**МИНОБРНАУКИ РОССИИ**

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**

**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**

**Кафедра Вычислительной Техники**

**ОТЧЕТ**

**по лабораторной работе №3**

**по дисциплине «Объектно-ориентированное программирование»**

**Тема: «Разработка интерфейса пользователя».**

|  |  |  |
| --- | --- | --- |
| Студентка гр. 3316 |  | Рябова Д.Д. |
| Преподаватель |  | Гречухин М.Н. |

Санкт-Петербург

2024

**Содержание:**

Цель работы.......................................................................................................................3

Постановка задачи……………………...…..…………………………………......3

Описание назначения экранной формы……….….……………………………………….………………………....3

Проверка работоспособности…...………………….………………………………….…….4

Текст программы………………………………………………………………….5

Вывод……………………………………………………………..………………12

**Цель работы.**

Целью является знакомство с правилами построения экранной формы.

**Задание курсовой работы (Вариант 13):**

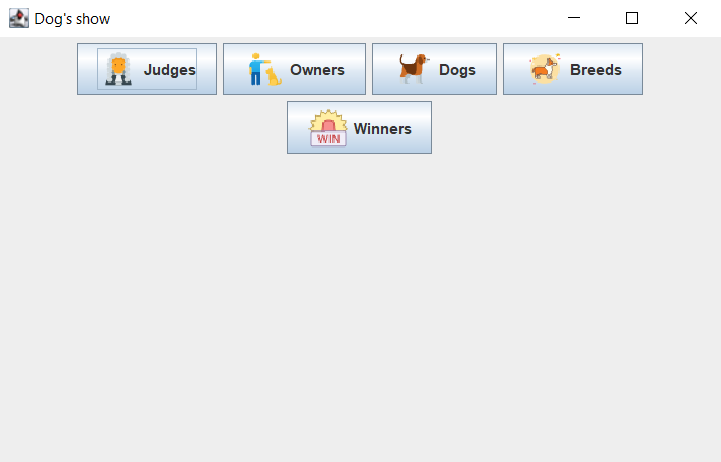
Разработать ПК для администратора выставки собак. В ПК должны храниться сведения о собаках, их владельцах и судьях. Администратор выставки может добавлять, изменять и удалять эти сведения. Ему может потребоваться следующая информация:

* список судей и какие породы они обслуживают;
* какая кличка и порода собаки у данного владельца;
* какие собаки получили награды на выставке;
* список владельцев собак указанной породы.

**Описание назначения экранной формы.**

В данной экранной форме необходимо добавить кнопки, соответствующие каждой из сущностей нашей базы данных. Для каждой из кнопок нужно подобрать подходящую иконку, чтобы сделать интерфейс более понятным.

Данная экранная форма предназначена для администратора выставки собак, поэтому в ней будут содержаться кнопки, которые впоследствии будут открывать нужные списки, такие, как: список владельцев собак, список собак, список судей, список пород и список победителей выставки.**Проверка работоспособности.**

****

В данном окне при запуске программы появляется пять кнопок без реализации открытия списка. Каждая из них впоследствии будет открывать соответствующий список.

**Текст программы.**

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import javax.swing.\*;

import javax.swing.table.DefaultTableModel;

public class dog\_show {

    public static void main(String[] args) {

        JFrame frame = new JFrame("Dog's show");

        frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

        JPanel buttonPanel = new JPanel();

        buttonPanel.setLayout(new FlowLayout());

        // Создаем иконки

        Icon judgeIcon = new ImageIcon(dog\_show.class.getResource("/judge.png"));

        Icon ownerIcon = new ImageIcon(dog\_show.class.getResource("/owner.png"));

        Icon dogIcon = new ImageIcon(dog\_show.class.getResource("/dogs.png"));

        Icon breedIcon = new ImageIcon(dog\_show.class.getResource("/breed.png"));

        Icon winIcon = new ImageIcon(dog\_show.class.getResource("/win.png"));

        JButton jBtn = new JButton("Judges", judgeIcon);

        JButton oBtn = new JButton("Owners", ownerIcon);

        JButton dBtn = new JButton("Dogs", dogIcon);

        JButton bBtn = new JButton("Breeds", breedIcon);

