

Министерство науки и высшего образования Российской Федерации  
Пензенский государственный университет  
Кафедра «Вычислительная техника»

## **ОТЧЕТ**

по лабораторной работе №2  
по курсу «Разработка кроссплатформенных приложений»  
на тему «Работа с коллекциями объектов»

Выполнили студенты группы 23ВВИ2:

Кокарев Д.С.

Монин И.А.

Приняли:

Юрова О.В.

Карамышева Н.С.

Пенза 2026

**Цель работы:** изучить библиотеку стандартных коллекций Java Collections Framework, позволяющую хранить различные структуры данных.

**Лабораторное задание:** модифицировать приложение из предыдущей лабораторной работы, реализовав хранение данных таблицы с использованием библиотеки коллекций. Для этого реализовать класс RecIntegral, способный хранить одну запись таблицы. Для нечетных вариантов в качестве класса-коллекции выбрать ArrayList, для четных - LinkedList. Кроме того, добавить пару кнопок: очистить / заполнить, которые будут очищать таблицу и заполнять ее данными из коллекции соответственно.

## Ход работы:

1. Создаем класс RecIntegral в RecIntegral.java для хранения одной записи таблицы.

```
1  package rcpa.labs.model;
2
3  /**
4   * @author Ivan Monin
5   * @author Danila Kokarev
6   *
7   * Класс, представляющий запись в таблице
8   * Содержит данные для одной строки таблицы
9   */
10 public class RecIntegral {
11
12     /**
13      * Нижняя граница интегрирования
14      */
15     public String lowBorder;
16
17
18     /**
19      * Верхняя граница интегрирования
20      */
21     public String highBorder;
22
23
24     /**
25      * Шаг интегрирования
26      */
27     public String stepIntegration;
28
29
30     /**
31      * Результат вычисления
32      */
33     public String result;
34
35     public RecIntegral(String lowBorder, String highBorder, String stepIntegration, String result) {
36         this.lowBorder = lowBorder;
37         this.highBorder = highBorder;
38         this.stepIntegration = stepIntegration;
39         this.result = result;
40     }
41
42     public String[] getStringArray() {
43         return new String[]{lowBorder, highBorder, stepIntegration, result};
44     }
45 }
```

Рисунок 1 — Создание класса RecIntegral

2. В качестве класса-коллекции нужно было выбрать ArrayList для хранения кнопок.

```
34
35     /**
36     * Массив для хранения кнопок
37     *
38     * @see ArrayList
39     * @see Button
40     */
41     private ArrayList<Button> buttons;
42
```

Рисунок 2 — Выбранный класс-коллекция ArrayList

3. Создали класс FillTableButton для восстановления данных таблицы из сохраненного состояния.

```
package rcpa.labs.view;

import rcpa.labs.config.Configuration;
import rcpa.labs.model.Button;
import rcpa.labs.model.ButtonData;

import javax.swing.*;
import java.awt.*;

/**
 * @author Ivan Monin
 * @author Danila Kokarev
 *
 * Класс кнопки заполнения
 * Наследуется от Button {@link Button}
 * При нажатии восстанавливает данные таблицы из сохраненного состояния
 */
public class FillTableButton extends Button {

    /**
     * Конструктор FillTableButton
     */
}
```

```

public FillTableButton() {
    super();
    this.setFont(new Font("Arial", Font.BOLD, 14));
    this.setForeground(Color.BLACK);
    this.setBackground(new Color(250,210,1));
}

```

```

/**

```

```

    * Метод назначения действия кнопки {@link

```

```

JButton#addActionListener(java.awt.event.ActionListener)}

```

```

    * При нажатии происходит заполнение таблицы данными из массива tableRows

```

```

    * @see IntegrationTable#fillTable()

```

```

    */

```

```

private void addEventListener() {
    this.addActionListener(e->{
        getButtonData().getLinkedTable().fillTable();
    });
}

```

```

/**

```

```

    * Переопределенный метод установки данных кнопки {@link

```

```

Button#setButtonData(ButtonData)}

```

```

    *

```

```

    * @param data - данные кнопки

```

```

    * @see FillTableButton#addEventListener()

```

```

    */

```

```

@Override

```

```

public void setButtonData(ButtonData data) {
    super.setButtonData(data);
    addEventListener();
}
}

```

4. Создали класс ClearTableButton для очистки данных из таблицы интегрирования.

```
package rcpa.labs.view;

import rcpa.labs.config.Configuration;
import rcpa.labs.model.Button;
import rcpa.labs.model.ButtonData;

import javax.swing.*;
import java.awt.*;
import java.util.Arrays;

/**
 * @author Ivan Monin
 * @author Danila Kokarev
 *
 * Класс кнопки очистки
 * Наследуется от Button {@link Button}
 */
public class ClearTableButton extends Button {

    /**
     * Конструктор ClearTableButton
     */
    public ClearTableButton() {
        super();
        this.setFont(new Font("Arial", Font.BOLD, 14));
        this.setForeground(Color.BLACK);
        this.setBackground(new Color(250,210,1));
    }

    /**
     * Метод назначения действия кнопки {@link
     JButton#addActionListener(java.awt.event.ActionListener)}
```

```
* При нажатии происходит очистка таблицы и скрывание кнопок вычисления
*/
```

```
private void addEventListener() {
    this.addActionListener(e->{
        getButtonData().getLinkedTable().clearTable();

        Component[] components = getButtonData().getParentPanel().getComponents();
        for (Component comp : components) {
            if (comp instanceof DeleteButton ||
                comp instanceof CalculateButton ||
                comp instanceof CalculateTrapButton) {
                ((Button) comp).setVisible(false);
            }
        }
    });
}
```

```
/**
```

```
 * Переопределенный метод установки данных кнопки {@link
Button#setButtonData(ButtonData)}
```

```
 *
```

```
 * @param data - данные кнопки
```

```
 * @see ClearTableButton#addEventListener()
```

```
*/
```

```
@Override
```

```
public void setButtonData(ButtonData data) {
    super.setButtonData(data);
    addEventListener();
}
}
```

↓ гр. интегрирования :

1

↑ гр. интегрирования :

2

Шаг интегрирования:

0.3

Добавить

Очистить

Заполнить

Вычислить

Метод трапеции

Удалить

↓ гр. интег...	↑ гр. интег...	Шаг интегр...	Результат
1	2	0.3	0.297562911...
1	2	0.3	0.234190897...



6. Пример очистки таблицы интегрирования с помощью кнопки «Очистить» от ранее заполненных значений.

↓ гр. интегрирования :  
1

↑ гр. интегрирования :  
2

Шаг интегрирования:  
0.3

Добавить

Очистить

Заполнить

↓ гр. интег...	↑ гр. интег...	Шаг интегр...	Результат
----------------	----------------	---------------	-----------

$e^{-x}$

Рисунок 4 — Кнопка «Очистить»

**Вывод:** в ходе выполнения лабораторной работы был получен опыт в использовании библиотеки стандартных коллекций Java Collections Framework, позволяющую хранить различные структуры данных.