**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ**

**ГОМЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ**

**УНИВЕРСИТЕТ ИМЕНИ П. О. СУХОГО**

Факультет автоматизированных и информационных систем

Кафедра «Информационные технологии»

ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №6

по дисциплине: «Технологии разработки программного обеспечения»

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

Выполнил: студент гр. ИТИ-22

Карпенко Д.Е.

Принял: преподаватель-стажёр

Малиновский И.Л.

Гомель 2023

**Цель работы**: научиться разрабатывать графический интерфейс *Java* с использованием 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 работы**

На рисунке 1 представлено окно графического интерфейса лабораторной работы №1.

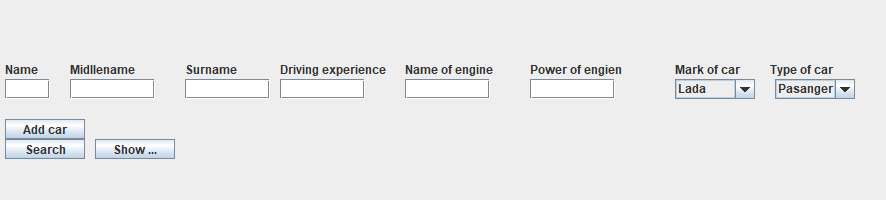
****

Рисунок 1 – Окно графического интерфейса

На рисунке 2 представлен графический интерфейс лабораторной работы №3.

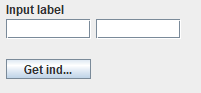
****

Рисунок 2 – Окно графического интерфейса

На рисунке 3 представлен процесс поиска числа в массиве.

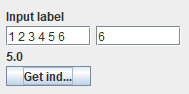
****

Рисунок 5 – Отсортированный массив

На рисунке 6 представлена ветка на *GitHub* с лабораторными со слиянием с главной веткой.

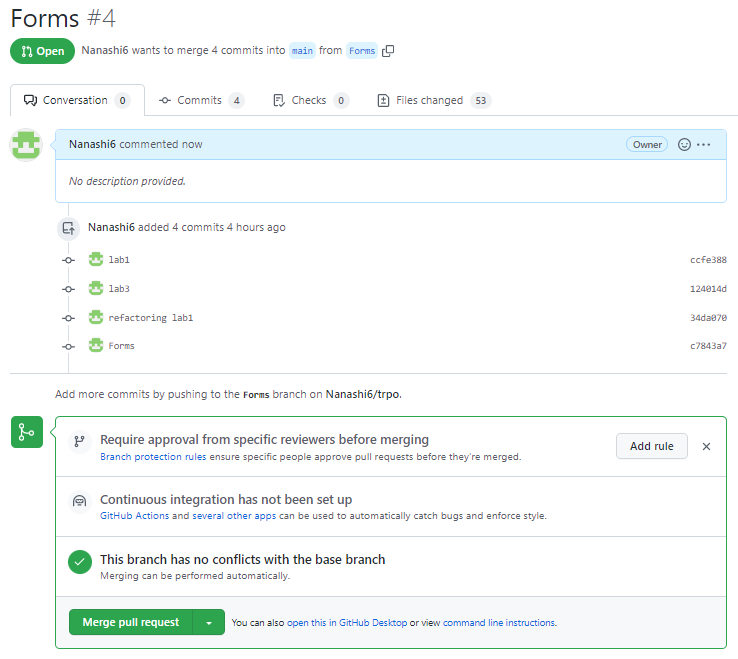
****

Рисунок 6 – Ветка на *GitHub*

**Вывод:** в результате выполнения лабораторной работы были получены знания в создании графических интерфейсов *Java* с использованием библиотеки *Java Swing*.

**ПРИЛОЖЕНИЕ А**

(обязательное)

**Текст программы**

**Файл *GUI.java (lab1)*:**

import javax.swing.\*;  
import java.awt.\*;  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
import java.util.ArrayList;  
import java.util.Arrays;  
  
