

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 КИ18-16Б

номер группы, зачетной книжки

подпись, дата

К.И Оглеzneв

инициалы, фамилия

Красноярск 2020

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 20

3 Ход работы

- 1) Выполнено задание:

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

Input a number: 4
0) Technic{Country='DFG', Year=567}
1) Technic{Country='DFG', Year=567}
2) HouseholdTech{Species='ASD', Test=321, Brand='QWE', Price=123}
3) AutomotiveTech{Species='ZXC', Test=321, Brand='QWE', Price=123}
4) GardenTech{Species='ABC', Test=666, Brand='ABC', Price=666}
5) Technic{Country='Unknown', Year=0}
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:
0
Enter the number of the second element:
1
Objects are equivalent!
```

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

```
import variant20.*;
import java.util.ArrayList;
import java.util.Scanner;

public class Main {

    private static final int ERROR = -1;
    private static final int CURRENT_YEAR = 2020;

    public static int checkNum(Scanner in) {
        if (in.hasNextInt())
            return in.nextInt();
        else {
            System.out.println(""" +
                "-----\n" +
                " Input error \n" +
                "-----");
            return ERROR;
        }
    }

    public static void main(String[] args) {

        var testList = new ArrayList<Technic>();

        boolean flag = true;
        while (flag) {
            System.out.println("Choose option:\n" +
                "1 - Add\n" +
                "2 - Delete\n" +
                "3 - Clear\n" +
                "4 - Show\n" +
                "5 - Compare\n" +
                "0 - Exit\n");
            Scanner in = new Scanner(System.in);
            System.out.print("Input a number: ");
            int option = checkNum(in);
            switch (option) {
                case 1:
                    System.out.println("Input class name:");
                    var className = in.next();
                    String tempCountry;
                    int tempYear;
                    String tempBrand;
                    int tempPrice;
                    switch (className) {
                        case "Technic":
                            tempCountry = in.next();
                            tempYear = checkNum(in);
                            if (tempYear <= CURRENT_YEAR)
                                testList.add(new Technic(tempCountry, tempYear));
                            break;
                        case "HouseholdTech":
                            tempCountry = in.next();
```

```

        tempYear = checkNum(in);
        if (tempYear == ERROR) break;
        tempBrand = in.next();
        tempPrice = checkNum(in);
        if ((tempYear <= CURRENT_YEAR) && tempPrice > 0)
            testList.add(new HouseholdTech(tempCountry, tempYear, tempBrand, tempPrice));
        break;
    case "GardenTech":
        tempCountry = in.next();
        tempYear = checkNum(in);
        if (tempYear == ERROR) break;
        tempBrand = in.next();
        tempPrice = checkNum(in);
        if ((tempYear <= CURRENT_YEAR) && tempPrice > 0)
            testList.add(new GardenTech(tempCountry, tempYear, tempBrand, tempPrice));
        break;
    case "AutomotiveTech":
        tempCountry = in.next();
        tempYear = checkNum(in);
        if (tempYear == ERROR) break;
        tempBrand = in.next();
        tempPrice = checkNum(in);
        if ((tempYear <= CURRENT_YEAR) && tempPrice > 0)
            testList.add(new AutomotiveTech(tempCountry, tempYear, tempBrand, tempPrice));
        break;
    default:
        System.out.println("
        -----\n" +
        " Invalid class name! \n" +
        "-----");
        continue;
    }
    break;
case 2:
    System.out.println("Input element number:");
    int numForDel = checkNum(in);
    if (numForDel == ERROR) break;
    try {
        testList.remove(numForDel - 1);
    } catch (Exception e) {
        System.out.println(e.getMessage());
    }
    break;
case 3:
    testList.clear();
    break;
case 4:
    for (int i = 0; i < testList.size(); i++) {
        System.out.println(i + " " + testList.get(i));
    }
    break;
case 5:
    if (testList.size() > 1) {
        System.out.println("Enter the number of the first element: ");
        int firstEl = checkNum(in);
        if (firstEl == ERROR) break;
        System.out.println("Enter the number of the second element: ");
        int secondEl = checkNum(in);
        if (secondEl == ERROR) break;
        if (testList.get(firstEl).equals(testList.get(secondEl)))
            System.out.println("Objects are equivalent!");
        else
            System.out.println("Objects are not equivalent!");
    }

```

```

    }
    break;
case 0:
    flag = false;
    break;
}
}
}
}

```

Листинг 2 – файл variant20/Technic .java

