wМинистерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение   
высшего образования

**Иркутский национальный исследовательский технический университет**

|  |
| --- |
| Институт информационных технологий и анализа данных |
| наименование института |

|  |
| --- |
| **Отчет** |
| по лабораторной работе №5 по дисциплине «Объектно-ориентированное программирование» «Разработка приложения с оконным графическим пользовательским интерфейсом» |
| наименование темы  Вариант №18 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Выполнил студент |  | АСУб-20-1 |  |  |  | В.М. Устюжанин |
|  |  | шифр |  | подпись |  | И.О. Фамилия |
| Проверил |  |  |  |  |  | Е.Н. Бакшеева |
|  |  |  |  | подпись |  | И.О. Фамилия |
| Работа защищена с оценкой | | | |  | | |

Иркутск 2021г.

**1 Постановка задачи**

Для выполнения задания необходимо разработать графический пользовательский интерфейс средствами java для работы с классами, реализованными при выполнении либо 3-ей, либо 4-ой лабораторной работы.  
Графический пользовательский интерфейс должен предоставлять возможности работы со всеми возможными функциями программы. Состав используемых компонентов определяется самостоятельно, но для них должны быть предусмотрены соответствующие планировщики раскладки компонентов в основных контейнерах.

**Задание:**

Реализовать работу службы по кормлению животных в зоопарке. В процессе работы необходимо, чтобы животные получали еду, на которую должны отреагировать в зависимости от своего типа питания (травоядное, плотоядное, всеядное). Реализовать систему сытости, для предотвращения бесконечной кормежки одного и того же животного.

**2 Проектирование графического интерфейса пользователя.**

1. Главная вкладка программы

На главной вкладке находится описание задания и принципа работы с программой.

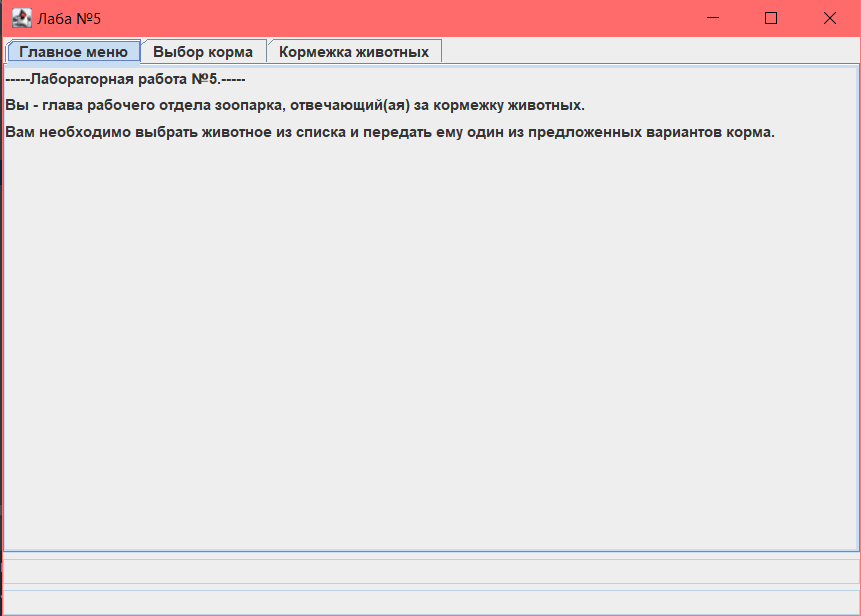


Рисунок 1 – Окно главного раздела программы

1. Раздел выбора еды

В данном разделе пользователь осуществляет выбор из предоставленных типов еды на выбор.

