

Министерство образования Республики Беларусь
Учреждение образования
«Брестский государственный технический университет»
Кафедра ИИТ

ОТЧЕТ
по лабораторной работе №5
по дисциплине СПП

Выполнил:
студ. гр.ПО-5
Харкевич
Д.А.

Проверил:
Крощенко А.А.
Ст.преп. кафедры ИИТ

Брест 2021

Цель работы: приобрести практические навыки в области объектно-ориентированного проектирования.

Вариант 13

Задание 1. Реализовать абстрактные классы или интерфейсы, а также наследование и полиморфизм для следующих классов:

interface Транспортное Средство ← abstract class Общественный Транспорт ← class Троллейбус

Реализация:

Main.java

```
package lab;

public class Main {

    public static void main(String[] args) {
        Trolleybus trbus1 = new Trolleybus("4A51", 54, 2, 1);
        Trolleybus trbus2 = new Trolleybus("8C51", 50, 50, 2.5);
        trbus1.show();
        trbus2.showAll();
    }
}

interface Transport {
    void show();
    void showAll();
}

abstract class PublicTransport implements Transport{
    private String number;
    private int seats;

    public PublicTransport(String _number, int _seats) {
        number = _number;
        seats = _seats;
    }

    public String getNumber() {
        return number;
    }

    public int getSeats() {
        return seats;
    }
}

class Trolleybus extends PublicTransport {
    private int route;
    private double fare;
    public Trolleybus (String _number, int _seats, int _route, double _fare) {
        super(_number, _seats);
        route = _route;
        fare = _fare;
    }
    @Override
    public void show() {
        System.out.println("Trolleybus №" + super.getNumber() + " works on route " + route);
    }
}
```

```

@Override
public void showAll() {
    System.out.println("Trolleybus №" + super.getNumber() + " works on route"
        + route +
        ". It costs " + fare + "$ and has " + super.getSeats() + "
seats." );
}
}

```

Результат работы программы

```

Trolleybus №4A51 works on route 2
Trolleybus №8C51 works on route 50. It costs 2.5$ and has 50 seats.

Process finished with exit code 0

```

Задание 2. В следующих заданиях требуется создать суперкласс (абстрактный класс, интерфейс) и определить общие методы для данного класса. Создать подклассы, в которых добавить специфические свойства и методы. Часть методов переопределить. Создать массив объектов суперкласса и заполнить объектами подклассов. Объекты подклассов идентифицировать конструктором по имени или идентификационному номеру. Использовать объекты подклассов для моделирования реальных ситуаций и объектов.

Создать суперкласс Домашнее животное и подклассы Собака, Кошка, Попугай. С помощью конструктора установить имя каждого животного и его характеристики.

Реализация:

Main.java

```

package lab;
import java.util.ArrayList;

public class Main {

    public static void main(String[] args) {
        ArrayList<Pet> pets = new ArrayList<>();
        pets.add(new Cat("Max", "Munchkin", 3));
        pets.add(new Dog("Isaak", "Husky", 7));
        pets.add(new Parrot("Mike", "Amazon", 4));

        for (Pet i: pets)
            i.show();
    }
}

abstract class Pet {
    private String name;
    private String breed;
    private int age;
    public Pet (String _name, String _breed, int _age) {
        name = _name;
        breed = _breed;
        age = _age;
    }
    public String getName() {
        return name;
    }
}

```

```

    public String getBreed() {
        return breed;
    }
    public int getAge() {
        return age;
    }

    public abstract void show();
}

class Dog extends Pet {
    private String voice = "Aww-Aww";
    public Dog (String _name, String _breed, int _age) {
        super(_name, _breed, _age);
    }
    public void show () {
        System.out.println(voice + "\nHi, my name is " + super.getName() + ". My
breed is " + super.getBreed() +
            " and I'm " + super.getAge() + " years old.\n");
    }
}

class Cat extends Pet {
    private String voice = "Meowww";
    public Cat (String _name, String _breed, int _age) {
        super(_name, _breed, _age);
    }
    public void show () {
        System.out.println(voice + "\nHi, my name is " + super.getName() + ". My
breed is " + super.getBreed() +
            " and I'm " + super.getAge() + " years old.\n");
    }
}

class Parrot extends Pet {
    private String voice = "Have a nice day";
    public Parrot (String _name, String _breed, int _age) {
        super(_name, _breed, _age);
    }
    public void show () {
        System.out.println(voice + "\nMy name is " + super.getName() + " and I'm
a talking parrot. My breed is " + super.getBreed() +
            " and I'm " + super.getAge() + " years old.\n");
    }
}

```

Результат работы программы:

Meowww

Hi, my name is Max. My breed is Munchkin and I'm 3 years old.

Aww-Aww

Hi, my name is Isaak. My breed is Husky and I'm 7 years old.

Have a nice day

My name is Mike and I'm a talking parrot. My breed is Amazon and I'm 4 years old.

Process finished with exit code 0

Задание 3. В задании 3 ЛР №4, где возможно, заменить объявления суперклассов объявлениями абстрактных классов или интерфейсов

Реализация:

```
package lab;

public class Person {
    private String name;
    private String surname;

    public Person(String _name, String _surname) {
        name = _name;
        surname = _surname;
    }
}

public class Abiturient extends Person{
    public Abiturient (String _name, String _surname) {
        super(_name, _surname);
    }
}

public class Teacher extends Person{
    private Exam.Subjects subject;
    public Teacher (String _name, String _surname, Exam _ex) {
        super(_name, _surname);
        subject = _ex.getSubject();
    }
}
```

Вывод: приобрели практические навыки в области объектно-ориентированного проектирования