```

package variant20;
import java.util.Objects;
public class Technic {
    private String country;
    private int year;

    public Technic() {
        this.country = "Unknown";
        this.year = 0;
    }

    public Technic(String country, int year) {
        this.country = country;
        this.year = year;
    }

    public String getCountry() {
        return country;
    }

    public void setCountry(String country) {
        country = country;
    }

    public int getYear() {
        return year;
    }

    public void setYear(int year) {
        this.year = year;
    }

    @Override
    public boolean equals(Object obj) {
        if (this == obj) return true;
        if (!(obj instanceof Technic)) return false;
        Technic Technic = (Technic) obj;
        return year == Technic.year &&
            country.equals(Technic.country);
    }

    @Override
    public int hashCode() {

```

```

        return Objects.hash(country, year);
    }

    @Override
    public String toString() {
        return "Technic{" +
            "Country='" + country + '\'' +
            ", Year=" + year +
            '}';
    }
}

```

Листинг 3 – файл variant20/AutomotiveTech.java

```

package variant20;
import java.util.Objects;

public class AutomotiveTech extends Technic {
    private String brand;
    private int price;

    public AutomotiveTech() {
        brand = "Unknown";
        price = 0;
    }

    public AutomotiveTech(String text, int num, String brand, int price) {
        super(text, num);
        this.brand = brand;
        this.price = price;
    }

    public String getBrand() {
        return brand;
    }

    public void setBrand(String brand) {
        this.brand = brand;
    }

    public int getPrice() {
        return price;
    }

    public void setPrice(int price) {
        this.price = price;
    }

    @Override
    public boolean equals(Object obj) {
        if (this == obj) return true;
        if (!(obj instanceof AutomotiveTech)) return false;
        if (!super.equals(obj)) return false;
        AutomotiveTech AutomotiveTech = (AutomotiveTech) obj;
        return price == AutomotiveTech.price &&
            this.brand.equals(AutomotiveTech.brand);
    }
}

```

```

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

@Override
public String toString() {
    return "AutomotiveTech{" +
        "Species=" + super.getCountry() + "\" +
        ", Test=" + super.getYear() +
        ", Brand=" + brand + "\" +
        ", Price=" + price +
        '"';
}
}

```

Листинг 4 – файл variant20/GardenTech.java

```

package ;
import java.util.Objects;

public class GardenTech extends Technic {
    private String brand;
    private int price;

    public GardenTech() {
        brand = "Unknown";
        price = 0;
    }

    public GardenTech(String text, int num, String brand, int price) {
        super(text, num);
        this.brand = brand;
        this.price = price;
    }

    public String getBrand() {
        return brand;
    }

    public void setBrand(String brand) {
        this.brand = brand;
    }

    public int getPrice() {
        return price;
    }

    public void setPrice(int price) {
        this.price = price;
    }

    @Override
    public boolean equals(Object obj) {
        if (this == obj) return true;
        if (!(obj instanceof GardenTech)) return false;
        if (!super.equals(obj)) return false;
        GardenTech GardenTech = (GardenTech) obj;
    }
}

```



```

        return price == GardenTech.price && this.brand.equals(GardenTech.brand);
    }

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

    @Override
    public String toString() {
        return "GardenTech{" +
            "Species=" + super.getCountry() + "\" +
            ", Test=" + super.getYear() +
            ", Brand=" + brand + "\" +
            ", Price=" + price +
            "'}";
    }
}

```

Листинг 5 – файл variant20/HouseholdTech.java

```

package variant20;
import java.util.Objects;

public class HouseholdTech extends Technic {
    private String brand;
    private int price;

    public HouseholdTech() {
        brand = "Unknown";
        price = 0;
    }

    public HouseholdTech(String text, int num, String brand, int price) {
        super(text, num);
        this.brand = brand;
        this.price = price;
    }

    public String getBrand() {
        return brand;
    }

    public void setBrand(String brand) {
        this.brand = brand;
    }

    public int getPrice() {
        return price;
    }

    public void setPrice(int price) {
        this.price = price;
    }

    @Override
    public boolean equals(Object obj) {
        if (this == obj) return true;
        if (!(obj instanceof HouseholdTech)) return false;
    }
}

```

```

        if (!super.equals(obj)) return false;
        HouseholdTech HouseholdTech = (HouseholdTech) obj;
        return price == HouseholdTech.price &&
            this.brand.equals(HouseholdTech.brand);
    }

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

    @Override
    public String toString() {
        return "HouseholdTech{" +
            "Species=" + super.getCountry() + "\" +
            ", Test=" + super.getYear() +
            ", Brand=" + brand + "\" +
            ", Price=" + price +
            "'";
    }
}

```