МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

“БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ”

КАФЕДРА ИИТ

ОТЧЁТ

по лабораторной работе №3

Выполнил:

студент 3курса

группы ПО-8

Кодиров Б.Т.

Проверил:

Крощенко А.А.

Брест 2024

**Цель работы:** научится создавать и использовать классы в программах на языке Java

**Задание 1:** Реализовать простой класс.

Требования к выполнению

• Реализовать пользовательский класс по варианту.

• Создать другой класс с методом main, в котором будут находится 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()

Равнобедренный треугольник, заданный длинами сторон – Предусмотреть возмож-

ность 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.

**Код программы:**

**Triangle.java**

package lab3.exercise1;

public class Triangle {

private double sideA;

private double sideB;

private double base;

public Triangle(double sideA, double sideB, double base) {

this.sideA = sideA;

this.sideB = sideB;

this.base = base;

}

public double getSideA() {

return sideA;

}

public void setSideA(double sideA) {

this.sideA = sideA;

}

public double getSideB() {

return sideB;

}

public void setSideB(double sideB) {

this.sideB = sideB;

}

public double getBase() {

return base;

}

public void setBase(double base) {

this.base = base;

}

public double calculatePerimeter() {

return sideA + sideB + base;

}

public double calculateArea() {

double height = Math.sqrt(sideA \* sideA - (base \* base / 4));

return 0.5 \* base \* height;

}

public boolean exists() {

return (sideA + sideB > base && sideA + base > sideB && sideB + base > sideA);

}

@Override

public String toString() {

return "Triangle{" +

"sideA=" + sideA +

", sideB=" + sideB +

", base=" + base +

'}';

}

@Override

public boolean equals(Object o) {

if (this == o) return true;

if (o == null || getClass() != o.getClass()) return false;

Triangle that = (Triangle) o;

return Double.compare(that.sideA, sideA) == 0 &&

Double.compare(that.sideB, sideB) == 0 &&

Double.compare(that.base, base) == 0;

}

}

**Main.java**

package lab3.exercise1;

public class Main {

public static void main(String[] args) {

Triangle triangle1 = new Triangle(5, 5, 6);

Triangle triangle2 = new Triangle(3, 4, 5);

System.out.println("Triangle 1:");

System.out.println("Perimeter: " + triangle1.calculatePerimeter());

System.out.println("Area: " + triangle1.calculateArea());

System.out.println("Exists: " + triangle1.exists());

System.out.println("\nTriangle 2:");

System.out.println("Perimeter: " + triangle2.calculatePerimeter());

System.out.println("Area: " + triangle2.calculateArea());

System.out.println("Exists: " + triangle2.exists());

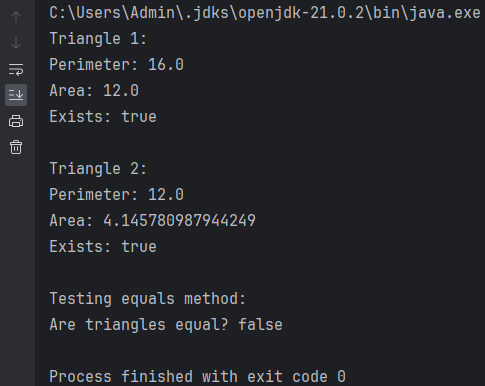
System.out.println("\nTesting equals method:");

System.out.println("Are triangles equal? " + triangle1.equals(triangle2));

}

}

**Результат программы:**



**Задание 2:** Автоматизированная система проката автомобилей

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

Брест 2024

* 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:**

**Main.java**

package lab3.exercise2;

import java.io.File;

import java.io.FileNotFoundException;

import java.util.ArrayList;

import java.util.List;

import java.util.Scanner;

class Car {

private int id;

private String brand;

private String model;

private int year;

private String color;

private double price;

private String regNumber;

private String carNumber;

private String client;

private String clientPassNum;

public Car(int id, String brand, String model, int year, String color, double price, String regNumber, String carNumber, String client, String clientPassNum) {

this.id = id;

this.brand = brand;

this.model = model;

this.year = year;

this.color = color;

this.price = price;

this.regNumber = regNumber;

this.carNumber = carNumber;

this.client = client;

this.clientPassNum = clientPassNum;

}

public int getId() {

return id;

}

public String getBrand() {

return brand;

}

public String getModel() {

return model;

}

public int getYear() {

return year;

}

public String getColor() {

return color;

}

public double getPrice() {

return price;

}

public String getRegNumber() {

return regNumber;

}

public String getCarNumber() {

return carNumber;

}

public String getClient() {

return client;

}

public String getClientPassNum() {

return clientPassNum;

}

}

public class Main {

public static List<Car> readCarsFromFile(String fileName) {

List<Car> cars = new ArrayList<>();

try {

File file = new File(fileName);

if (!file.exists()) {

System.out.println("Файл не найден.");

return cars;

}

Scanner scanner = new Scanner(file);

while (scanner.hasNextLine()) {

String line = scanner.nextLine();

String[] data = line.split(",");

Car car = new Car(Integer.parseInt(data[0]), data[1], data[2], Integer.parseInt(data[3]), data[4], Double.parseDouble(data[5]), data[6], data[7], data[8], data[9]);

cars.add(car);

}

scanner.close();

} catch (FileNotFoundException e) {

System.out.println("Ошибка при чтении файла.");

}

return cars;

