

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ
«БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
ФАКУЛЬТЕТ ЭЛЕКТРОННО-ИНФОРМАЦИОННЫХ СИСТЕМ
Кафедра интеллектуальных информационных технологий

Отчет по лабораторной работе №3

Выполнил

Булей Е.В.

студент группы АС50

Проверил

А. А. Крощенко,

ст. преп. кафедры ИИТ,

Цель работы:

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

ВАРИАНТ 2

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()` 2) Напишите метод `reverse(double[] array)`, который меняет порядок элементов в массиве на обратный.

2) Равносторонний треугольник, заданный длинами сторон – Предусмотреть возможность определения площади и периметра, а так же логический метод, определяющий существует или такой треугольник. Конструктор должен позволять создавать объекты с начальной инициализацией. Реализовать метод `equals`, выполняющий сравнение объектов данного типа.

Код программы:

```
import static java.lang.Math.*;

public class EquilTriangle {
    private double a;

    public EquilTriangle() {
        a = 1;
    }

    public EquilTriangle(double a) {
```

```

        this.a = a;
    }

    public boolean isTriangle(){
        if(a<=0) return false;
        return true;
    }

    public double perimeter(){
        return 3*a;
    }

    public double square(){
        return a*a*sqrt(3)/4;
    }

    public double getA() {
        return a;
    }

    public void setA(double a) {
        this.a = a;
    }

    @Override
    public String toString() {
        return "EquilTriangle{" +
            "a=" + a +
            '}';
    }

    @Override
    public boolean equals(Object o) {
        if (this == o) return true;
        if (!(o instanceof EquilTriangle)) return false;
        EquilTriangle that = (EquilTriangle) o;
        return Double.compare(that.a, a) == 0;
    }
}

```

Входные параметры:

```

public class Main {
    public static void main(String[] args){
        EquilTriangle first = new EquilTriangle(3.0);
        EquilTriangle second = new EquilTriangle(3);
        EquilTriangle trird = new EquilTriangle(-1);

        System.out.println("first "+ first.isTriangle());
        System.out.println("trird "+ trird.isTriangle());

        System.out.println("Perimeter first "+ first.perimeter());
        System.out.println("Square first "+ first.square());

        System.out.println("first equals second - "+first.equals(second));
        System.out.println("first equals trird - "+ first.equals(trird));

    }
}

```

Вывод:

```
first true
trird false
Perimeter first 9.0
Square first 3.8971143170299736
first equals second - true
first equals trird - false

Process finished with exit code 0
```

Задание 2

Разработать автоматизированную систему на основе некоторой структуры дан- ных, 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 написатьвручную.

2) Автоматизированная система проката автомобилей

Составить программу, которая хранит и обрабатывает информацию о прокате автомобилей. Окаждом автомобиле (Car) содержится следующая информация:

- id;
- Марка;
- Модель;
- Год выпуска;
- Цвет;
- Цена;
- Регистрационный номер;
- Номер машины.
- ФИО лица, взявшего на прокат (при наличии);

- Номер паспорта лица-арендатора (при наличии).

Программа должна обеспечить вывод списков:

- автомобилей;
- автомобилей заданной марки;
- автомобилей заданной модели, которые эксплуатируются больше n лет;
- автомобилей заданного года выпуска, цена которых больше указанной;
- автомобилей, взятых на прокат;
- автомобилей, взятых на прокат с выводом личной информации об арендаторах.

