Министерство образования и науки РФ   
федеральное государственное бюджетное образовательное учреждение высшего профессионального образования   
«РЯЗАНСКИЙ ГОСУДАРСТВЕННЫЙ РАДИОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ» КАФЕДРА ВЫСШЕЙ ПРИКЛАДНОЙ МАТЕМАТИКИ

**Курсовая работа**

по дисциплине: «Объектно-ориентированное программирование»

на тему: Разработка программы «Разновидности зверей в мире»

Выполнила: ст-ка гр.606  
 Пахомова Н.Ф.

Проверил: Королев Е.Е.

Рязань, 2018 год

**Задание на курсовую работу**

Студентки 2 курса

по дисциплине «Объектно-ориентированное программирование»

Пахомовой Натальи Федоровны

Тема: Разработка программы «Разновидности зверей в мире»

Преподаватель от кафедры: Королев Евгений Евгеньевич

Срок сдачи выполненной работы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Цель работы:** используя принципы и возможности объектно-ориентированного программирования, разработать программу на языке объектно-ориентированного программирования Java.

Для достижения поставленной цели необходимо решить **следующие задачи**.

1.      Изучить основы объектно-ориентированного программирования.

2.      Выбрать предметную область.

3.      Создать репозиторий для разработки приложения.

4.      Создать диаграмму классов выбранной предметной области.

5.      Реализовать описанный диаграммой классов функционал.

*Вопросы, подлежащие разработке:*

* изучение методов и средства создания приложений на основе объектно-ориентированного подхода;
* инструменты создания диаграммы классов предметной области;
* изучение средств контроля версий и способы их использования;
* основные принципы ООП;
* зачем нужна диаграмма классов;
* оформление курсовой работы.

# Содержание

[Введение 4](#_Toc516953097)

[1 описание Предметной области программы 5](#_Toc516953098)

[2 Диаграмма классов 6](#_Toc516953099)

[3 Код программы 7](#_Toc516953100)

[3.1 Класс родителя «zveri» 7](#_Toc516953101)

[3.2 Класс наследника «cat» 8](#_Toc516953102)

[3.3 Класс наследника «mays» 8](#_Toc516953103)

[3.4 Класс наследника «dog» 9](#_Toc516953104)

[3.5 Выполнение программы «proverka» 9](#_Toc516953105)

[Заключение 10](#_Toc516953106)

[Список литературы 11](#_Toc516953107)

[Приложение демонстрация выполнения программы 12](#_Toc516953108)

# Введение

Объектно-ориентированный подход в последнее десятилетие стал одним из наиболее интенсивно развивающихся направлений в программировании и наиболее популярным средством разработки программного обеспечения. С точки зрения языка программирования класс объектов можно рассматривать как тип данных, а отдельные объекты – как данные этого типа. Определение программистом собственных классов объектов должно позволить описывать конкретную задачу в терминах ее предметной области.

Целью курсовой работы является разработка программы «Разновидности зверей в мире».

Для достижения данной цели поставлены следующие задачи:

* изучить основы объектно-ориентированного программирования;
* выбрать предметную область;
* создать репозиторий для разработки приложения;
* создать диаграмму классов выбранной предметной области;
* реализовать описанный диаграммой классов функционал.

Вопросы, подлежащие разработке:

* изучение методов и средства создания приложений на основе объектно-ориентированного подхода;
* использование инструментов для создания диаграммы классов предметной области;
* изучение средств контроля версий и способы их использования;
* изучение основных принципов ООП.

Систематическое применение объектно-ориентированного подхода позволяет разрабатывать хорошо структурированные, надежные в эксплуатации, достаточно просто модифицируемые программные системы.

# 1 описание Предметной области программы

Программа «Разновидности зверей в Мире» предназначена для обработки предметной области «Звери». Данная предметная область – это неотъемлемая часть жизни человека. Существует множество различных зверей, их внешний вид, манера поведения и характерные черты отличаются друг от друга. И поэтому человеку, для правильного представления об этих зверях, необходимо знать как общие черты, так и отличия в них.

В ходе разработки были использованы принципы и возможности объектно-ориентированного программирования. Программа была реализована в среде разработки IntelliJ IDEA Community Edition 2017.2.4 на языке объектно-ориентированного программирования Java.

# 2 Диаграмма классов

Диаграмма классов определяет типы классов системы и различного рода статические связи, которые существуют между ними. На диаграммах классов изображаются также атрибуты классов, операции классов и ограничения, которые накладываются на связи между классами.

Диаграмма классов UML - это граф, узлами которого являются элементы статической структуры проекта (классы), а дугами - отношения между узлами (наследование).

