**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ**

**ГОМЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ**

**УНИВЕРСИТЕТ ИМЕНИ П. О. СУХОГО**

Факультет автоматизированных и информационных систем

Кафедра «Информационные технологии»

ЛАБОРАТОРНАЯ РАБОТА №6

По дисциплине: **«Технологии разработки программного обеспечения»**

на тему: ***Принципы и технологии создания дружественных пользовательских интерфейсов в Java***

Выполнил: студент группы ИТП-21

Пронуза М.Ю.

Проверил:

Малиновский И.Л.

Гомель 2024

**Цель работы**: научиться создавать дружественные пользовательские интерфейсы с помощью библиотеки Java Swing.

**Задание:**

1. Создать новую ветку для графического приложения в ранее созданном git-репозитории.

2. Используя библиотеку Java Swing, создать графический интерфейс для разработанных ранее лабораторных работ 1 и 3.

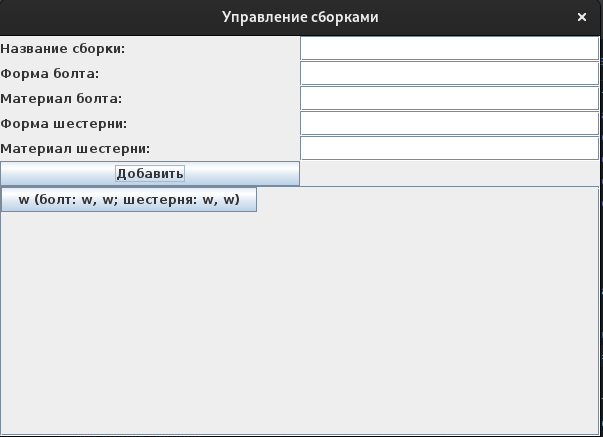
3. Элементы интерфейса должны позволять пользователю создавать, удалять и обновлять элементы коллекции, созданного в лабораторной работе 5 класса репозитория.

4. Создать и одобрить пул-реквест на слияние ветки с интерфейсом с главной веткой.

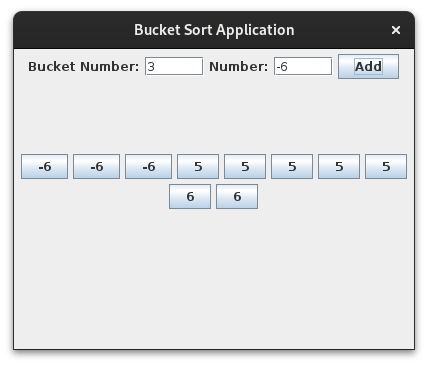
5. Составить отчет о проделанной работе.

**Ход выполнения и результаты работы:**

На рисунке 1 представлен интерфейс для первой лабораторной работы.

Рисунок 1 – интерфейс для первой лабораторной работы

На рисунке 2 представлен интерфейс для третей лабораторной работы.

Рисунок 2 – интерфейс для третей лабораторной работы

Текст класса *BucketSortGUI.java* представлен в приложении А.

Текст класса *MenuGUI.java* представлен в приложении Б.

**Вывод:**  научись применять принципы и технологии создания качественного кода в Java.

**ПРИЛОЖЕНИЕ А**

(обязательное)

**Текст *BucketSortGUI.java***

package lab3;

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.util.ArrayList;

import java.util.Collections;

public class BucketSortGUI extends JFrame {

private JTextField bucketNumberField;

private JTextField numberField;

private JPanel sortedArrayPanel;

private ArrayList<Integer> array = new ArrayList<>();

private ArrayList<JButton> sortedButtons = new ArrayList<>();

public BucketSortGUI() {

setTitle("Bucket Sort Application");

setSize(400, 300);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setLayout(new GridLayout(3, 1));

JPanel inputPanel = new JPanel();

inputPanel.setLayout(new FlowLayout());

JLabel bucketNumberLabel = new JLabel("Bucket Number:");

bucketNumberField = new JTextField(5);

JLabel numberLabel = new JLabel("Number:");

numberField = new JTextField(5);

JButton addButton = new JButton("Add");

addButton.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

addNumber();

}

});

inputPanel.add(bucketNumberLabel);

inputPanel.add(bucketNumberField);

inputPanel.add(numberLabel);

inputPanel.add(numberField);

inputPanel.add(addButton);

sortedArrayPanel = new JPanel();

sortedArrayPanel.setLayout(new FlowLayout());

add(inputPanel);

add(sortedArrayPanel);

