

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт Космических и информационных технологий

институт

Кафедра «Информатика»

кафедра

ОТЧЕТ О ПРАКТИЧЕСКОЙ РАБОТЕ №1

Java Core, Наследование

тема

Преподаватель

подпись, дата

А.С. Черниговский

инициалы, фамилия

Студент КИ18-17/16 031830504

номер группы, зачетной книжки

подпись, дата

Е.В. Железкин

инициалы, фамилия

Красноярск 2020

СОДЕРЖАНИЕ

Содержание	2
1 Цель работы	3
2 Задача работы	3
3 Ход работы.....	3
4 Вывод.....	4

1 Цель работы

Ознакомиться с механизмом наследования в языке Java. Повторить основные языковые конструкции языка Java.

2 Задача работы

Для каждого варианта имеется набор из четырех сущностей. Необходимо выстроить иерархию наследования. В каждом классе (базовом и производных) должно быть минимум одно числовое и одно текстовое поле. При вводе числовых параметров обязательна проверка на число и на диапазон (даже если число может быть любое, проверку необходимо реализовать).

Для всех классов должны быть реализованы конструкторы (по умолчанию, с параметрами), методы equals(), hashCode(), toString(); Реализовать консольное Java-приложение, которое имеет простейшее пользовательское меню, состоящее как минимум из следующих пунктов:

- Добавить новый элемент. (Элементы должны добавляться в коллекцию элементов типа базового класса. Необходимо предусмотреть возможность добавления любого объекта производного класса в данную коллекцию.)

- Удалить элемент по индексу.

- Вывод всех элементов в консоль.

- Сравнение двух элементов на равенство (по индексам).

- Завершение работы приложения.

Вариант 9 - Животное, млекопитающее, парнокопытное, птица.

3 Ход работы

1) Выполнено задание (Приложение А):

Choose option:

- 1 - Add
- 2 - Delete
- 3 - Clear
- 4 - Show
- 5 - Compare
- 0 - Exit

Input a number: 5

Enter the number of the first element:

4

Enter the number of the second element:

5

Objects are equivalent!

Choose option:

- 1 - Add
- 2 - Delete
- 3 - Clear
- 4 - Show
- 5 - Compare
- 0 - Exit

Input a number: 4

1) Animal{Species='DFG', Test=567}

2) Mammal{Species='ASD', Test=321, Subspecies='QWE', SubTest=123}

3) Bird{Species='ZXC', Test=321, Subspecies='QWE', SubTest=123}

4) ClovenHoofed{Species='ABC', Test=666, Subspecies='ABC', SubTest=666, Name='ABC', legs=4}

5) Animal{Species='Unknown', Test=0}

5) Animal{Species='Unknown', Test=0}

5) Animal{Species='Unknown', Test=0}

Рисунок 1 – Результат выполнения

4 Вывод

В ходе данной лабораторной работы был изучен механизм наследования, основные языковые конструкции языка Java.

