

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

Лабораторная работа №3

По дисциплине: «Современные платформы программирования»

Выполнил:
студент 3 курса
группы ПО-8
Бубен С.О.
Проверил:
Крощенко А.А

Брест, 2024

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

Вариант 2

Задание 1 :Равносторонний треугольник, заданный длинами сторон – Предусмотреть возможность определения площади и периметра, а так же логический метод, определяющий существует ли такой треугольник. Конструктор должен позволить создавать объекты с начальной инициализацией. Реализовать метод equals, выполняющий сравнение объектов данного типа.

Работа программы:

```
Первый треугольник: { sideA: 2.0, sideB: 2.0, sideC: 2.0 }  
Периметр треугольника : 6.0  
Площадь треугольника : 1.7320508075688772  
Существует ли треугольник ? true  
Является ли треугольник равносторонним? true  
  
Второй треугольник : { sideA: 4.0, sideB: 4.0, sideC: 4.0 }  
Периметр треугольника : 12.0  
Площадь треугольника : 6.928203230275509  
Существует ли треугольник ? true  
Является ли треугольник равносторонним? true  
  
Равны ли треугольники ? false
```

Код:

```
public class rovnotriangle {  
    private double sideA, sideB, sideC;  
  
    public rovnotriangle(double sideA) {  
        this.sideA = sideA;  
        this.sideB = sideA;  
        this.sideC = sideA;  
    }  
  
    public double SideA() {  
        return sideA;  
    }  
  
    public double SideB() {  
        return sideB;  
    }  
  
    public double SideC() {  
        return sideC;  
    }  
  
    public void setSideA(double sideA) {  
        this.sideA = sideA;  
    }  
}
```

```

    }

    public void setSideB(double sideB) {
        this.sideB = sideA;
    }

    public void setSideC(double sideC) {
        this.sideC = sideA;
    }

    public double peremiter() {
        return sideA + sideB + sideC;
    }

    public double square() {
        return (Math.sqrt(3) / 4) * sideA * sideA;
    }

    public boolean existornot() {
        return sideA + sideB > sideC && sideA + sideC > sideB && sideB + sideC > sideA;
    }

    public boolean sidesequale() {
        return (sideA == sideB && sideB == sideC && sideA == sideC);
    }

    @Override
    public String toString() {
        return "{ sideA: " + sideA + ", sideB: " + sideB + ", sideC: " + sideC + " }";
    }

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

        if (obj == null || this.getClass() != obj.getClass()) {
            return false;
        }

        rovntriangle other = (rovntriangle) obj;
        return sideA == other.sideA && sideB == other.sideB && sideC == other.sideC;
    }

    public static void main(String[] args) {
        rovntriangle triangle1 = new rovntriangle(2);
        rovntriangle triangle2 = new rovntriangle(4);

        System.out.println("Первый треугольник: " + triangle1);
        System.out.println("Периметр треугольника : " + triangle1.peremiter());
        System.out.println("Площадь треугольника : " + triangle1.square());
        System.out.println("Существует ли треугольник ? " + triangle1.existornot());
        System.out.println("Является ли треугольник равностаронним? " + triangle1.sidesequale());

        System.out.println("\nВторой треугольник : " + triangle2);
        System.out.println("Периметр треугольника : " + triangle2.peremiter());
        System.out.println("Площадь треугольника : " + triangle2.square());
        System.out.println("Существует ли треугольник ? " + triangle2.existornot());
        System.out.println("Является ли треугольник равностаронним? " + triangle2.sidesequale());

        System.out.println("\nРавны ли треугольники ? " + triangle1.equals(triangle2));
    }
}

```

Задание 2: Автоматизированная система проката автомобилей. Составить программу, которая хранит и обрабатывает информацию о прокате автомобилей. О каждом 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.

Работа программы:

```
(1) Вывод списка всех автомобилей.
(2) Вывод списка всех автомобилей заданной марки.
(3) Вывод списка автомобилей заданной модели, которые эксплуатируются больше n лет.
(4) Вывод списка всех автомобилей заданного года выпуска, цена которых больше указанной.
(5) Вывод списка всех автомобилей, взятых на прокат.
(6) Вывод списка всех автомобилей, взятых на прокат с выводом личной информации об арендаторах.
(7) Выход.
Выберите пункт меню:
```