public class GUI  
{  
 Meneger manager = new Meneger();  
 JFrame frame = new JFrame("GUI");  
 JTextField textFieldName = new JTextField();  
 JTextField textFieldMidellename = new JTextField();  
 JTextField textFieldSurname = new JTextField();  
 JTextField textFieldDrivingExperience = new JTextField();  
 JTextField textFieldNameOfEngien = new JTextField();  
 JTextField textFieldPowerOfEngen = new JTextField();  
 JTextField textFieldArray = new JTextField();  
 JTextField textFieldChoseNumber = new JTextField();  
 JComboBox listOfCars = new JComboBox(Enums.MarkOfCar.*values*());  
 JComboBox listOfTypes = new JComboBox(new String[] {"Pasanger", "Cargo"});  
 JButton buttonAdd = new JButton("Add car");  
 JButton buttonSearchSovietCars = new JButton("Search");  
 JButton buttonShowAllDriversWhithExperinceOverThanFiveEars = new JButton("Show Drivers");  
 JButton buttonShowIndex = new JButton("Get index");  
 // Переменные  
 String name;  
 String surnam;  
 String midllename;  
 String drivingExperience;  
 String nameOfEngien;  
 String powerOfEngen;  
 String markOfCar;  
 String typeOfCar;  
 JLabel labelShow = new JLabel();  
 JLabel labelInputArray = new JLabel("Input label");  
 JLabel labelResult = new JLabel();  
 public void load()  
 {  
 loadLabels();  
 loadTextBoxes();  
 loadLists();  
 loadButtons();  
  
 frame.setSize(1000,600);  
 frame.setLayout(null);  
 frame.setVisible(true);  
 frame.setDefaultCloseOperation(WindowConstants.*EXIT\_ON\_CLOSE*);  
 }  
  
 private void loadButtons()  
 {  
 buttonAdd.setBounds(10,120,80,20);  
 buttonAdd.addActionListener(new ActionListener()  
 {  
 @Override  
 public void actionPerformed(ActionEvent e)  
 {  
 name = textFieldName.getText();  
 surnam = textFieldSurname.getText();  
 midllename = textFieldMidellename.getText();  
 drivingExperience = textFieldDrivingExperience.getText();  
 nameOfEngien = textFieldNameOfEngien.getText();  
 powerOfEngen = textFieldPowerOfEngen.getText();  
 markOfCar = listOfCars.getSelectedItem().toString();  
 typeOfCar = (String)listOfTypes.getSelectedItem().toString();  
 Driver driver = new Driver(name, surnam, midllename, Integer.*parseInt*(drivingExperience));  
 Engen engien = new Engen(nameOfEngien,Double.*parseDouble*(powerOfEngen));  
 Car car = new Car(markOfCar, typeOfCar, driver, engien, 10);  
 manager.AddCarToList(car);  
 }  
 });  
  
 buttonSearchSovietCars.setBounds(10,140,80,20);  
 buttonSearchSovietCars.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e)  
 {  
 ArrayList<Car> sovietCars = manager.GetNewListCarOfSovietCars();  
 String str = new String();  
 for (int i = 0; i < sovietCars.size(); i++)  
 str = str + sovietCars.get(i).GetNameCar() + " ";  
  
 labelShow.setText(str);  
 }  
 });  
  
 buttonShowAllDriversWhithExperinceOverThanFiveEars.setBounds(100,140,80,20);  
 buttonShowAllDriversWhithExperinceOverThanFiveEars.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 ArrayList<Car> drivers = manager.GetNewListCarOfDrwExpOverFiveYears();  
 String str = new String();  
 for (int i = 0; i < drivers.size(); i++)  
 str = str + drivers.get(i).GetDriverCar().GetFIO()[0] + " " + drivers.get(i).GetDriverCar().GetFIO()[1] + " " + drivers.get(i).GetDriverCar().GetFIO()[2] + " \n";  
  
 labelShow.setText(str);  
 }  
 });  
  
 buttonShowIndex.setBounds(10,320,85,20);  
 buttonShowIndex.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 String str[] = textFieldArray.getText().split(" ");  
  
 double array[] = new double[str.length];  
 for(int i = 0; i < str.length; i++) {  
 System.*out*.println("[" + str[i] +"]");  
 array[i] = Double.*parseDouble*(str[i]);  
 }  
 BinarrySearch binarrySearch = new BinarrySearch(array);  
 System.*out*.println(textFieldChoseNumber.getText());  
 double index = binarrySearch.serachElement(Double.*parseDouble*(textFieldChoseNumber.getText()));  
 labelResult.setText(Double.*toString*(index));  
 }  
 });  
  
 frame.add(buttonAdd);  
 frame.add(buttonSearchSovietCars);  
 frame.add(buttonShowAllDriversWhithExperinceOverThanFiveEars);  
 frame.add(buttonShowIndex);  
 }  
 private void loadLists()  
 {  
 listOfCars.setVisible(true);  
 listOfCars.setBounds(680,80,80,20);  
 frame.add(listOfCars);  
  
 listOfTypes.setVisible(true);  
 listOfTypes.setBounds(780,80,80,20);  
 frame.add(listOfTypes);  
 }  
 private void loadLabels()  
 {  
 JLabel labelName = new JLabel("Name");  
 frame.add(labelName);  
 labelName.setVisible(true);  
 labelName.setBounds(10,60,85,20);  
  
