

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ
«БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
КАФЕДРА ИИТ

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №3

Выполнил:

Студент группы АС-50

ФЭИС, Левкович Р.А.

Проверил:

Крощенко А.А.

Брест 2020

Цель работы: Научиться создавать и использовать классы в программах на языке программирования Java.

Вариант 3

Задание 1: Реализовать простой класс.

Требования к выполнению

- Реализовать пользовательский класс по варианту.
- Создать другой класс с методом main, в котором будут находиться примеры использования

пользовательского класса.

Для каждого класса

- Создать поля классов
- Создать методы классов
- Добавьте необходимые get и set методы (по необходимости)
- Укажите соответствующие модификаторы видимости
- Добавьте конструкторы
- Переопределить методы toString() и equals()

3) Прямоугольный треугольник, заданный длинами сторон – Предусмотреть возможность

определения площади и периметра, а так же логический метод, определяющий существует или

такой треугольник. Конструктор должен позволить создавать объекты с начальной инициализацией.

Реализовать метод equals, выполняющий сравнение объектов данного типа.

3) Автоматизированная система в автобусном парке

Составить программу, которая содержит информацию о наличие автобусов в автобусном парке.

Сведения о каждом автобусе содержат (Bus) содержат:

- Фамилия и инициалы водителя;
- Номер автобуса;
- Номер маршрута;
- Марка;
- Год начала эксплуатации;
- Пробег;
- Местонахождение в настоящий момент времени (парк/маршрут).

Программа должна обеспечивать:

- Формирование данных обо всех автобусах в виде списка;

- Формирование списка автобусов выехавших из парка;
- Формирование списка автобусов оставшихся в парке;
- Список автобусов для заданного номера маршрута;
- Список автобусов, которые эксплуатируются больше 10 лет;
- Список автобусов, пробег у которых больше 100000 км.
- Вывод сведений об автобусах, находящихся на маршруте и об автобусах, оставшихся в парке.

Код программы:

```
package com.company;

class Triangle
{
    double gipotenyza;
    double katet1, katet2;

    public Triangle(double gipotenyza, double katet1, double katet2) {
        this.gipotenyza = gipotenyza;
        this.katet1 = katet1;
        this.katet2 = katet2;
    }

    public Triangle() {
        gipotenyza = katet1 = katet2 = 0;
    }

    public boolean isExists() {
        if(gipotenyza <= 0 || katet1 <= 0 || katet2 <= 0)
            return false;

        if(Math.pow(katet1, 2) + Math.pow(katet2, 2) != Math.pow(gipotenyza,
2))
            return false;

        return true;
    }

    public double getPerimeter() {
        return gipotenyza + katet1 + katet2;
    }

    public double getSquare() {
        return katet1*katet2*0.5;
    }

    public String toString() {
        return "katet1="+katet1+", katet2="+katet2+",
gipotenyza="+gipotenyza;
    }

    public boolean equals(Triangle other) {
        if(other.gipotenyza == gipotenyza) {

            if(other.katet1 == katet1 && other.katet2 == katet2)
                return true;
            else if(other.katet1 == katet2 && other.katet2 == katet1)
                return true;
        }
    }
}
```

```

        return false;
    }

    public void setKatet1(double katet1){
        this.katet1 = katet1;
    }

    public double getKatet1(){
        return katet1;
    }

    public void setKatet2(double katet2){
        this.katet2 = katet2;
    }

    public double getKatet2(){
        return katet2;
    }

    public void setGipotenyza(double gipotenyza){
        this.gipotenyza = gipotenyza;
    }

    public double getGipotenyza(){
        return gipotenyza;
    }
}

public class Main {

    public static void main(String[] args) {
        Triangle t1 = new Triangle(5, 3, 4);
        System.out.println(t1.toString());

        System.out.println("Perimeter "+t1.getPerimeter());
        System.out.println("Square "+t1.getSquare());

        Triangle t2 = new Triangle();
        t2.setGipotenyza(5);
        t2.setKatet1(4);
        t2.setKatet2(3);

        System.out.println("t1 equals t2 - "+ t1.equals(t2));

        t2.setKatet2(4);

        System.out.println("exists triangle t2 - "+t2.isExists());

        System.out.println("t1 equals t2 - "+ t1.equals(t2));
    }
}

```

Полученный результат:

```
katet1=3.0, katet2=4.0, gipotenyza=5.0
Perimeter 12.0
Square 6.0
t1 equals t2 - true
exists triangle t2 - false
t1 equals t2 - false
```

Задание 2: Разработать автоматизированную систему на основе некоторой структуры данных, манипулирующей объектами пользовательского класса. Реализовать требуемые функции обработки данных

Требования к выполнению

- Задание посвящено написанию классов, решающих определенную задачу автоматизации;
- Данные для программы загружаются из файла (формат произволен). Файл создать и написать вручную.