ПРИЛОЖЕНИЕ А

Листинг 1 – файл Main.java

```
package com.superzloyuser;

import com.superzloyuser.entities.*;

import java.util.ArrayList;
import java.util.Scanner;

public class Main {

    public static void main(String[] args) {

        var testList = new ArrayList<Animal>();

        testList.add(new Animal("DFG", 567));
        testList.add(new Mammal("ASD", 321, "QWE", 123));
        testList.add(new Bird("ZXC", 321, "QWE", 123));
        testList.add(new ClovenHoofed("ABC", 666, "ABC", 666, "ABC", 4));
        testList.add(new Animal());
        testList.add(new Animal());
        testList.add(new Animal());

        boolean flag = true;

        while(flag){
            System.out.println("Choose option:\n1 - Add\n2 - Delete\n3 - Clear\n4 - Show\n5 - Compare\n0 - Exit\n");
            Scanner in = new Scanner(System.in);
            System.out.print("Input a number: ");
            var option = in.nextInt();

            switch (option){
                case 1:
                    System.out.println("Input class name:");
                    var className = in.next();
                    switch (className){
                        case "Animal":
                            var tempSpecies = in.next();
                            var tempTest = in.nextInt();
                            testList.add(new Animal(tempSpecies, tempTest));
                            break;
                        case "Mammal":
                            tempSpecies = in.next();
                            tempTest = in.nextInt();
                            var tempSubspecies = in.next();
                            var tempSubTest = in.nextInt();
                            testList.add(new Mammal(tempSpecies, tempTest, tempSubspecies, tempSubTest));
```

```

        break;
    case "ClovenHoofed":
        tempSpecies = in.next();
        tempTest = in.nextInt();
        tempSubspecies = in.next();
        tempSubTest = in.nextInt();
        var tempName = in.next();
        var tempLegs = in.nextInt();
        testList.add(new ClovenHoofed(tempSpecies,
tempTest, tempSubspecies, tempSubTest, tempName, tempLegs));
        break;
    case "Bird":
        tempSpecies = in.next();
        tempTest = in.nextInt();
        tempSubspecies = in.next();
        tempSubTest = in.nextInt();
        testList.add(new Bird(tempSpecies, tempTest,
tempSubspecies, tempSubTest));
        break;
    default:
        System.out.println("Invalid class name!");
        continue;
    }
    break;
case 2:
    System.out.println("Input element number:");
    var numForDel = in.nextInt();
    try {
        testList.remove(numForDel - 1);
    }
    catch (Exception e){
        System.out.println(e.getMessage());
    }
    break;
case 3:
    testList.clear();
    break;
case 4:
    for (Animal animal : testList) {
        System.out.println((testList.indexOf(animal) + 1) +
") " + animal.toString());
    }
    break;
case 5:
    if (testList.size() > 1) {
        System.out.println("Enter the number of the first
element: ");

        var firstEl = in.nextInt();

        System.out.println("Enter the number of the second
element: ");

        var secondEl = in.nextInt();

```

```

        try {
            if
(testList.get(firstEl).equals(testList.get(secondEl)))
                System.out.println("Objects are
equivalent!");
            else
                System.out.println("Objects are not
equivalent!");
        }
        catch (Exception e){
            System.out.println(e.getMessage());
        }
    }
    break;
case 0:
    flag = false;
    break;
}
}
}
}
}

```

Листинг 2 – файл entities/Animal.java

```

package com.superzloyuser.entities;

import java.util.Objects;

public class Animal {

    private String Species;
    private int Test;

    public Animal(){
        this.Species = "Unknown";
        this.Test = 0;
    }

    public Animal(String species, int test){
        this.Species = species;
        this.Test = test;
    }

    public String getSpecies() {
        return Species;
    }

    public void setSpecies(String species) {

```

```

        Species = species;
    }

    public int getTest() {
        return Test;
    }

    public void setTest(int test) {
        Test = test;
    }

    @Override
    public boolean equals(Object o) {
        if (this == o) return true;
        if (!(o instanceof Animal)) return false;
        Animal animal = (Animal) o;
        return Test == animal.Test &&
            Species.equals(animal.Species);
    }

    @Override
    public int hashCode() {
        return Objects.hash(Species, Test);
    }

    @Override
    public String toString() {
        return "Animal{" +
            "Species='" + Species + '\'' +
            ", Test=" + Test +
            '}';
    }
}

```

Листинг 3 – файл entities/Mammal.java

```

package com.superzloyuser.entities;

import java.util.Objects;

public class Mammal extends Animal {

    private String Subspecies;
    private int SubTest;

    public Mammal() {
        Subspecies = "Unknown";
        SubTest = 0;
    }
}