        JButton wBtn = new JButton("Winners", winIcon);

        buttonPanel.add(jBtn);

        buttonPanel.add(oBtn);

        buttonPanel.add(dBtn);

        buttonPanel.add(bBtn);

        buttonPanel.add(wBtn);

        frame.getContentPane().add(buttonPanel, BorderLayout.CENTER);

        jBtn.addActionListener(new ActionListener() {

            @Override

            public void actionPerformed(ActionEvent e) {

                showTable3("Judges", "List");

            }

        });

        oBtn.addActionListener(new ActionListener() {

            @Override

            public void actionPerformed(ActionEvent e) {

                showTable3o("Owners", "List");

            }

        });

        dBtn.addActionListener(new ActionListener() {

            @Override

            public void actionPerformed(ActionEvent e) {

                showTable4("Dogs", "List");

            }

        });

        bBtn.addActionListener(new ActionListener() {

            @Override

            public void actionPerformed(ActionEvent e) {

                showTable2("Breeds", "List");

            }

        });

        wBtn.addActionListener(new ActionListener() {

            @Override

            public void actionPerformed(ActionEvent e) {

                showTable2w("Winners", "List");

            }

        });

        frame.setSize(300, 200);

        frame.setVisible(true);

    }

    private static void showTable3(String tableName, String title) {

        JFrame tableFrame = new JFrame(tableName);

        tableFrame.setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);

        DefaultTableModel model = new DefaultTableModel();

        JTable table = new JTable(model);

        // Добавляем заголовки столбцов

        String[] columns = {"ID", "Name", "Surname"};

        for (String column : columns) {

            model.addColumn(column);

        }

        // Добавляем данные в таблицу

        Object[][] data = {

            {1, "Bene", "Jerson"},

            {2, "Darto", "Uta"},

            {3, "Fasr", "Meow"}

        };

        for (Object[] row : data) {

            model.addRow(row);

        }

        // Создаем JScrollPane для прокрутки таблицы

        JScrollPane scrollPane = new JScrollPane(table);

        // Устанавливаем содержимое окна

        tableFrame.getContentPane().add(scrollPane, BorderLayout.CENTER);

        // Устанавливаем размеры окна

        tableFrame.setSize(400, 300);

        // Отображаем окно

        tableFrame.setVisible(true);

    }

    private static void showTable2(String tableName, String title) {

        JFrame tableFrame = new JFrame(tableName);

        tableFrame.setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);

        DefaultTableModel model = new DefaultTableModel();

        JTable table = new JTable(model);

        // Добавляем заголовки столбцов

        String[] columns = {"ID", "Name"};

        for (String column : columns) {

            model.addColumn(column);

        }

        // Добавляем данные в таблицу

        Object[][] data = {

            {1, "Haski"},

            {2, "Corgi"},

            {3, "Shiba-Inu"}

        };

        for (Object[] row : data) {

            model.addRow(row);

        }

        // Создаем JScrollPane для прокрутки таблицы

        JScrollPane scrollPane = new JScrollPane(table);

        // Устанавливаем содержимое окна

        tableFrame.getContentPane().add(scrollPane, BorderLayout.CENTER);

        // Устанавливаем размеры окна

        tableFrame.setSize(400, 300);

        // Отображаем окно

        tableFrame.setVisible(true);

    }

    private static void showTable2w(String tableName, String title) {

        JFrame tableFrame = new JFrame(tableName);

        tableFrame.setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);

        DefaultTableModel model = new DefaultTableModel();

        JTable table = new JTable(model);

        // Добавляем заголовки столбцов

        String[] columns = {"ID winner", "Dog's id"};

        for (String column : columns) {

            model.addColumn(column);

        }

        // Добавляем данные в таблицу

        Object[][] data = {

            {1, "2"},

            {2, "7"},

            {3, "8"}

        };

        for (Object[] row : data) {

            model.addRow(row);

        }

        // Создаем JScrollPane для прокрутки таблицы

        JScrollPane scrollPane = new JScrollPane(table);

        // Устанавливаем содержимое окна

        tableFrame.getContentPane().add(scrollPane, BorderLayout.CENTER);

        // Устанавливаем размеры окна

        tableFrame.setSize(400, 300);

        // Отображаем окно

        tableFrame.setVisible(true);

    }

    private static void showTable3o(String tableName, String title) {

        JFrame tableFrame = new JFrame(tableName);

        tableFrame.setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);

        DefaultTableModel model = new DefaultTableModel();

        JTable table = new JTable(model);

        // Добавляем заголовки столбцов

        String[] columns = {"ID", "Name", "Surname"};

        for (String column : columns) {

            model.addColumn(column);