Код:

```
import java.io.BufferedReader;
import java.io.File;
import java.io.FileReader;
import java.time.Year;
import java.util.ArrayList;
import java.util.Scanner;

class Car {
    private int carId;
    private String carBrand;
    private String carModel;
    private int year;
    private String carColor;
    private double carPrice;
```

```
private String regNumber;
private String carNumber;
private String renterFIO;
private String renterPassportNumber;

public Car(int carId, String carBrand, String carModel, int year, String carColor, double carPrice,
           String regNumber, String carNumber, String renterFIO, String renterPassportNumber) {
    this.carId = carId;
    this.carBrand = carBrand;
    this.carModel = carModel;
    this.year = year;
    this.carColor = carColor;
    this.carPrice = carPrice;
    this.regNumber = regNumber;
    this.carNumber = carNumber;
    this.renterFIO = renterFIO;
    this.renterPassportNumber = renterPassportNumber;
}

public Car() {
    this.carId = 0;
    this.carBrand = "Undefined";
    this.carModel = "Undefined";
    this.year = 0;
    this.carColor = "Undefined";
    this.carPrice = 0.0;
    this.regNumber = "Undefined";
    this.carNumber = "Undefined";
    this.renterFIO = "Undefined";
    this.renterPassportNumber = "Undefined";
}

public void setCarId(int carId) {
    this.carId = carId;
}

public void setCarBrand(String carBrand) {
    this.carBrand = carBrand;
}

public void setCarModel(String carModel) {
    this.carModel = carModel;
}

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

public void setCarColor(String carColor) {
    this.carColor = carColor;
}

public void setCarPrice(double carPrice) {
    this.carPrice = carPrice;
}

public void setRegNum(String regNumber) {
    this.regNumber = regNumber;
}

public void setCarNumber(String carNumber) {
    this.carNumber = carNumber;
}

public void setRenterFIO(String renterFIO) {
```

```

        this.renterFIO = renterFIO;
    }

    public void setRenterPassportNumber(String renterPassportNumber) {
        this.renterPassportNumber = renterPassportNumber;
    }

    public int getCarId() {
        return carId;
    }

    public String getCarBrand() {
        return carBrand;
    }

    public String getCarModel() {
        return carModel;
    }

    public int getYear() {
        return year;
    }

    public String getCarColor() {
        return carColor;
    }

    public double getCarPrice() {
        return carPrice;
    }

    public String getRegNumber() {
        return regNumber;
    }

    public String getCarNumber() {
        return carNumber;
    }

    public String getRenterFullName() {
        return renterFIO;
    }

    public String getRenterPassportNumber() {
        return renterPassportNumber;
    }

    public void printCarInfo() {
        System.out.println("id: " + this.carId);
        System.out.println("brand: " + this.carBrand);
        System.out.println("model: " + this.carModel);
        System.out.println("year: " + this.year);
        System.out.println("color: " + this.carColor);
        System.out.println("price: " + this.carPrice + "$");
        System.out.println("registration number: " + this.regNumber);
        System.out.println("number: " + this.carNumber);
        System.out.println("Renter FullName: " + this.renterFIO);
        System.out.println("Renter passport number: " +
            this.renterPassportNumber + "\n");
    }
}

class CarRent {
    private ArrayList<Car> cars;

```

```

public CarRent(String filePath) {
    this.cars = new ArrayList<>();
    File file = new File(filePath);
    try {
        BufferedReader reader = new BufferedReader(new FileReader(file));
        String line;
        Car car = null;
        while ((line = reader.readLine()) != null) {
            String[] arg = line.split(":");
            if (arg.length == 2) {
                String key = arg[0];
                String value = arg[1];
                switch (key) {
                    case "id":
                        car = new Car();
                        car.setCarId(Integer.parseInt(value));
                        break;
                    case "Brand":
                        car.setCarBrand(value);
                        break;
                    case "Model":
                        car.setCarModel(value);
                        break;
                    case "Year":
                        car.setYear(Integer.parseInt(value));
                        break;
                    case "Color":
                        car.setCarColor(value);
                        break;
                    case "Price":
                        car.setCarPrice(Double.parseDouble(value));
                        break;
                    case "RegNumber":
                        car.setRegNum(value);
                        break;
                    case "CarNumber":
                        car.setCarNumber(value);
                        break;
                    case "RenterFIO":
                        car.setRenterFIO(value);
                        break;
                    case "RenterPassportNumber":
                        car.setRenterPassportNumber(value);
                        cars.add(car);
                        break;
                    default:
                        System.out.println("Invalid data format: " +
                                           key);
                        break;
                }
            }
        }
    } catch (Exception ex) {
        System.out.println(ex.getMessage());
    }
}

// Вывод списка всех автомобилей
public void printAllCars() {
    System.out.println("Cars information: ");
    for (Car car : this.cars) {
        car.printCarInfo();
    }
}