Код программы:

Bus.class:

```
package com.company;

public class Bus {
    private String driverFIO;
    private String number;
    private int route;
    private String producer;
    private int year;
    private int mileage;
    private int locationCode; //0 - парк, 1 - маршрут

    public String getDriverFIO() {
        return driverFIO;
    }

    public void setDriverFIO(String driverFIO) {
        this.driverFIO = driverFIO;
    }

    public String getNumber() {
        return number;
    }

    public void setNumber(String number) {
        this.number = number;
    }

    public int getRoute() {
        return route;
    }
}
```

```

    }

    public void setRoute(int route) {
        this.route = route;
    }

    public String getProducer() {
        return producer;
    }

    public void setProducer(String producer) {
        this.producer = producer;
    }

    public int getYear() {
        return year;
    }

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

    public int getMileage() {
        return mileage;
    }

    public void setMileage(int mileage) {
        this.mileage = mileage;
    }

    public int getLocationCode() {
        return locationCode;
    }

    public void setLocationCode(int locationCode) {
        this.locationCode = locationCode;
    }

    public String toString(){
        String location = null;
        if(locationCode==0)
            location = "в парке";
        else if(locationCode==1)
            location = "на маршруте";

        return driverFIO+", "+number+", "+route+", "+producer+", "+year+",
"+mileage+", "+location;
    }
}

```

Buses.class:

```

package com.company;

import com.company.Bus;

import java.io.File;
import java.util.ArrayList;
import java.util.Scanner;

public class Buses {

```

```

ArrayList<Bus> buses = new ArrayList<Bus>();
int currentYear = 2020;

public Buses() throws Exception
{
    File file = new File("file.txt");
    Scanner scanner = new Scanner(file, "utf-8");

    while(scanner.hasNextLine())
    {
        String line = scanner.nextLine();
        String[] words = line.split(",");

        Bus bus = new Bus();
        bus.setDriverFIO(words[0]);
        bus.setNumber(words[1]);
        bus.setRoute(Integer.parseInt(words[2]));
        bus.setProducer(words[3]);
        bus.setYear(Integer.parseInt(words[4]));
        bus.setMileage(Integer.parseInt(words[5]));
        bus.setLocationCode(Integer.parseInt(words[6]));

        buses.add(bus);
    }
}

public void show()
{
    for(int i=0; i<buses.size(); i++)
        System.out.println(buses.get(i).toString());
}

public void showPark()
{
    for(int i=0; i<buses.size(); i++)
        if(buses.get(i).getLocationCode()==0)
            System.out.println(buses.get(i).toString());
}

public void showRoute()
{
    for(int i=0; i<buses.size(); i++)
        if(buses.get(i).getLocationCode()==1)
            System.out.println(buses.get(i).toString());
}

public void showRoute(int route)
{
    for(int i=0; i<buses.size(); i++)
        if(buses.get(i).getRoute() == route)
            System.out.println(buses.get(i).toString());
}

public void showLonger()
{
    for(int i=0; i<buses.size(); i++)
        if(currentYear - buses.get(i).getYear() > 10)
            System.out.println(buses.get(i).toString());
}

public void showLongMileAge()
{
    for(int i=0; i<buses.size(); i++)
        if(buses.get(i).getMileage() > 100000)

```

```

        System.out.println(buses.get(i).toString());
    }

    public void showInfo()
    {
        System.out.println("В парке:\n");

        for(int i=0;i<buses.size(); i++)
            if(buses.get(i).getLocationCode()==0)
                System.out.println(buses.get(i).toString());

        System.out.println("На маршруте:\n");

        for(int i=0;i<buses.size(); i++)
            if(buses.get(i).getLocationCode()==1)
                System.out.println(buses.get(i).toString());
    }
}