.

cat

Private color;  
private volosi;  
privatelapki;

public zvyki:мяукает  
public hodit:аккуратно

dog

Private color;  
private volosi;  
privatelapki;

public zvyki:гавкает  
public hodit:бегает

Zveri

Private color;  
private volosi;  
privatelapki;

public zvyki;  
public hodit;

mays

Private color;  
private volosi;  
privatelapki;

public zvyki:пищит  
public hodit:бесшумно

Рисунок 1 – Диаграмма классов «Звери»

# 3 Код программы

## 3.1 Класс родителя «zveri»

public class zveri {

private String color;

private boolean volosi;

private int lapki;

public String zvyki;

public String hodit;

public zveri(){}

public zveri(boolean vol, String col, int lki, String zv, String hod){

this.color=col;

this.volosi=vol;

this.lapki=lki;

this.zvyki=zv;

this.hodit=hod;

}

public String getColor() {

return color;

}

public void setColor(String color) {

this.color = color;

}

public int getLapki () {

return lapki;

}

public void lapki (int lapki) {

this.lapki = lapki;

}

public void setVolosi(boolean volosi) {

this.volosi = volosi;

}

public boolean isVolosi() {

return volosi;

}

public String getZvyki() { return zvyki; }

public String getHodit() { return hodit; }

};

## 3.2 Класс наследника «cat»

public class Cat extends zveri {

private String color;

private boolean volosi;

private int lapki;

public String zvyki;

public String hodit;

public Cat(boolean vol, String col, int lki, String zv, String hod, boolean raz ) {

super(vol, col, lki, zv,hod);

this.color = col;

this.volosi=vol;

this.lapki=lki;

this.zvyki=zv;

this.hodit=hod;

}

public Cat(boolean b, String трехшерстная, int i, String мяукает, String аккуратно) {

}

}

## 3.3 Класс наследника «mays»

public class Mays extends zveri {

private String color;

private boolean volosi;

private int lapki;

public String zvyki;

public String hodit;

public Mays(boolean vol, String col, int lki, String zv, String hod ) {

super(vol, col, lki, zv,hod);

this.color = col;

this.volosi=vol;

this.lapki=lki;

this.zvyki=zv;

this.hodit=hod;

}

}

## 3.4 Класс наследника «dog»

class Dog extends zveri {

private String color;

private boolean volosi;

private int lapki;

public String zvyki;

public String hodit;

public Dog(boolean vol, String col, int lki, String zv, String hod ) {

super(vol, col, lki, zv,hod);

this.color = col;

this.volosi=vol;

this.lapki=lki;

this.zvyki=zv;

this.hodit=hod;

}

}

## 3.5 Выполнение программы «proverka»

public class proverka {

public static void main(String[] args){

Cat cat = new Cat(true, "Трехшерстная", 4, "Мяукает", "Аккуратно" );

System.out.println("У кошки есть волосеной покров ? " + cat.isVolosi());

System.out.println("Какого цвета эта кошка? " + cat.getColor());

System.out.println("Сколько лап у кошки? " + cat.getLapki());

System.out.println("Какой звук издает кошка? " + cat.getZvyki());

System.out.println("Как передвигается кошка? " + cat.getHodit());

System.out.println();

zveri mays = new Mays(true, "Серая", 4, "Пищит", "Бесшумно");

System.out.println("Есть ли волосеной покров у мышки? " + mays.isVolosi());

System.out.println("Какого цвета мышка? " + mays.getColor());

System.out.println("Сколько лап у мышки? " + mays.getLapki());

System.out.println("Какой звук издает мышка? " + mays.getZvyki());

System.out.println("Как передвигается мышки? " + mays.getHodit());

System.out.println();

zveri dog = new Dog(true, "Коричневая", 4, "Гавкает", "Бегает");

System.out.println("Есть ли волосеной покров у мышки? " + mays.isVolosi());

System.out.println("Какого цвета мышка? " + dog.getColor());

System.out.println("Сколько лап у мышки? " + dog.getLapki());

System.out.println("Какой звук издает мышка? " + dog.getZvyki());

System.out.println("Как передвигается мышки? " + dog.getHodit());

}

}

# Заключение

В результате разработки данной программы были решены следующие задачи:

* изучены основы объектно-ориентированного программирования;
* выбрана предметная область;
* создан репозиторий для разработки приложения;
* создана диаграмма классов для выбранной предметной области;
* реализован описанный диаграммой классов функционал.

Данная программа легка для освоения и не требует обладание большими навыками работы на компьютере. Все операции программы выполняются при помощи диалогового окна.

Итогом курсовой работы можно считать закрепление знаний в области использования принципов и возможностей объектно-ориентированного программирования, получение практических навыков программирования на языке Java и создании программ с его использованием.

# Список литературы

1. Давыдов В.И. Кунгурцев А.Б. Объектно-ориентированное программирование.
2. Философия Java. Б.Эккель
3. Язык UML. Руководство пользователя. Г.Буч, Д.Рамбо
4. https://studfiles.net/preview/2806643/page:4/

# Приложение демонстрация выполнения программы