```

```

// Вывод списка всех автомобилей заданной марки
public void printCarsByBrand(String carBrand) {
    boolean isFind = false;
    for (Car car : this.cars) {
        if (car.getCarBrand().equalsIgnoreCase(carBrand)) {
            isFind = true;
            car.printCarInfo();
        }
    }
    if (!isFind) {
        System.out.println("Ничего не найдено.");
    }
}

public void printOlderCarsByModel(String model, int years) {
    boolean isFind = false;
    int currentYear = Year.now().getValue();
    for (Car car : this.cars) {
        if (car.getCarModel().equalsIgnoreCase(model) && currentYear -
            car.getYear() > years) {
            isFind = true;
            car.printCarInfo();
        }
    }
    if (!isFind) {
        System.out.println("Ничего не найдено.");
    }
}

// Вывод списка всех автомобилей заданного года выпуска, цена которых больше
// указанной;

public void printCarsByYearAndPrice(int year, double price) {
    int currentYear = Year.now().getValue();
    if (year > currentYear) {
        System.out.println("Введите верные данные.");
        return;
    }
    for (Car car : this.cars) {
        if (car.getYear() == year && car.getCarPrice() > price) {
            car.printCarInfo();
        }
    }
}

// Вывод списка всех автомобилей, взятых на прокат
public void printRentedCars() {
    boolean isFind = false;
    System.out.println("Автомобили взятые на прокат: ");
    for (Car car : this.cars) {
        if (!car.getRenterFullName().equals("-") &&
            !car.getRenterPassportNumber().equals("-")) {
            System.out.println("id: " + car.getCarId() + "\nbrand: " +
                car.getCarBrand() + "\nmodel: " + car.getCarModel());
            isFind = true;
        }
    }
    if (!isFind) {
        System.out.println("Ничего не найдено.");
    }
}

public void printRentedCarsWithRenterInfo() {
    boolean isFind = false;
    System.out.println("Автомобили взятые на прокат: ");
}

```



```

        for (Car car : this.cars) {
            if (!car.getRenterFullName().equals("-") &&
                !car.getRenterPassportNumber().equals("-")) {
                System.out.println("id: " + car.getCarId() + "\nbrand: " +
                    car.getCarBrand() + "\nmodel: " + car.getCarModel()
                    + "\nRenter FullName: " + car.getRenterFullName() + "\nRenter passport number: "
                    + car.getRenterPassportNumber() + "\n");
                isFind = true;
            }
        }
        if (!isFind) {
            System.out.println("Ничего не найдено.");
        }
    }
}

public class task2 {
    public static void main(String[] args) {
        final String FILE_PATH = "C:/Users/stas/Рабочий стол/Carinfo.txt";
        CarRent carRentalSystem = new CarRent(FILE_PATH);
        while (true) {
            int choice;
            System.out.println("(1) Вывод списка всех автомобилей.");
            System.out.println("(2) Вывод списка всех автомобилей заданной марки.");
            System.out.println("(3) Вывод списка автомобилей заданной модели, которые эксплуатируются
больше n лет.");
            System.out.println("(4) Вывод списка всех автомобилей заданного года выпуска, цена которых
больше указанной.");
            System.out.println("(5) Вывод списка всех автомобилей, взятых на прокат.");
            System.out.println("(6) Вывод списка всех автомобилей, взятых на прокат с выводом личной
информации об арендаторах.");
            System.out.println("(7) Выход.");
            System.out.println("Выберите пункт меню: ");
            Scanner scanner = new Scanner(System.in);
            choice = scanner.nextInt();
            switch (choice) {
                case 1:
                    carRentalSystem.printAllCars();
                    break;
                case 2:
                    System.out.println("Введите марку автомобиля: ");
                    scanner.nextLine();
                    String brand = scanner.nextLine();
                    carRentalSystem.printCarsByBrand(brand);
                    break;
                case 3:
                    scanner.nextLine();
                    System.out.println("Введите модель автомобиля: ");
                    String model = scanner.nextLine();
                    System.out.println("Введите количество лет эксплуатации: ");
                    int years = scanner.nextInt();
                    carRentalSystem.printOlderCarsByModel(model, years);
                    break;
                case 4:
                    System.out.println("Введите год автомобиля: ");
                    int year = scanner.nextInt();
                    System.out.println("Введите цену: ");
                    double price = scanner.nextDouble();
                    carRentalSystem.printCarsByYearAndPrice(year, price);
                    break;
                case 5:
                    carRentalSystem.printRentedCars();
                    break;
                case 6:
                    carRentalSystem.printRentedCarsWithRenterInfo();

```

```
        break;
    case 7:
        return;
    default:
        System.out.println("Неверный пункт меню");
        break;
    }
}
}
```

Вывод: в ходе выполнения лабораторной работы я научился создавать и использовать классы в программах на языке программирования Java