}

public static void displayAllCars(List<Car> cars) {

for (Car car : cars) {

System.out.println("ID: " + car.getId());

System.out.println("Марка: " + car.getBrand());

System.out.println("Модель: " + car.getModel());

System.out.println("Год: " + car.getYear());

System.out.println("Цвет: " + car.getColor());

System.out.println("Цена: " + car.getPrice());

System.out.println("Регистрационный номер: " + car.getRegNumber());

System.out.println("Номер автомобиля: " + car.getCarNumber());

System.out.println("Клиент: " + car.getClient());

System.out.println("Паспорт клиента: " + car.getClientPassNum());

System.out.println();

}

}

public static void displayCarsByBrand(List<Car> cars, String brand) {

List<Car> brandCars = new ArrayList<>();

for (Car car : cars) {

if (car.getBrand().equalsIgnoreCase(brand)) {

brandCars.add(car);

}

}

if (!brandCars.isEmpty()) {

System.out.println("Список автомобилей марки " + brand + ":");

displayAllCars(brandCars);

} else {

System.out.println("Нет автомобилей марки " + brand + ".");

}

}

public static void displayCarsByModelAndAge(List<Car> cars, String model, int age) {

List<Car> modelCars = new ArrayList<>();

for (Car car : cars) {

if (car.getModel().equalsIgnoreCase(model) && (2024 - car.getYear()) > age) {

modelCars.add(car);

}

}

if (!modelCars.isEmpty()) {

System.out.println("Список автомобилей модели " + model + ", которые эксплуатируются больше " + age + " лет:");

displayAllCars(modelCars);

} else {

System.out.println("Нет автомобилей модели " + model + ", которые эксплуатируются больше " + age + " лет.");

}

}

public static void displayCarsByYearAndPrice(List<Car> cars, int year, double price) {

List<Car> yearPriceCars = new ArrayList<>();

for (Car car : cars) {

if (car.getYear() == year && car.getPrice() > price) {

yearPriceCars.add(car);

}

}

if (!yearPriceCars.isEmpty()) {

System.out.println("Список автомобилей года выпуска " + year + ", цена которых больше " + price + ":");

displayAllCars(yearPriceCars);

} else {

System.out.println("Нет автомобилей года выпуска " + year + ", цена которых больше " + price + ".");

}

}

public static void displayRentalCars(List<Car> cars) {

List<Car> rentalCars = new ArrayList<>();

for (Car car : cars) {

if (!car.getClient().isEmpty()) {

rentalCars.add(car);

}

}

if (!rentalCars.isEmpty()) {

System.out.println("Список автомобилей, взятых на прокат:");

displayAllCars(rentalCars);

} else {

System.out.println("Нет автомобилей, взятых на прокат.");

}

}

public static void displayRentalCarsWithClients(List<Car> cars) {

List<Car> rentalCars = new ArrayList<>();

for (Car car : cars) {

if (!car.getClient().isEmpty()) {

rentalCars.add(car);

}

}

if (!rentalCars.isEmpty()) {

System.out.println("Список автомобилей, взятых на прокат с информацией о арендаторах:");

for (Car car : rentalCars) {

System.out.println("ID: " + car.getId());

System.out.println("Марка: " + car.getBrand());

System.out.println("Модель: " + car.getModel());

System.out.println("Клиент: " + car.getClient());

System.out.println("Паспорт клиента: " + car.getClientPassNum());

System.out.println();

}

} else {

System.out.println("Нет автомобилей, взятых на прокат.");

}

}

public static void main(String[] args) {

String fileName = "cars.txt";

List<Car> cars = readCarsFromFile(fileName);

Scanner input = new Scanner(System.in);

while (true) {

System.out.println("\nМеню:");

System.out.println("1. Вывести список всех автомобилей");

System.out.println("2. Вывести список автомобилей заданной марки");

System.out.println("3. Вывести список автомобилей заданной модели, которые эксплуатируются больше n лет");

System.out.println("4. Вывести список автомобилей заданного года выпуска, цена которых больше указанной");

System.out.println("5. Вывести список автомобилей, взятых на прокат");

System.out.println("6. Вывести список автомобилей, взятых на прокат с выводом личной информации об арендаторах");

System.out.println("7. Выход");

System.out.print("Выберите действие: ");

String choice = input.nextLine();

switch (choice) {

case "1":

displayAllCars(cars);

break;

case "2":

System.out.print("Введите марку автомобиля: ");

String brand = input.nextLine();

displayCarsByBrand(cars, brand);

break;

case "3":

System.out.print("Введите модель автомобиля: ");

String model = input.nextLine();

System.out.print("Введите количество лет эксплуатации: ");

int age = Integer.parseInt(input.nextLine());

displayCarsByModelAndAge(cars, model, age);

break;

case "4":

System.out.print("Введите год выпуска автомобилей: ");

int year = Integer.parseInt(input.nextLine());

System.out.print("Введите минимальную цену: ");

double minPrice = Double.parseDouble(input.nextLine());

displayCarsByYearAndPrice(cars, year, minPrice);

break;

case "5":

displayRentalCars(cars);

break;

case "6":

displayRentalCarsWithClients(cars);

break;

case "7":

System.out.println("Выход из программы.");

System.exit(0);

default:

System.out.println("Некорректный ввод. Пожалуйста, выберите действие из меню.");

}

}

}

}

**Результат программы:**