```



```

    public Mammal(String species, int test, String subspecies, int subTest)
    {
        super(species, test);
        Subspecies = subspecies;
        SubTest = subTest;
    }

    public String getSubspecies() {
        return Subspecies;
    }

    public void setSubspecies(String subspecies) {
        Subspecies = subspecies;
    }

    public int getSubTest() {
        return SubTest;
    }

    public void setSubTest(int subTest) {
        SubTest = subTest;
    }

    @Override
    public boolean equals(Object o) {
        if (this == o) return true;
        if (!(o instanceof Mammal)) return false;
        if (!super.equals(o)) return false;
        Mammal mammal = (Mammal) o;
        return SubTest == mammal.SubTest &&
            Subspecies.equals(mammal.Subspecies);
    }

    @Override
    public int hashCode() {
        return Objects.hash(super.hashCode(), Subspecies, SubTest);
    }

    @Override
    public String toString() {
        return "Mammal{" +
            "Species='" + super.getSpecies() + '\'' +
            ", Test=" + super.getTest() +
            ", Subspecies='" + Subspecies + '\'' +
            ", SubTest=" + SubTest +
            '}';
    }
}

```

Листинг 4 – файл entities/ ClovenHoofed.java

```

package com.superzloyuser.entities;

import java.util.Objects;

public class ClovenHoofed extends Mammal{

    private String Name;
    private int legs;

    public ClovenHoofed() {
        this.Name = "Unknown";
        this.legs = 4;
    }

    public ClovenHoofed(String species, int test, String subspecies, int
subTest, String name, int legs) {
        super(species, test, subspecies, subTest);
        this.Name = name;
        this.legs = legs;
    }

    @Override
    public String getSpecies() {
        return Name;
    }

    public void setName(String name) {
        Name = name;
    }

    public int getLegs() {
        return legs;
    }

    public void setLegs(int legs) {
        this.legs = legs;
    }

    @Override
    public boolean equals(Object o) {
        if (this == o) return true;
        if (!(o instanceof ClovenHoofed)) return false;
        if (!super.equals(o)) return false;
        ClovenHoofed that = (ClovenHoofed) o;
        return legs == that.legs &&
            Name.equals(that.Name);
    }

    @Override
    public int hashCode() {
        return Objects.hash(super.hashCode(), Name, legs);
    }
}

```

```

@Override
public String toString() {
    return "ClovenHoofed{" +
        "Species='" + super.getSpecies() + '\'' +
        ", Test=" + super.getTest() +
        ", Subspecies='" + super.getSubspecies() + '\'' +
        ", SubTest=" + super.getSubTest() +
        ", Name='" + Name + '\'' +
        ", legs=" + legs +
        '}';
}
}

```

Листинг 5 – файл entities/ Bird.java

```

package com.superzloyuser.entities;

import java.util.Objects;

public class Bird extends Animal{

    private String Subspecies;
    private int SubTest;

    public Bird() {
        this.Subspecies = "Unknown";
        this.SubTest = 0;
    }

    public Bird(String Name, int Test, String subspecies, int subTest) {
        super(Name, Test);
        Subspecies = subspecies;
        SubTest = subTest;
    }

    public String getSubspecies() {
        return Subspecies;
    }

    public void setSubspecies(String subspecies) {
        Subspecies = subspecies;
    }

    public int getSubTest() {
        return SubTest;
    }

    public void setSubTest(int subTest) {
        SubTest = subTest;
    }
}

```

```

@Override
public boolean equals(Object o) {
    if (this == o) return true;
    if (!(o instanceof Bird)) return false;
    if (!super.equals(o)) return false;
    Bird bird = (Bird) o;
    return SubTest == bird.SubTest &&
        this.Subspecies.equals(bird.Subspecies);
}

@Override
public int hashCode() {
    return Objects.hash(super.hashCode(), Subspecies, SubTest);
}

@Override
public String toString() {
    return "Bird{" +
        "Species='" + super.getSpecies() + '\'' +
        ", Test=" + super.getTest() +
        ", Subspecies='" + Subspecies + '\'' +
        ", SubTest=" + SubTest +
        '}';
}
}

```