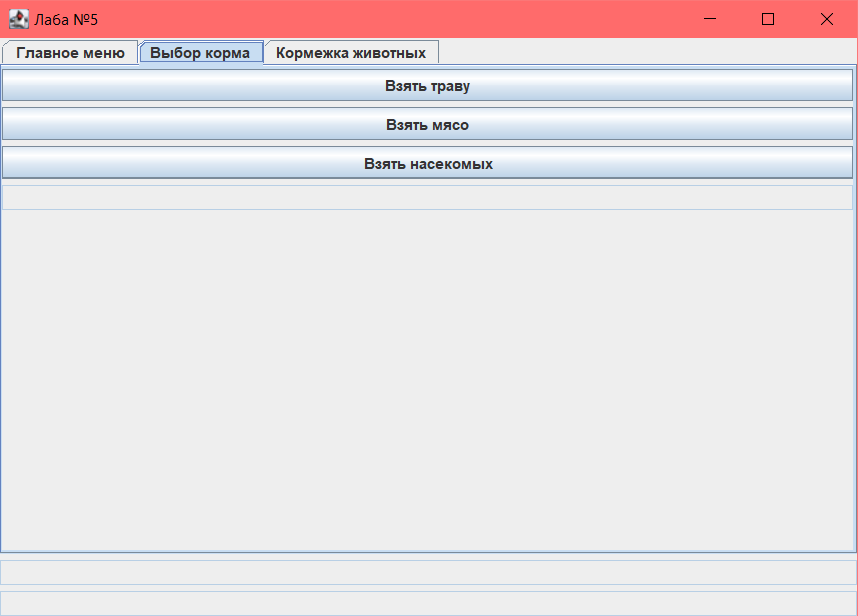


Рисунок 2 – Окно раздела выбора еды

1. Раздел, осуществляющий процесс кормления.

В данном разделе пользователь выдает еду, выбранную в предыдущем разделе. Животное, принимая еду, сперва проверит свою сытость, после этого, оно проверит тип еды, передаваемый ей и сверится с своим типом питания (кроме всеядных). В результате пользователь увидит соответствующую реакцию в одном из двух полей снизу.

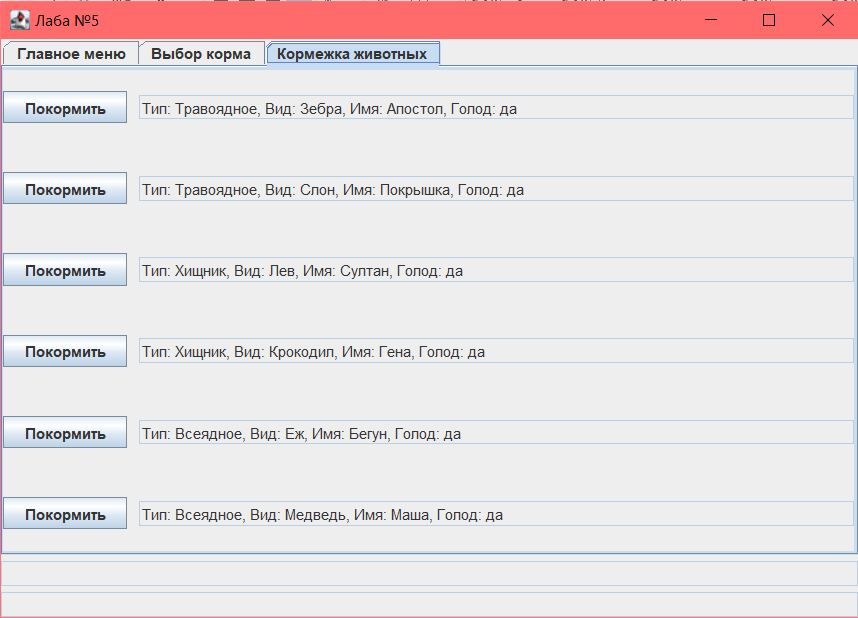


Рисунок 3 – Окно раздела кормежки животных

3 Таблицы описания классов

Таблица 1 – Описание класса LabRab5

|  |  |  |  |
| --- | --- | --- | --- |
| **Поля** | | | |
| - | | | |
| **Методы** | | | |
| **Параметры и сигнатура** | **Описание** | **Локальные переменные** | |
| Имя | Тип |
| public static void main(String[] args) | Главный метод, осуществляющий работу программы. | gui | GuiForm |
| **Конструкторы** | | | |
| - | | | |

