Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

Лабораторная работа №5

Выполнила:

Студент 3 курса

Группы ПО-5

Нерода А.А.

Проверил:

Крощенко А.А.

Брест 2021

# Вариант-10

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

**Задание 1.** interface Диск ← abstract class Директория ← class Файл.

**Код программы**

public class DiskRunner {

    public static void main(String[] args) {

        File one = new File("Data", 239);

        File two = new File("Git", 25);

        one.info();

        two.info();

    }

}

interface Disk {

    String getName();

    int getSize();

}

abstract class Dir implements Disk {

    private String name;

    private int size;

    public Dir(String name, int size) {

        this.name = name;

        this.size = size;

    }

    @Override

    public String getName() { return name; }

    @Override

    public int getSize() { return size; }

    public abstract void info();

}

class File extends Dir {

    public File(String name, int size) {

        super(name, size);

    }

    @Override

    public void info() {

        System.out.println("Filename: " + super.getName() + " || filesize: " + super.getSize());

    }

}

**Результаты работы**

Filename: Data || filesize: 239

Filename: Git || filesize: 25

**Задание 2.** Создать суперкласс Учащийся и подклассы Школьник и Студент. Создать массив объектов суперкласса и заполнить этот массив объектами. Показать отдельно студентов и школьников.

**Код программы**

import java.util.Scanner;

public class Learner {

    public static void main(String[] args)

    {

        Scanner in = new Scanner(System.in);

        System.out.print("Enter number of pupils: ");

        int n = in.nextInt(); in.nextLine();

        Pupil[] a = new Pupil[n];

        int max = 1, min = 0;

        for (int x = 0; x < n; x++) {

            int randNumber = (int)(Math.random()\*((max-min)+1))+min;

            if (randNumber == 0) {

                a[x] = new Student(x + 1);

            }

            else {

                a[x] = new Schoolboy(x + 1);

            }

        }

        for (int x = 0; x < n; x++) {

            if (a[x].num == 1) {

                System.out.print(a[x].n + ". ");

                a[x].type();

            }

            else if (a[x].num == 2) {

                System.out.print(a[x].n + ". ");

                a[x].type();

            }

        }

        in.close();

    }

}

class Pupil

{

        public int n;

        public int num;

        public Pupil(int n, int num) {

            this.n = n;

            this.num = num;

        }

        public void type() {

            System.out.println("Pupil");

        }

}

class Schoolboy extends Pupil

{

        public Schoolboy(int x) {

            super(x, 1);

        }

        @Override

        public void type()

        {

            System.out.println("Schoolboy");

        }

}

class Student extends Pupil

{

        public Student(int x) {

            super(x, 2);

        }

        @Override

        public void type() {

            System.out.println("Student");

        }

}

**Результаты работы**

Enter number of pupils: 5

1. Student

2. Schoolboy

3. Student

4. Schoolboy

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

**Код программы**

import java.util.ArrayList;

import java.util.Arrays;

import java.util.List;

public class Solution {

    public static void main(String[] args) {

        Bus bus1 = new Bus(1);

        Bus bus2 = new Bus(38);

        Bus bus3 = new Bus(21);

        Trolleybus trolley1 = new Trolleybus(4);

        Trolleybus trolley2 = new Trolleybus(5);

        Trolleybus trolley3 = new Trolleybus(228);

        Route route = new Route("Больница - Красный двор",4,8);

        route.add(bus1,bus2,bus3, trolley1, trolley2);

        route.addAddition(trolley3);

        route.start();

    }

    public static class Route {

        String name;

        Checker checker = new Checker();

        double distance;

        int numberOfStops;

        double interval = (distance/numberOfStops)\*60;//мин

        public Route(String name,double distance, int numberOfStops) {

            this.name = name;

            this.distance = distance;

            this.numberOfStops = numberOfStops;

        }

        void add(CityVehicle...vehicles){

            for(CityVehicle temp: vehicles){

                this.checker.vehicles.add(new VehicleByRoute(temp,distance,numberOfStops));

            }

        }

        void addAddition(CityVehicle...vehicles){

            for(CityVehicle temp: vehicles){

                this.checker.vehiclesAddition.add(new VehicleByRoute(temp,distance,numberOfStops));

            }

        }

        void start(){

            new Thread(checker).start();

            synchronized (checker) {

                while (!checker.vehicles.isEmpty()) {

                    new Thread(checker.vehicles.get(0)).start();

                    checker.vehicles.remove(0);

                    try {

                        Thread.sleep((long)(100\*checker.intervalMultiply));

                    } catch (InterruptedException e) {

                        e.printStackTrace();