Код программы:

```
public class Car {
    private String ID;
    private String mark;
    private String model;
    private String yearIssue;
    private String color;
    private String price;
    private String registrationNumber;
    private String numberCar;
    private String personRented;
    private String pasportID;

    public Car(String ID, String mark, String model, String yearIssue, String
color,
                String price, String registrationNumber,String numberCar,
String personRented, String pasportID) {
        this.ID = ID;
        this.mark = mark;
        this.model = model;
        this.yearIssue = yearIssue;
        this.color = color;
        this.price = price;
        this.registrationNumber = registrationNumber;
        this.numberCar = numberCar;
        this.personRented = personRented;
        this.pasportID = pasportID;
    }

    public String getCar(){
        return "Car{" +
            "ID=" + ID +
            ", mark='" + mark + '\'' +
            ", model='" + model + '\'' +
            ", yearIssue=" + yearIssue +
            ", color='" + color + '\'' +
            ", price='" + price + '\'' +
            ", registrationNumber='" + registrationNumber + '\'' +
            ", numberCar=" + numberCar +
            '}';
    }
}
```

```

@Override
public String toString() {
    return "Car{" +
        "ID='" + ID + '\'' +
        ", mark='" + mark + '\'' +
        ", model='" + model + '\'' +
        ", yearIssue='" + yearIssue + '\'' +
        ", color='" + color + '\'' +
        ", price='" + price + '\'' +
        ", registrationNumber='" + registrationNumber + '\'' +
        ", numberCar='" + numberCar + '\'' +
        ", personRented='" + personRented + '\'' +
        ", passportID='" + passportID + '\'' +
        '}';
}

public String getID() {
    return ID;
}

public void setID(String ID) {
    this.ID = ID;
}

public String getMark() {
    return mark;
}

public void setMark(String mark) {
    this.mark = mark;
}

public String getModel() {
    return model;
}

public void setModel(String model) {
    this.model = model;
}

public String getYearIssue() {
    return yearIssue;
}

public void setYearIssue(String yearIssue) {
    this.yearIssue = yearIssue;
}

public String getColor() {
    return color;
}

public void setColor(String color) {
    this.color = color;
}

public String getPrice() {
    return price;
}

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

```

```

    public String getRegistrationNumber() {
        return registrationNumber;
    }

    public void setRegistrationNumber(String registrationNumber) {
        this.registrationNumber = registrationNumber;
    }

    public String getNumberCar() {
        return numberCar;
    }

    public void setNumberCar(String numberCar) {
        this.numberCar = numberCar;
    }

    public String getPersonRented() {
        return personRented;
    }

    public void setPersonRented(String personRented) {
        this.personRented = personRented;
    }

    public String getPasportID() {
        return pasportID;
    }

    public void setPasportID(String pasportID) {
        this.pasportID = pasportID;
    }
}

```

```

import java.io.*;
import java.util.ArrayList;

public class Main {
    private static ArrayList<Car> list;
    public static void main(String[] args) {
        list = new ArrayList<>();
        fillList(list);
        System.out.println("All cars");
        outputData(list);
        System.out.println("Cars mark bmw");
        carMark(list, "bmw");
        System.out.println("Cars model q7, yers explotion more 1 year");
        carModelAndYear(list, "q7", 1);
        System.out.println("Cars year issue 2018 and prise more 250000");
        carYearAndPrice(list, 2018, 250000);
        System.out.println("Cars what was rented");
        retnedCar(list);
        System.out.println("Cars what was rented with information about
man");
        retnedCarAndInformation(list);
    }

    public static void fillllList(ArrayList<Car> list) {
        try {
            File file = new File("car.txt");
            FileReader fr = new FileReader(file);
            BufferedReader reader = new BufferedReader(fr);
            String line = reader.readLine();

```

```

        Car car;
        while(line!=null){
            String[] words = line.split("\t");
            car = new Car(words[0],words[1],words[2],words[3],words[4],
                words[5],words[6],words[7],words[8],words[9]);
            list.add(car);
            line = reader.readLine();
        }
    } catch (FileNotFoundException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    }
}

public static void outputData(ArrayList<Car> list){
    for(Car car : list){
        System.out.println(car.getCar());
    }
}

public static void carMark(ArrayList<Car> list, String mark){
    for(Car car : list){
        if(car.getMark().equals(mark)){
            System.out.println(car.getCar());
        }
    }
}

public static void carModelAndYear(ArrayList<Car> list, String model, int
year){
    for(Car car : list){
        int yearIssue = 2020 - Integer.parseInt(car.getYearIssue());
        if(car.getModel().equals(model) && yearIssue>year){
            System.out.println(car.getCar());
        }
    }
}

public static void carYearAndPrice(ArrayList<Car> list, int year, int
price){
    String yearIssue = Integer.toString(year);
    for(Car car : list){
        if(car.getYearIssue().equals(yearIssue) &&
Integer.parseInt(car.getPrice())>price){
            System.out.println(car.getCar());
        }
    }
}

public static void retnedCar(ArrayList<Car> list){
    for(Car car : list){
        if(!car.getPersonRented().equals("-")){
            System.out.println(car.getCar());
        }
    }
}

public static void retnedCarAndInformation(ArrayList<Car> list){
    for(Car car : list){
        if(!car.getPersonRented().equals("-")){
            System.out.println(car.toString());
        }
    }
}

```



```
}  
}
```

Входные параметры:

Вывод:

```
All cars  
Car{ID=1, mark='audi', model='q7', yearIssue=2018, color='blue', price='1000000', registrationNumber='0000 AB-0', numberCar=0000}  
Car{ID=2, mark='audi', model='A8', yearIssue=2018, color='red', price='500000', registrationNumber='1111 AB-1', numberCar=1111}  
Car{ID=3, mark='audi', model='q7', yearIssue=2020, color='green', price='1000000', registrationNumber='2222 AB-2', numberCar=2222}  
Car{ID=4, mark='bmw', model='M3', yearIssue=2005, color='blue', price='234345', registrationNumber='1212 AB-1', numberCar=1212}  
Car{ID=5, mark='audi', model='A1', yearIssue=2011, color='red', price='463466', registrationNumber='3333 AB-3', numberCar=3333}  
Car{ID=6, mark='audi', model='A7', yearIssue=2010, color='blue', price='43665767', registrationNumber='1115 AB-1', numberCar=1115}  
Car{ID=7, mark='audi', model='q2', yearIssue=2018, color='red', price='24355', registrationNumber='1171 AB-6', numberCar=1171}  
Car{ID=8, mark='audi', model='q5', yearIssue=2008, color='blue', price='85654', registrationNumber='1611 AB-2', numberCar=1611}  
Car{ID=9, mark='bmw', model='m3', yearIssue=2013, color='green', price='14688', registrationNumber='5111 AB-1', numberCar=5111}  
Car{ID=10, mark='audi', model='RS4', yearIssue=2018, color='red', price='13546', registrationNumber='1119 AB-1', numberCar=1119}  
Cars mark bmw  
Car{ID=4, mark='bmw', model='M3', yearIssue=2005, color='blue', price='234345', registrationNumber='1212 AB-1', numberCar=1212}  
Car{ID=9, mark='bmw', model='m3', yearIssue=2013, color='green', price='14688', registrationNumber='5111 AB-1', numberCar=5111}  
Cars model q7, yers explotion more 1 year  
Car{ID=1, mark='audi', model='q7', yearIssue=2018, color='blue', price='1000000', registrationNumber='0000 AB-0', numberCar=0000}  
Cars year issue 2018 and prise more 250000  
Car{ID=1, mark='audi', model='q7', yearIssue=2018, color='blue', price='1000000', registrationNumber='0000 AB-0', numberCar=0000}  
Car{ID=2, mark='audi', model='A8', yearIssue=2018, color='red', price='500000', registrationNumber='1111 AB-1', numberCar=1111}  
Cars what was rented  
Car{ID=2, mark='audi', model='A8', yearIssue=2018, color='red', price='500000', registrationNumber='1111 AB-1', numberCar=1111}  
Car{ID=4, mark='bmw', model='M3', yearIssue=2005, color='blue', price='234345', registrationNumber='1212 AB-1', numberCar=1212}  
Car{ID=6, mark='audi', model='A7', yearIssue=2010, color='blue', price='43665767', registrationNumber='1115 AB-1', numberCar=1115}  
Car{ID=8, mark='audi', model='q5', yearIssue=2008, color='blue', price='85654', registrationNumber='1611 AB-2', numberCar=1611}  
Car{ID=10, mark='audi', model='RS4', yearIssue=2018, color='red', price='13546', registrationNumber='1119 AB-1', numberCar=1119}
```

```
Cars what was rented with information about man  
Car{ID=2, mark='audi', model='A8', yearIssue='2018', color='red', price='500000', registrationNumber='1111 AB-1', numberCar='1111', personRented='Ivanon'},  
Car{ID=4, mark='bmw', model='M3', yearIssue='2005', color='blue', price='234345', registrationNumber='1212 AB-1', numberCar='1212', personRented='Petrov'},  
Car{ID=6, mark='audi', model='A7', yearIssue='2010', color='blue', price='43665767', registrationNumber='1115 AB-1', numberCar='1115', personRented='Sidor'},  
Car{ID=8, mark='audi', model='q5', yearIssue='2008', color='blue', price='85654', registrationNumber='1611 AB-2', numberCar='1611', personRented='Ivashenk'},  
Car{ID=10, mark='audi', model='RS4', yearIssue='2018', color='red', price='13546', registrationNumber='1119 AB-1', numberCar='1119', personRented='Kulinko'}  
  
Process finished with exit code 0
```

Вывод: Приобрел практические навыки создания и использования классов в программах на языке программирования Java.