}

private void addNumber() {

try {

int bucketNumber = Integer.parseInt(bucketNumberField.getText());

int number = Integer.parseInt(numberField.getText());

array.add(number);

Collections.sort(array);

updateSortedButtons();

} catch (NumberFormatException ex) {

JOptionPane.showMessageDialog(this, "Please enter valid integers.");

}

}

private void updateSortedButtons() {

sortedArrayPanel.removeAll();

sortedButtons.clear();

for (int num : array) {

JButton button = new JButton(String.valueOf(num));

button.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

removeNumber(num);

}

});

sortedButtons.add(button);

sortedArrayPanel.add(button);

}

sortedArrayPanel.revalidate();

sortedArrayPanel.repaint();

}

private void removeNumber(int number) {

array.remove(Integer.valueOf(number));

updateSortedButtons();

}

public static void main(String[] args) {

SwingUtilities.invokeLater(new Runnable() {

@Override

public void run() {

BucketSortGUI gui = new BucketSortGUI();

gui.setVisible(true);

}

});

}

}

**ПРИЛОЖЕНИЕ Б**

(обязательное)

**Текст *MenuGUI.java***

package lab1;

import javax.swing.\*;

import java.awt.\*;

public class MenuGUI extends JFrame {

private final AssemblyRepository assemblyRepository;

private final JPanel assemblyPanel;

private final JTextField assemblyNameField, boltFormField, boltMaterialField, gearFormField, gearMaterialField;

private final JButton addAssemblyButton;

public MenuGUI() {

assemblyRepository = new AssemblyRepository();

assemblyPanel = new JPanel();

assemblyPanel.setLayout(new BoxLayout(assemblyPanel, BoxLayout.Y\_AXIS));

assemblyNameField = new JTextField(20);

boltFormField = new JTextField(20);

boltMaterialField = new JTextField(20);

gearFormField = new JTextField(20);

gearMaterialField = new JTextField(20);

addAssemblyButton = new JButton("Добавить");

setupUI();

}

private void setupUI() {

setTitle("Управление сборками");

setSize(600, 400); // Adjusted for additional fields

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setLayout(new BorderLayout());

JPanel inputPanel = new JPanel(new GridLayout(0, 2)); // Use GridLayout for structured layout

inputPanel.add(new JLabel("Название сборки:"));

inputPanel.add(assemblyNameField);

inputPanel.add(new JLabel("Форма болта:"));

inputPanel.add(boltFormField);

inputPanel.add(new JLabel("Материал болта:"));

inputPanel.add(boltMaterialField);

inputPanel.add(new JLabel("Форма шестерни:"));

inputPanel.add(gearFormField);

inputPanel.add(new JLabel("Материал шестерни:"));

inputPanel.add(gearMaterialField);

inputPanel.add(addAssemblyButton);

add(inputPanel, BorderLayout.NORTH);

add(new JScrollPane(assemblyPanel), BorderLayout.CENTER);

addAssemblyButton.addActionListener(e -> addAssembly());

setVisible(true);

}

private void addAssembly() {

String name = assemblyNameField.getText();

String boltForm = boltFormField.getText();

String boltMaterial = boltMaterialField.getText();

String gearForm = gearFormField.getText();

String gearMaterial = gearMaterialField.getText();

if (!name.isEmpty() && !boltForm.isEmpty() && !boltMaterial.isEmpty() && !gearForm.isEmpty() && !gearMaterial.isEmpty()) {

Assembly assembly = new Assembly(name, boltForm, boltMaterial, gearForm, gearMaterial); // Adjusted constructor call

assemblyRepository.addAssembly(assembly);

// Adjusted to include form and material in the button label

String buttonLabel = String.format("%s (болт: %s, %s; шестерня: %s, %s)", name, boltForm, boltMaterial, gearForm, gearMaterial);

JButton assemblyButton = new JButton(buttonLabel);

assemblyButton.addActionListener(e -> {

assemblyRepository.deleteAssembly(name);

assemblyPanel.remove(assemblyButton);

assemblyPanel.revalidate();

assemblyPanel.repaint();

});

assemblyPanel.add(assemblyButton);

assemblyPanel.revalidate();

assemblyPanel.repaint();

assemblyNameField.setText("");

// Clear the form and material fields

boltFormField.setText("");

boltMaterialField.setText("");

gearFormField.setText("");

gearMaterialField.setText("");

}

}

public static void main(String[] args) {

SwingUtilities.invokeLater(MenuGUI::new);

}

}