                    }

                }

            }

        }

        private class VehicleByRoute implements Runnable{

            CityVehicle vehicle;

            double distance;

            int numberOfStops;

            public VehicleByRoute(CityVehicle vehicle, double distance,int numberOfStops) {

                this.vehicle = vehicle;

                this.distance = distance;

                this.numberOfStops = numberOfStops;

            }

            @Override

            public void run() {

                double interval = distance/numberOfStops;

                startTrip();

                for(double d = 0, i = 0; d < distance;d+=interval,i++){

                    try {

                        Thread.sleep((long)((interval/this.vehicle.speed)\*60\*60) );

                    } catch (InterruptedException e) {

                        e.printStackTrace();

                    }

                    Arrived(i+1);

                    int accident = (int)(Math.random()\*10);

                    if(accident < 1) {

                        System.out.print("Произошла авария!!!!");

                        synchronized (vehicle) {

                            this.vehicle.isAlive = false;

                        }

                        break;

                    }

                }

                    System.out.println(vehicle.show() + " закончил движение");

                    vehicle.isArrive = true;

            }

            private synchronized void startTrip(){

                vehicle.start();

            }

            private synchronized void Arrived(double num) {

                System.out.println(vehicle.show() + " прибыл на остановку №" + num);

            }

        }

        public class Checker implements Runnable {

            boolean interrupt=false;

            double intervalMultiply=1;

            List<VehicleByRoute> vehicles = new ArrayList<>();

            List<VehicleByRoute> vehiclesAddition = new ArrayList<>();

            @Override

            public void run() {

                List<VehicleByRoute> list = new ArrayList<>(vehicles);

                while(!interrupt) {

                    interrupt = true;

                    synchronized (vehicles) {

                        for (VehicleByRoute vr : list) {

                            if (!vr.vehicle.isAlive) {

                                if (!vehiclesAddition.isEmpty()) {

                                    list.add(vehiclesAddition.get(0));

                                    vehicles.add(0,vehiclesAddition.get(0));

                                    vehiclesAddition.remove(0);

                                    break;

                                } else

                                    intervalMultiply += 5;

                            }

                            if(!vr.vehicle.isArrive)

                                interrupt=false;

                        }

                    }

                }

            }

        }

    }

    public static class Bus extends CityVehicle {

        public Bus(int number) {

            super(number,50);

            this.name = "Автобус";

        }

        @Override

        public String show() {

            return this.name + " №" + this.number;

        }

    }

    public static class Trolleybus extends CityVehicle {

        public Trolleybus(int number) {

            super(number,40);

            this.name = "Троллейбус";

        }

        @Override

        public String show() {

            return this.name + " №" + this.number;

        }

    }

}

abstract class CityVehicle {

    boolean isAlive;

    boolean isArrive = false;

    double speed;

    int number;

    String name;

    boolean start(){

        if(isAlive) {

            System.out.println(this.name +" №"+this.number + " начал движение");

        }

        else{

            System.out.println(this.name +" №"+this.number + " поломан");

        }

        return isAlive;

    }

    public CityVehicle(int number, int speed) {

        isAlive = true;

        this.number = number;

        this.speed = speed;

    }

    public abstract String show();

}

**Результаты работы**

Автобус №38 начал движение

Автобус №1 начал движение

Автобус №1 прибыл на остановку №1.0

Автобус №38 прибыл на остановку №1.0

Автобус №21 начал движение

Автобус №38 прибыл на остановку №2.0

Автобус №1 прибыл на остановку №2.0

Автобус №21 прибыл на остановку №1.0

Произошла авария!!!!Автобус №21 закончил движение

Автобус №38 прибыл на остановку №3.0

Автобус №1 прибыл на остановку №3.0

Автобус №38 прибыл на остановку №4.0

Автобус №1 прибыл на остановку №4.0

Троллейбус №228 начал движение

Автобус №38 прибыл на остановку №5.0

Автобус №1 прибыл на остановку №5.0

Троллейбус №228 прибыл на остановку №1.0

Автобус №1 прибыл на остановку №6.0

Автобус №38 прибыл на остановку №6.0

Троллейбус №228 прибыл на остановку №2.0

Автобус №1 прибыл на остановку №7.0

Автобус №38 прибыл на остановку №7.0

Троллейбус №228 прибыл на остановку №3.0

Автобус №1 прибыл на остановку №8.0

Автобус №38 прибыл на остановку №8.0

Автобус №1 закончил движение

Автобус №38 закончил движение

Троллейбус №228 прибыл на остановку №4.0

Троллейбус №228 прибыл на остановку №5.0

Троллейбус №228 прибыл на остановку №6.0

Троллейбус №228 прибыл на остановку №7.0

Троллейбус №228 прибыл на остановку №8.0

Троллейбус №228 закончил движение

**Вывод:** в ходе работы приобрела практические навыки в области объектно- ориентированного проектирования.