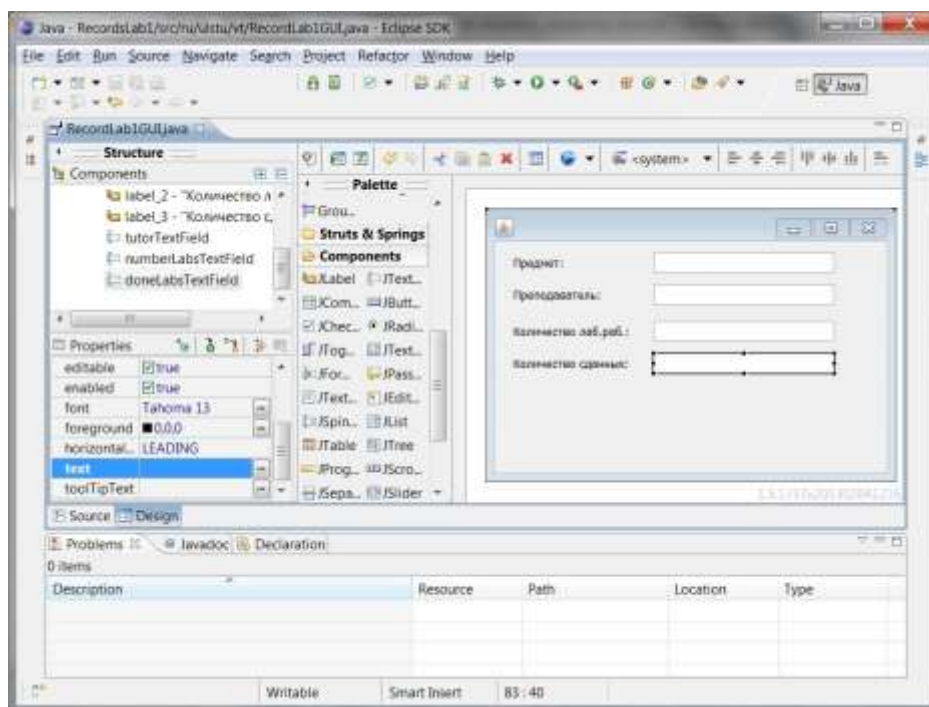
```
public class RecordLab1GUI {  
    private JFrame frame;  
    private JTextField subjectTextField;
```

Во-вторых в метод добавился код, отвечающий за создание и инициализацию нового элемента:

```
private void initialize() {  
    frame = new JFrame();  
    frame.setBounds(100, 100, 450, 300);  
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
    frame.getContentPane().setLayout(null);  
    JLabel label = new Label("\u041f\u0440\u0435\u0434\u043c\u0435\u0442\u044b");  
    label.setBounds(49, 50, 56, 16);  
    frame.getContentPane().add(label);  
    subjectTextField = new JTextField(); subjectTextField.setBounds(130, 47, 198, 22);  
    frame.getContentPane().add(subjectTextField); subjectTextField.setColumns(10);  
}
```

#### 4 Добавление на форму элементов для оставшихся полей.

1 Применяя шаги аналогичные разделам III и IV нужно добавить метки и элементы редактирования для всех полей. В итоге получится такой вид формы:



Код класса также изменится. В частности будут добавлены приватные поля для всех элементов редактирования:

```
public class RecordLab1GUI {  
    private JFrame frame;  
    private JTextField subjectTextField; private JTextField tutorTextField; private  
    JTextField numberLabsTextField; private JTextField doneLabsTextField;
```