        }

        // Добавляем данные в таблицу

        Object[][] data = {

            {1, "Ben", "Jer"},

            {2, "Daryy", "Opera"},

            {3, "Fedor", "Meow"}

        };

        for (Object[] row : data) {

            model.addRow(row);

        }

        // Создаем JScrollPane для прокрутки таблицы

        JScrollPane scrollPane = new JScrollPane(table);

        // Устанавливаем содержимое окна

        tableFrame.getContentPane().add(scrollPane, BorderLayout.CENTER);

        // Устанавливаем размеры окна

        tableFrame.setSize(400, 300);

        // Отображаем окно

        tableFrame.setVisible(true);

    }

    private static void showTable4(String tableName, String title) {

        JFrame tableFrame = new JFrame(tableName);

        tableFrame.setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);

        DefaultTableModel model = new DefaultTableModel();

        JTable table = new JTable(model);

        // Добавляем заголовки столбцов

        String[] columns = {"ID", "Name", "Breed's id", "Owner's id"};

        for (String column : columns) {

            model.addColumn(column);

        }

        // Добавляем данные в таблицу

        Object[][] data = {

            {1, "Pie", "4", "6"},

            {2, "Kleo", "8", "12"},

            {3, "Zoe", "5", "9"}

        };

        for (Object[] row : data) {

            model.addRow(row);

        }

        // Создаем JScrollPane для прокрутки таблицы

        JScrollPane scrollPane = new JScrollPane(table);

        // Устанавливаем содержимое окна

        tableFrame.getContentPane().add(scrollPane, BorderLayout.CENTER);

        // Устанавливаем размеры окна

        tableFrame.setSize(400, 300);

        // Отображаем окно

        tableFrame.setVisible(true);

    }

}

/\*

public class TableSearchWindow {

    private JFrame frame;

    private JTextField searchField;

    private JTable originalTable;

    private DefaultTableModel originalModel;

    private DefaultTableModel filteredModel;

    private JTable filteredTable;

    private JPanel searchPanel;

    public TableSearchWindow() {

        createAndShowGUI();

    }

    private void createAndShowGUI() {

        frame = new JFrame("Поиск в таблице");

        frame.setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);

        frame.setSize(800, 600);

        searchPanel = new JPanel();

        searchField = new JTextField(20);

        JButton searchButton = new JButton("Найти");

        searchPanel.add(searchField);

        searchPanel.add(searchButton);

        searchButton.addActionListener(new ActionListener() {

            @Override

            public void actionPerformed(ActionEvent e) {

                performSearch();

            }

        });

        originalModel = new DefaultTableModel();

        String[] columnNames = {"Имя", "Возраст"};

        Object[][] data = {

            {"Анна", "25"},

            {"Петр", "30"},

            {"Мария", "28"},

            {"Иван", "35"}

        };

        originalModel.setDataVector(data, columnNames);

        originalTable = new JTable(originalModel);

        JScrollPane originalScrollPane = new JScrollPane(originalTable);

        frame.getContentPane().add(searchPanel, BorderLayout.NORTH);

        frame.getContentPane().add(originalScrollPane, BorderLayout.CENTER);

        frame.setVisible(true);

    }

    private void performSearch() {

        String searchTerm = searchField.getText().toLowerCase();

        filteredModel = new DefaultTableModel();

        filteredModel.addColumn(originalModel.getColumnName(0)); // Первый столбец

        filteredModel.addColumn(originalModel.getColumnName(1));

        for (int i = 0; i < originalModel.getRowCount(); i++) {

            boolean found = false;

            for (int j = 0; j < originalModel.getColumnCount(); j++) {

                String cellValue = originalModel.getValueAt(i, j).toString().toLowerCase();

                if (cellValue.contains(searchTerm)) {

                    found = true;

                    break;

                }

            }

            if (found) {

                filteredModel.addRow(originalModel.getDataVector().elementAt(i));

            }

        }

        filteredTable = new JTable(filteredModel);

        frame.getContentPane().removeAll();

        frame.getContentPane().add(searchPanel, BorderLayout.NORTH);

        frame.getContentPane().add(new JScrollPane(filteredTable), BorderLayout.CENTER);

        frame.revalidate();

        frame.repaint();

    }

}\*/

**Вывод.**

В результате выполнения работы была создана экранная форма для приложения. В дальнейшем будет реализована функция кнопок.