Таблица 2 – Описание класса GuiForm

|  |  |  |  |
| --- | --- | --- | --- |
| **Поля** | | | |
| private JPanel panel1; private JTabbedPane tavvedPane1; private JButton TakeInsects; private JButton TakeMeat; private JButton TakeGrass; private JTextField FoodCondition; private JButton Feed1Button; private JButton Feed2Button; private JButton Feed3Button; private JButton Feed4Button; private JButton Feed5Button; private JButton Feed6Button; JTextField FeedCondition; JTextField Animal1; JTextField Animal2; JTextField Animal3; JTextField Animal3; JTextField Animal4; JTextField Animal5; JTextField Animal6; JTextField CheckCondition; String foodWeHave; | | | |
| **Методы** | | | |
| **Параметры и сигнатура** | **Описание** | **Локальные переменные** | |
| Имя | Тип |
| TakeGrass.addActionListener(new ActionListener() | Listener для кнопки из раздела выбора еды. | - | - |
| TakeMeat.addActionListener(new ActionListener() | Listener для кнопки из раздела выбора еды. | - | - |
| TakeInsects.addActionListener(new ActionListener() | Listener для кнопки из раздела выбора еды. | - | - |
| Feed1Button.addActionListener(new ActionListener() | Listener для кнопки из раздела кормежки. | - | - |
| Feed2Button.addActionListener(new ActionListener() | Listener для кнопки из раздела кормежки. | - | - |
| Feed3Button.addActionListener(new ActionListener() | Listener для кнопки из раздела кормежки. | - | - |
| Feed4Button.addActionListener(new ActionListener() | Listener для кнопки из раздела кормежки. | - | - |
| Feed5Button.addActionListener(new ActionListener() | Listener для кнопки из раздела кормежки. | - | - |
| Feed6Button.addActionListener(new ActionListener() | Listener для кнопки из раздела кормежки. | - public GuiForm() | - |
| **Конструкторы** | | | |
| public GuiForm() | Нулевой конструктор, вызываемый классом LabRab5 | names  species  types | String[]  String[]  String[] |

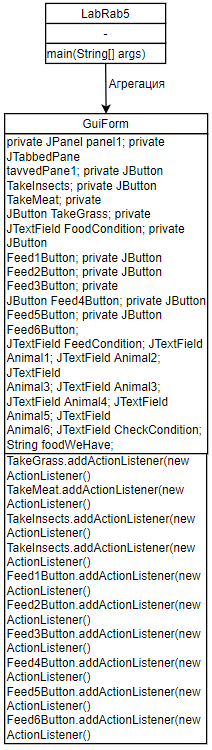


Рисунок 4 – UML диаграмма классов проекта LabRab5

3 Таблица тестов

|  |  |  |  |
| --- | --- | --- | --- |
| Таблица 3 –Тестирование | | | |
| № | Выполняемое действие | Исходные данные | Результат |
| 1 | Запуск программы | - | Запуск программы выполнен успешно |
| 2 | Животное получило неверный вид еды | - | Тест выполнен успешно |
| 3 | Животное получило верный вид еды | - | Тест выполнен успешно |
| 4 | Животное получило еду, будучи в сытом состоянии | - | Тест выполнен успешно |
| 5 | Пользователь не выбрал еду | - | Тест выполнен успешно |