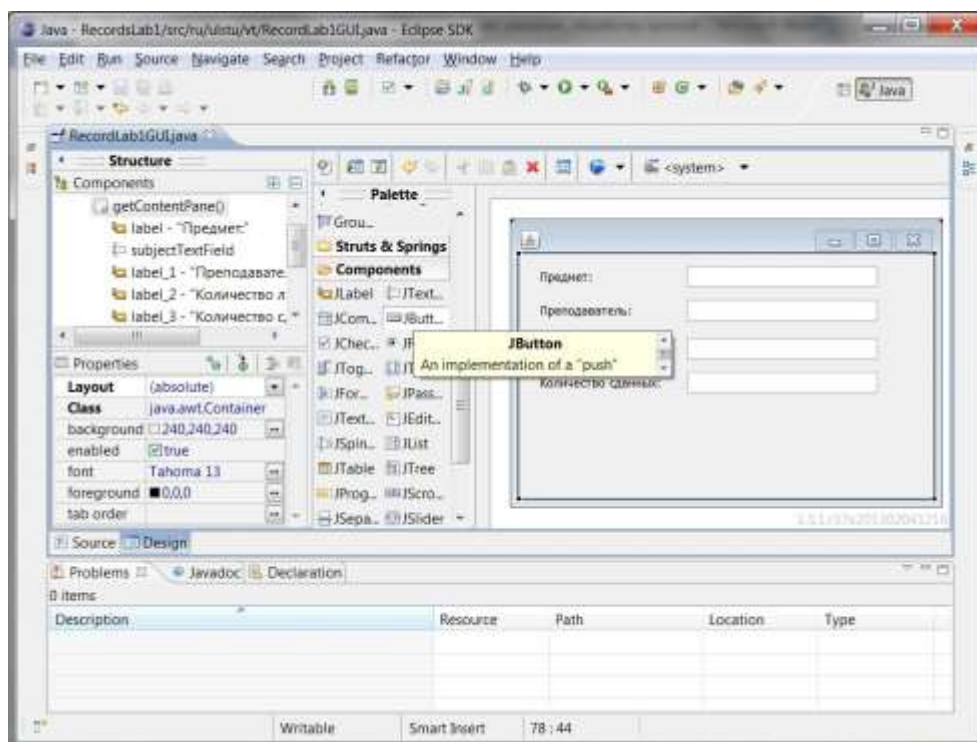
В методе `initialize` появится код, который обеспечивает создание всех элементов и их инициализацию:

[illegible]

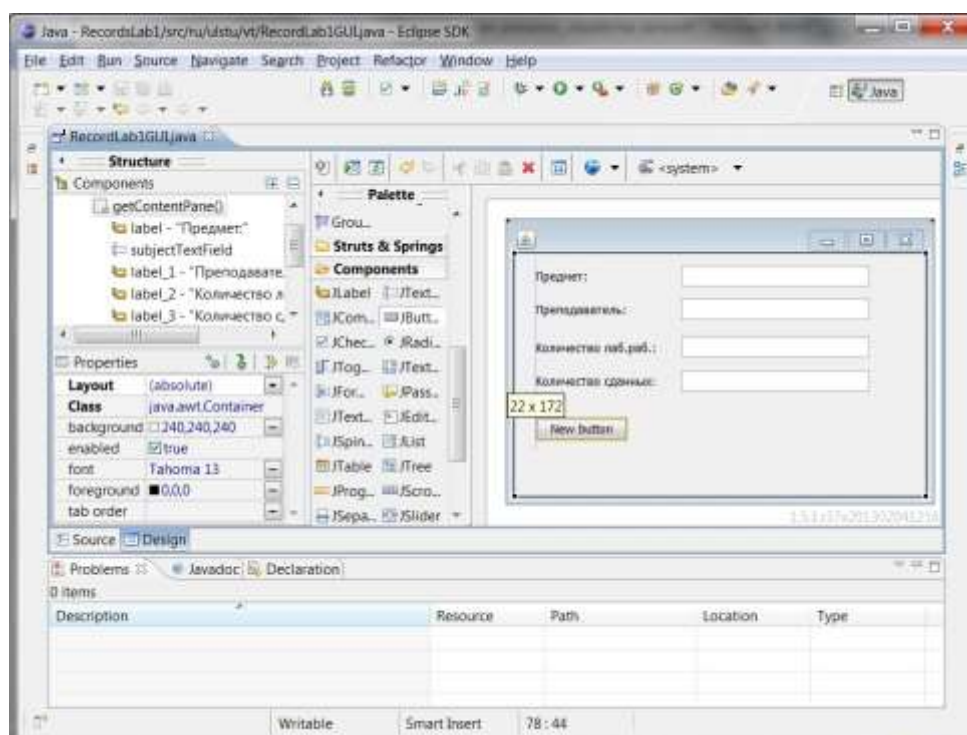
## 5 Добавление на форму кнопки.