 JLabel labelMidllename = new JLabel("Midllename");  
 frame.add(labelMidllename);  
 labelMidllename.setVisible(true);  
 labelMidllename.setBounds(labelName.getWidth() - 10, 60, 85,20);  
  
 JLabel labelSurname = new JLabel("Surname");  
 frame.add(labelSurname);  
 labelSurname.setVisible(true);  
 labelSurname.setBounds(labelName.getWidth()\*2 + 20, 60, 85,20);  
  
 JLabel labelDrivingExperience = new JLabel("Driving experience");  
 frame.add(labelDrivingExperience);  
 labelDrivingExperience.setVisible(true);  
 labelDrivingExperience.setBounds(labelName.getWidth()\*3 + 30, 60, 120,20);  
  
 JLabel labelNameOfEngien = new JLabel("Name of engine");  
 frame.add(labelNameOfEngien);  
 labelNameOfEngien.setVisible(true);  
 labelNameOfEngien.setBounds(labelName.getWidth()\*4 + 70, 60, 120,20);  
  
 JLabel labelPowerOfEngen = new JLabel("Power of engien");  
 frame.add(labelPowerOfEngen);  
 labelPowerOfEngen.setVisible(true);  
 labelPowerOfEngen.setBounds(labelName.getWidth()\*5 + 110, 60, 120,20);  
  
 JLabel labelMarkOfCar = new JLabel("Mark of car");  
 frame.add(labelMarkOfCar);  
 labelMarkOfCar.setVisible(true);  
 labelMarkOfCar.setBounds(labelName.getWidth()\*6 + 170, 60, 85,20);  
  
 JLabel labelTypeOfCar = new JLabel("Type of car");  
 frame.add(labelTypeOfCar);  
 labelTypeOfCar.setVisible(true);  
 labelTypeOfCar.setBounds(labelName.getWidth()\*7 + 180, 60, 85,20);  
  
 labelShow.setBounds(10,160,800,20);  
 frame.add(labelShow);  
  
 frame.add(labelInputArray);  
 labelInputArray.setVisible(true);  
 labelInputArray.setBounds(10,260,85,20);  
  
 frame.add(labelResult);  
 labelResult.setVisible(true);  
 labelResult.setBounds(10,300,85,20);  
 }  
  
 private void loadTextBoxes()  
 {  
 frame.add(textFieldName);  
 textFieldName.setVisible(true);  
 textFieldName.setBounds(10,80,85,20);  
  
 frame.add(textFieldMidellename);  
 textFieldMidellename.setVisible(true);  
 textFieldMidellename.setBounds(textFieldName.getWidth() - 10,80,85,20);  
  
 frame.add(textFieldSurname);  
 textFieldSurname.setVisible(true);  
 textFieldSurname.setBounds(textFieldName.getWidth() \* 2 + 20,80,85,20);  
  
 frame.add(textFieldDrivingExperience);  
 textFieldDrivingExperience.setVisible(true);  
 textFieldDrivingExperience.setBounds(textFieldName.getWidth() \* 3 + 30,80,85,20);  
  
 frame.add(textFieldNameOfEngien);  
 textFieldNameOfEngien.setVisible(true);  
 textFieldNameOfEngien.setBounds(textFieldName.getWidth() \* 4 + 70,80,85,20);  
  
 frame.add(textFieldPowerOfEngen);  
 textFieldPowerOfEngen.setVisible(true);  
 textFieldPowerOfEngen.setBounds(textFieldName.getWidth() \* 5 + 110,80,85,20);  
  
 textFieldName.setBounds(10,80,45,20);  
  
 frame.add(textFieldArray);  
 textFieldArray.setVisible(true);  
 textFieldArray.setBounds(10,280,85,20);  
  
 frame.add(textFieldChoseNumber);  
 textFieldChoseNumber.setVisible(true);  
 textFieldChoseNumber.setBounds(100,280,85,20);  
 }  
  
}