**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 №4

по дисциплине: «Визуальные средства разработки программных приложений»

на тему: «Построение модели программной системы»

Выполнил: студент гр. ИТИ-31

Ковшаров Г. Ю.  
Принял: преподаватель

Ястребов А. А.

Гомель 2024

**Цель работы**: научиться создавать модели программных систем и реализовывать их с использованием языка программирования Java.

**Задание:**

1. Определить иерархию классов (в соответствии с вариантом).

2. Определить в классе статическую компоненту - ссылку на начало связанного списка объектов и статическую функцию для просмотра списка.

3. Реализовать классы.

4. Написать демонстрационную программу, в которой создаются объекты различных классов и помещаются в список, после чего список просматривается.

5. Реализовать вариант, когда объект добавляется в список при создании, т.е. в конструкторе.

Перечень классов:

14) млекопитающие, парнокопытные, птицы, животное;

**Ход работы:**

Была реализована иерархия классов животных, визуализация которой представлена на рисунке 1.

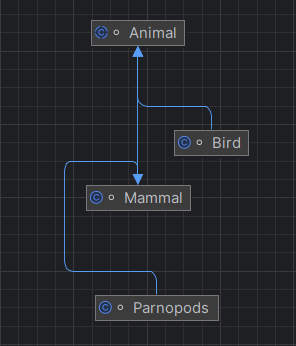


Рисунок 1 – Иерархия классов

Результат работы программы представлен на рисунке 2.

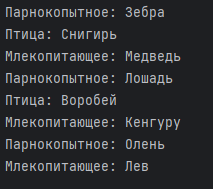


Рисунок 2 – Результат работы программы

Текст разработанных классов на языке *Java* представлен в приложении А.

**Выводы:** в ходе проделанной работы получены навыки создания модели программных систем и реализовывал их с использованием языка программирования Java. Так же было создано приложение которое при помощи абстрактного класса и статического поля реализует связный список.

**ПРИЛОЖЕНИЕ А**

(обязательное)

**Текст классов**

**Animal.java:**

abstract class Animal {

private static Animal head = null;

protected Animal next;

protected String name;

public Animal(String name) {

this.name = name;

this.next = head;

head = this;

}

public abstract void show();

public static void showList() {

Animal current = head;

while (current != null) {

current.show();

current = current.next;

}

}

}

**Bird.java:**

class Bird extends Animal {

public Bird(String name) {

super(name);

}

@Override

public void show() {

System.out.println("Птица: " + name);

}

}

**Mammal.java:**

class Mammal extends Animal {

public Mammal(String name) {

super(name);

}

@Override

public void show() {

System.out.println("Млекопитающее: " + name);

}

}

**Parnopods.java:**

class Parnopods extends Mammal {

public Parnopods(String name) {

super(name);

}

@Override

public void show() {

System.out.println("Парнокопытное: " + name);

}

}

**Main.java:**

public class Main {

public static void main(String[] args) {

new Mammal("Лев");

new Parnopods("Олень");

new Mammal("Кенгуру");

new Bird("Воробей");

new Parnopods("Лошадь");

new Mammal("Медведь");

new Bird("Снигирь");

new Parnopods("Зебра");

Animal.showList();

}

}