4 Результаты тестирования

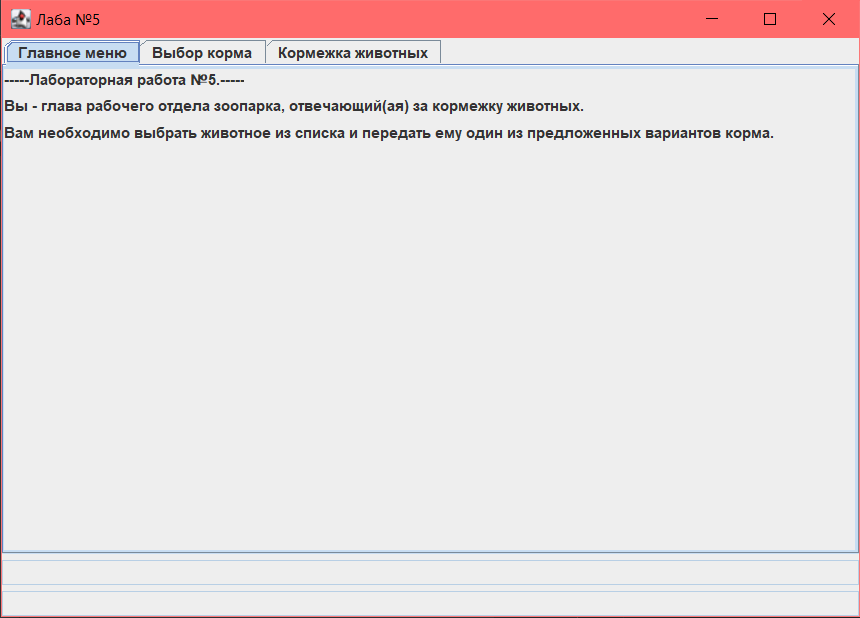


Рисунок 5 – Результат теста запуска программы

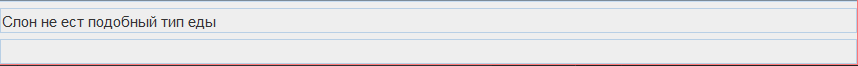


Рисунок 6 – Результат теста №2

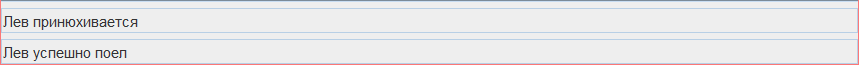


Рисунок 7 – Результат теста №3

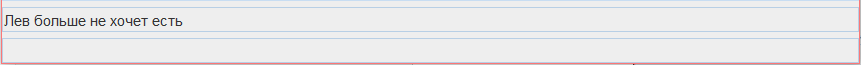


Рисунок 8 – Результат теста №4

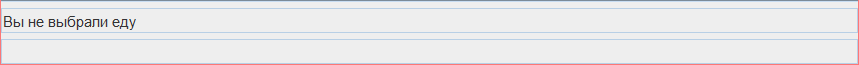


Рисунок 9 – Результат теста №5

5 Исходный код

1) Класс LabRab3

import javax.swing.\*;  
  
public class LabRab5  
{  
 public static void main(String[] args)  
 {  
 GuiForm gui = new GuiForm();  
  
 gui.setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);  
 gui.setSize(700, 500);  
 gui.setVisible(true);  
 }  
}

2) Класс UserMenu

import javax.swing.\*;  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
import java.net.http.HttpRequest;  
import java.util.ArrayList;  
  
public class GuiForm extends JFrame  
{  
 private JPanel panel1;  
 private JTabbedPane tabbedPane1;  
 private JButton TakeInsects;  
 private JButton TakeMeat;  
 private JButton TakeGrass;  
 private JTextField FoodCondition;  
 private JButton Feed1Button;  
 private JButton Feed6Button;  
 private JButton Feed5Button;  
 private JButton Feed4Button;  
 private JButton Feed3Button;  
 private JButton Feed2Button;  
 private JTextField FeedCondition;  
 private JTextField Animal6;  
 private JTextField Animal5;  
 private JTextField Animal4;  
 private JTextField Animal3;  
 private JTextField Animal2;  
 private JTextField Animal1;  
 private JTextField CheckCondition;  
 String foodWeHave = "";  
 public GuiForm()  
 {  
 super("Лаба №5");  
 String[] names = new String[]{"Апостол","Покрышка","Султан","Гена","Бегун","Маша"};  
 String[] species = new String[]{"Зебра","Слон","Лев","Крокодил","Еж","Медведь"};  
 String[] types = new String[]{"Травоядное","Травоядное","Хищник","Хищник","Всеядное","Всеядное"};  
 this.add(panel1);  
 Animal1.setText("Тип: "+types[0]+", Вид: "+species[0]+", Имя: "+names[0]+", Голод: да");  
 Animal2.setText("Тип: "+types[1]+", Вид: "+species[1]+", Имя: "+names[1]+", Голод: да");  
 Animal3.setText("Тип: "+types[2]+", Вид: "+species[2]+", Имя: "+names[2]+", Голод: да");  
 Animal4.setText("Тип: "+types[3]+", Вид: "+species[3]+", Имя: "+names[3]+", Голод: да");  
 Animal5.setText("Тип: "+types[4]+", Вид: "+species[4]+", Имя: "+names[4]+", Голод: да");  
 Animal6.setText("Тип: "+types[5]+", Вид: "+species[5]+", Имя: "+names[5]+", Голод: да");  
 TakeGrass.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 FoodCondition.setText("Вы выбрали траву");  
 foodWeHave = "Трава";  
 }  
 });  
 TakeMeat.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 FoodCondition.setText("Вы выбрали мясо");  
 foodWeHave = "Мясо";  
 }  
 });  
 TakeInsects.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 FoodCondition.setText("Вы выбрали насекомых");  
 foodWeHave = "Насекомые";  
 }  
 });  
  