```

Main.class:

```

package com.company;
import com.company.Menu;

public class Main {

    public static void main(String[] args) throws Exception {
        Menu menu = new Menu();
        menu.start();
    }
}

```

Menu.class:

```

package com.company;
import com.company.Buses;

import java.util.Scanner;

public class Menu {
    Buses buses;

    void start() throws Exception
    {
        buses = new Buses();

        Scanner scanner = new Scanner(System.in);

        for(boolean key=true; key;)
        {
            System.out.println("Введите 0, чтобы посмотреть все автобусы");
            System.out.println("Введите 1, чтобы посмотреть автобусы,
которые выехали из парка");
            System.out.println("Введите 2, чтобы посмотреть автобусы,
которые остались в парке");
            System.out.println("Введите 3, чтобы посмотреть автобусы по
заданному номеру маршрута");
            System.out.println("Введите 4, чтобы посмотреть автобусы,
которые эксплуатируются больше 10 лет");
            System.out.println("Введите 5, чтобы посмотреть автобусы, пробег
которых больше 100000 км");

```



```

        System.out.println("Введите 6, чтобы посмотреть автобусы,
которые находятся в парке, и автобусы, которые находятся на маршруте");
        System.out.println("Введите 7, чтобы выйти");

        int k = Integer.parseInt(scanner.nextLine());

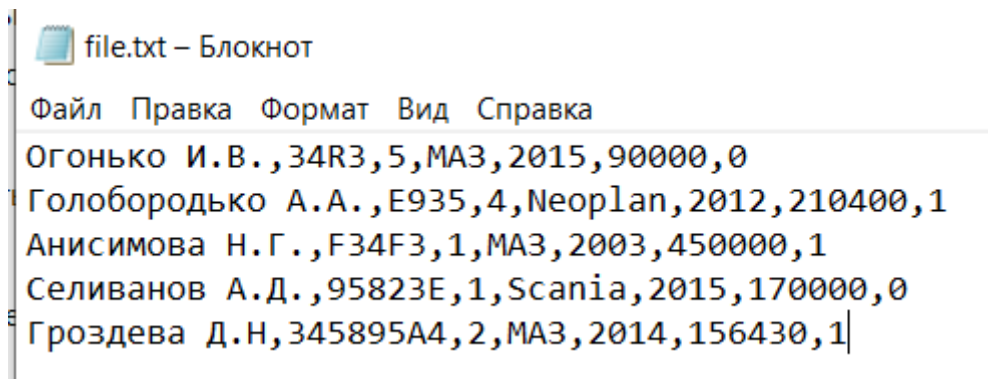
        switch(k)
        {
            case 0: buses.show(); break;
            case 1: buses.showRoute(); break;
            case 2: buses.showPark(); break;
            case 3: System.out.println("Введите номер маршрута:");
                    int route = Integer.parseInt(scanner.nextLine());
                    buses.showRoute(route); break;
            case 4: buses.showLonger(); break;
            case 5: buses.showLongMileAge(); break;
            case 6: buses.showInfo(); break;
            case 7: key = false;
        }

        scanner.nextLine();
        System.out.println("\n\n\n\n\n\n\n");

    }
}

```

File.txt:

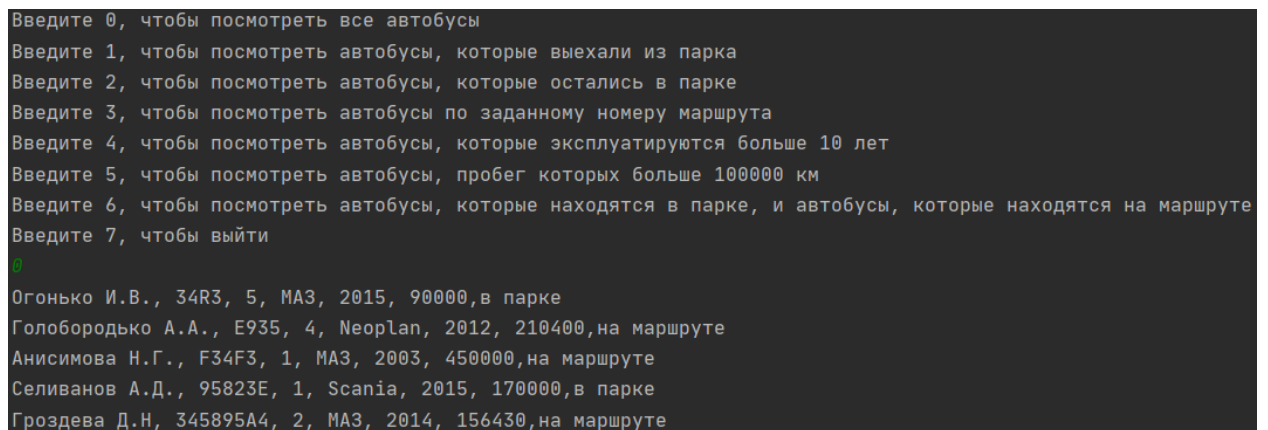


file.txt – Блокнот

Файл Правка Формат Вид Справка

Огонько И.В., 34R3, 5, MA3, 2015, 90000, 0
Голобородько А.А., E935, 4, Neoplan, 2012, 210400, 1
Анисимова Н.Г., F34F3, 1, MA3, 2003, 450000, 1
Селиванов А.Д., 95823E, 1, Scania, 2015, 170000, 0
Гроздева Д.Н, 345895A4, 2, MA3, 2014, 156430, 1

Полученный результат:



Введите 0, чтобы посмотреть все автобусы
Введите 1, чтобы посмотреть автобусы, которые выехали из парка
Введите 2, чтобы посмотреть автобусы, которые остались в парке
Введите 3, чтобы посмотреть автобусы по заданному номеру маршрута
Введите 4, чтобы посмотреть автобусы, которые эксплуатируются больше 10 лет
Введите 5, чтобы посмотреть автобусы, пробег которых больше 100000 км
Введите 6, чтобы посмотреть автобусы, которые находятся в парке, и автобусы, которые находятся на маршруте
Введите 7, чтобы выйти

Огонько И.В., 34R3, 5, MA3, 2015, 90000, в парке
Голобородько А.А., E935, 4, Neoplan, 2012, 210400, на маршруте
Анисимова Н.Г., F34F3, 1, MA3, 2003, 450000, на маршруте
Селиванов А.Д., 95823E, 1, Scania, 2015, 170000, в парке
Гроздева Д.Н, 345895A4, 2, MA3, 2014, 156430, на маршруте

```
Введите 0, чтобы посмотреть все автобусы
Введите 1, чтобы посмотреть автобусы, которые выехали из парка
Введите 2, чтобы посмотреть автобусы, которые остались в парке
Введите 3, чтобы посмотреть автобусы по заданному номеру маршрута
Введите 4, чтобы посмотреть автобусы, которые эксплуатируются больше 10 лет
Введите 5, чтобы посмотреть автобусы, пробег которых больше 100000 км
Введите 6, чтобы посмотреть автобусы, которые находятся в парке, и автобусы, которые находятся на маршруте
Введите 7, чтобы выйти
0
Голобородько А.А., Е935, 4, Neoplan, 2012, 210400,на маршруте
Анисимова Н.Г., F34F3, 1, MA3, 2003, 450000,на маршруте
Гроздева Д.Н, 345895A4, 2, MA3, 2014, 156430,на маршруте
```

```
Введите 0, чтобы посмотреть все автобусы
Введите 1, чтобы посмотреть автобусы, которые выехали из парка
Введите 2, чтобы посмотреть автобусы, которые остались в парке
Введите 3, чтобы посмотреть автобусы по заданному номеру маршрута
Введите 4, чтобы посмотреть автобусы, которые эксплуатируются больше 10 лет
Введите 5, чтобы посмотреть автобусы, пробег которых больше 100000 км
Введите 6, чтобы посмотреть автобусы, которые находятся в парке, и автобусы, которые находятся на маршруте
Введите 7, чтобы выйти
0
В парке:
Огонько И.В., 34R3, 5, MA3, 2015, 90000,в парке
Селиванов А.Д., 95823E, 1, Scania, 2015, 170000,в парке
На маршруте:
Голобородько А.А., Е935, 4, Neoplan, 2012, 210400,на маршруте
Анисимова Н.Г., F34F3, 1, MA3, 2003, 450000,на маршруте
Гроздева Д.Н, 345895A4, 2, MA3, 2014, 156430,на маршруте
```

Вывод: Научился создавать и использовать классы в программах на языке программирования Java.