Для управления нам потребуется несколько кнопок на форме. Добавим первую из них – «Принять изменения»

1 На панели Palette выбираем элемент JButton:

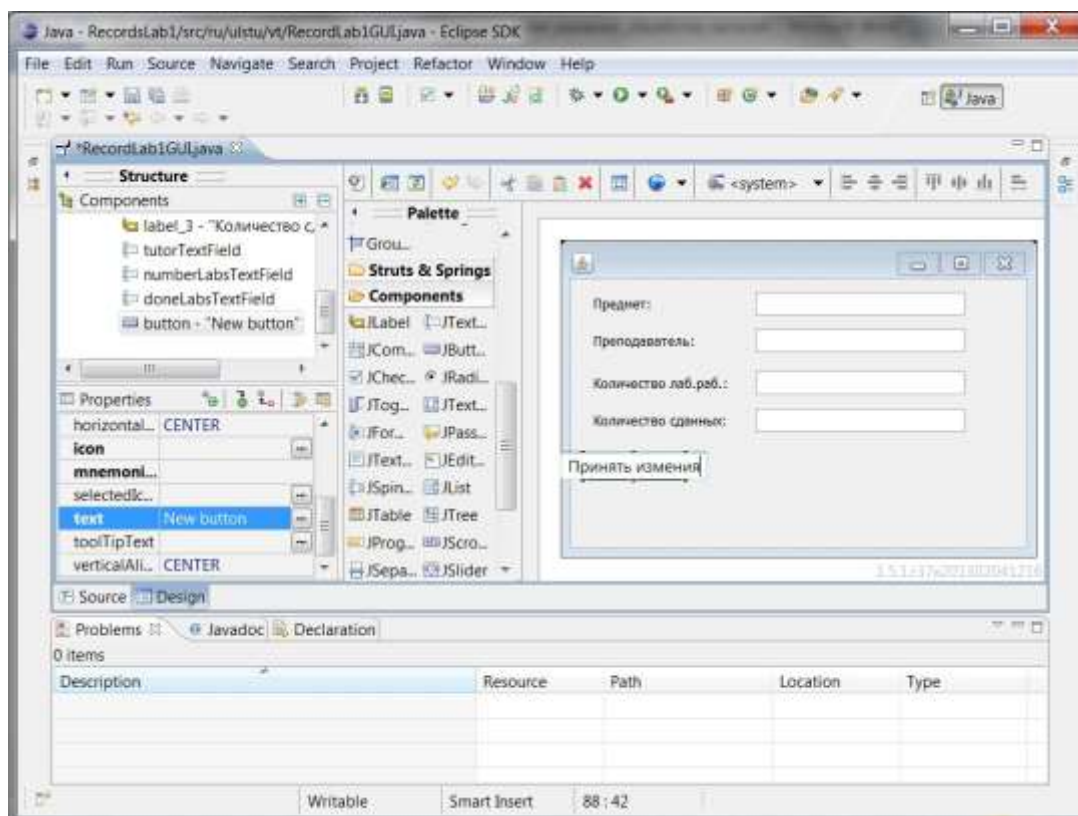


2 Выбираем место, куда нужно добавить кнопку:

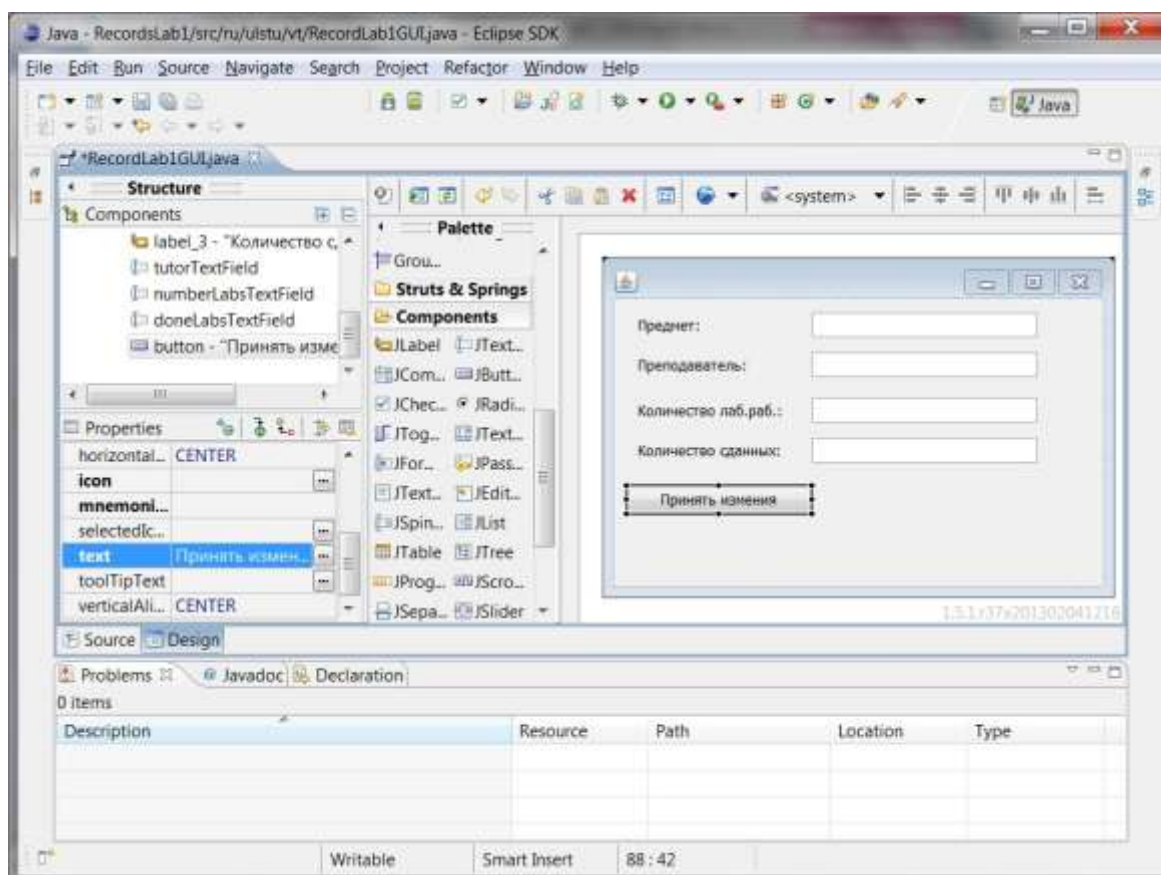




### 3 Вводим текст:



### 4 Меняем размеры и местоположение:



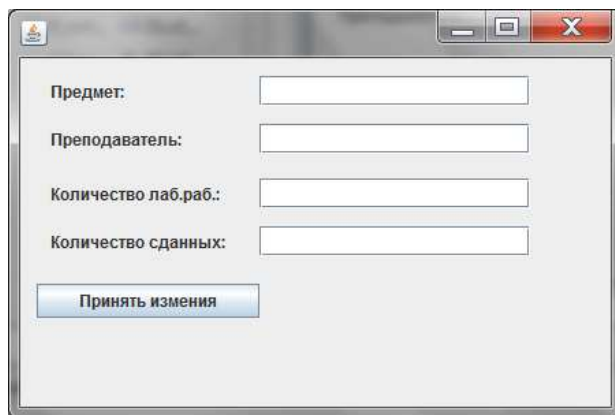




Весь код, который должен выполняться при нажатии кнопки, будет выполняться или вызываться из функции:

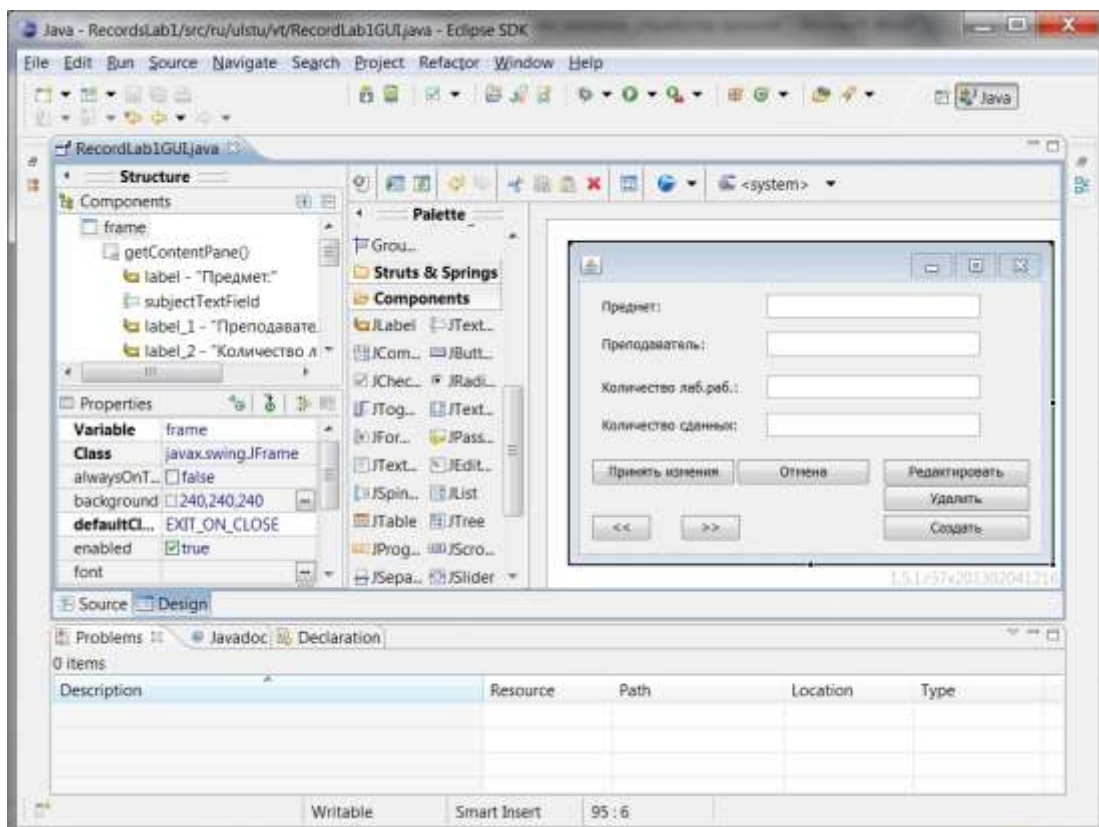
```
saveChangesButton.addActionListener(new ActionListener() {  
public void actionPerformed(ActionEvent arg0) {  
});
```

Если сейчас запустить на выполнение получившийся код, то мы увидим следующую форму:



## 6 Добавление на форму оставшихся элементов

Аналогично ранее описанным методам на форму добавляются недостающие кнопки и другие элементы, необходимые для реализации описанного в задании функционала. В итоге получится следующая форма:



Варианты для выполнения лабораторной работы.

### **Общее задание.**

Во всех вариантах требуется создать графический пользовательский интерфейс (GUI) при помощи Swing и визуального дизайнера Windows Builder в среде Eclipse.

В каждом варианте требуется обеспечить GUI для следующих операций:

- 1 Просмотр одной записи в один момент времени.
- 2 Переход к следующей записи (если она существует)
- 3 Переход к предыдущей записи (если она существует)
- 4 Удаление текущей записи
- 5 Редактирование текущей записи
- 6 Создание новой записи

Варианты содержат название объекта (записи) и список полей, для которых требуется создать GUI.

### **Варианты**

- 1 Автобус (номер маршрута, название начальной и конечной остановки, количество остановок по маршруту)
- 2 Книга (название книги, автор, издательство, количество страниц)
- 3 Файл (имя, время создания, дата создания, размер)
- 4 Спортсмен (ФИО, вид спорта, сколько лет занимается)
- 5 Студент (ФИО, группа, средний балл зачетки)
- 6 Фирма (название, вид услуг, годовой оборот)
- 7 Компьютер (производитель, частота процессора, количество ядер, объем памяти)
- 8 Спорт (вид спорта, количество занимающихся в городе, средняя стоимость комплекта инвентаря)
- 9 Автомобиль (марка, фирма-производитель, год выпуска, пробег)
- 10 Статья (название, автор, название журнала, год и месяц издания)