 Feed1Button.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 CheckCondition.setText("");  
 FeedCondition.setText("");  
 if (foodWeHave=="")  
 {  
 CheckCondition.setText("Вы не выбрали еду");  
 }  
 else if (Animal1.getText().equals("Тип: "+types[0]+", Вид: "+species[0]+", Имя: "+names[0]+", Голод: да"))  
 {  
 CheckCondition.setText(species[0]+" принюхивается");  
 if (foodWeHave.equals("Трава"))  
 {  
 FeedCondition.setText(species[0]+" успешно поел");  
 Animal1.setText("Тип: "+types[0]+", Вид: "+species[0]+", Имя: "+names[0]+", Голод: нет");  
 }  
 else  
 CheckCondition.setText(species[0]+" не ест подобный тип еды");  
 }  
 else  
 CheckCondition.setText(species[0]+" больше не хочет есть");  
 }  
 });  
 Feed2Button.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 CheckCondition.setText("");  
 FeedCondition.setText("");  
 if (foodWeHave=="")  
 {  
 CheckCondition.setText("Вы не выбрали еду");  
 }  
 else if (Animal2.getText().equals("Тип: "+types[1]+", Вид: "+species[1]+", Имя: "+names[1]+", Голод: да"))  
 {  
 CheckCondition.setText(species[1]+" принюхивается");  
 if (foodWeHave.equals("Трава"))  
 {  
 FeedCondition.setText(species[1]+" успешно поел");  
 Animal2.setText("Тип: "+types[1]+", Вид: "+species[1]+", Имя: "+names[1]+", Голод: нет");  
 }  
 else  
 CheckCondition.setText(species[1]+" не ест подобный тип еды");  
 }  
 else  
 CheckCondition.setText(species[0]+" больше не хочет есть");  
 }  
 });  
 Feed3Button.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 CheckCondition.setText("");  
 FeedCondition.setText("");  
 if (foodWeHave=="")  
 {  
 CheckCondition.setText("Вы не выбрали еду");  
 }  
 else if (Animal3.getText().equals("Тип: "+types[2]+", Вид: "+species[2]+", Имя: "+names[2]+", Голод: да"))  
 {  
 CheckCondition.setText(species[2]+" принюхивается");  
 if (foodWeHave.equals("Мясо"))  
 {  
 FeedCondition.setText(species[2]+" успешно поел");  
 Animal3.setText("Тип: "+types[2]+", Вид: "+species[2]+", Имя: "+names[2]+", Голод: нет");  
 }  
 else  
 CheckCondition.setText(species[2]+" не ест подобный тип еды");  
 }  
 else  
 CheckCondition.setText(species[2]+" больше не хочет есть");  
 }  
 });  
 Feed4Button.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 CheckCondition.setText("");  
 FeedCondition.setText("");  
 if (foodWeHave=="")  
 {  
 CheckCondition.setText("Вы не выбрали еду");  
 }  
 else if (Animal4.getText().equals("Тип: "+types[3]+", Вид: "+species[3]+", Имя: "+names[3]+", Голод: да"))  
 {  
 CheckCondition.setText(species[3]+" принюхивается");  
 if (foodWeHave.equals("Мясо"))  
 {  
 FeedCondition.setText(species[3]+" успешно поел");  
 Animal4.setText("Тип: "+types[3]+", Вид: "+species[3]+", Имя: "+names[3]+", Голод: нет");  
 }  
 else  
 CheckCondition.setText(species[3]+" не ест подобный тип еды");  
 }  
 else  
 CheckCondition.setText(species[3]+" больше не хочет есть");  
 }  
 });  
 Feed5Button.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 CheckCondition.setText("");  
 FeedCondition.setText("");  
 if (foodWeHave=="")  
 {  
 CheckCondition.setText("Вы не выбрали еду");  
 }  
 else if (Animal5.getText().equals("Тип: "+types[4]+", Вид: "+species[4]+", Имя: "+names[4]+", Голод: да"))  
 {  
 CheckCondition.setText(species[4]+" принюхивается");  
 FeedCondition.setText(species[4]+" успешно поел");  
 Animal5.setText("Тип: "+types[4]+", Вид: "+species[4]+", Имя: "+names[4]+", Голод: нет");  
 }  
 else  
 CheckCondition.setText(species[4]+" больше не хочет есть");  
  
 }  
 });  
 Feed6Button.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 CheckCondition.setText("");  
 FeedCondition.setText("");  
 if (foodWeHave=="")  
 {  
 CheckCondition.setText("Вы не выбрали еду");  
 }  
 else if (Animal6.getText().equals("Тип: "+types[5]+", Вид: "+species[5]+", Имя: "+names[5]+", Голод: да"))  
 {  
 CheckCondition.setText(species[5]+" принюхивается");  
 FeedCondition.setText(species[5]+" успешно поел");  
 Animal6.setText("Тип: "+types[5]+", Вид: "+species[5]+", Имя: "+names[5]+", Голод: нет");  
 }  
 else  
 CheckCondition.setText(species[5]+" больше не хочет есть");  
  
 }  
 });  
 }  
}

Заключение

В процессе выполнения лабораторной работы № 5 по объектно-ориентированному программированию был освоен навык создания графического пользовательского интерфейса на основе Swing и GUI.

6 Список использованных источников

1. javarush.ru
2. [www.cyberforum.ru](http://www.cyberforum.ru)
3. Лекция Аршинского В.Л. по теме «Swing»

https://www.youtube.com/watch?v=pJfX9tmd-Sw&t=3677s