

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ
“БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ”
ФАКУЛЬТЕТ ЭЛЕКТРОННО-ИНФОРМАЦИОННЫХ СИСТЕМ
КАФЕДРА ИНТЕЛЛЕКТУАЛЬНЫХ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

Отчет по лабораторной работе №3

Специальность ПО-9

Выполнил
А. С. Мисюк,
студент группы ПО-9
Проверил
А. А. Крощенко,
ст. преп. кафедры ИИТ,
«__» _____ 2024 г.

Брест 2024

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()

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.

Код программы:

```
public class EquilateralTriangle
{
    private double sideLength;

    /** Конструктор с параметром */
    public EquilateralTriangle(double sideLength)
    {
        this.sideLength = sideLength;
    }

    /** Метод для определения существования треугольника */
    public boolean isTriangleValid()
    {
        return sideLength > 0; // Треугольник существует, если длина стороны положительна
    }

    /** Метод для вычисления периметра */
    public double calculatePerimeter()
    {
        if (isTriangleValid()) {
            return 3 * sideLength;
        } else {
            return 0; // Возвращаем 0, если треугольник не существует
        }
    }
}
```

```

    }
}

/** Метод для вычисления площади */
public double calculateArea()
{
    if (isTriangleValid()) {
        // Формула площади равностороннего треугольника
        return (Math.sqrt(3) / 4) * Math.pow(sideLength, 2);
    } else {
        return 0; // Возвращаем 0, если треугольник не существует
    }
}

public double getSideLength()
{
    return sideLength;
}

public void setSideLength(double sideLength)
{
    this.sideLength = sideLength;
}

@Override
public String toString()
{
    return "EquilateralTriangle{" +
        "sideLength=" + sideLength +
        '}';
}

@Override
public boolean equals(Object obj)
{
    if (this == obj) return true;
    if (obj == null || getClass() != obj.getClass()) return false;
    EquilateralTriangle that = (EquilateralTriangle) obj;
    return Double.compare(that.sideLength, sideLength) == 0;
}
}

```

Пример

```

>java.exe Main
Triangle 1: EquilateralTriangle{sideLength=5.0}
Perimeter: 15.0
Area: 10.825317547305483
Triangle 2: EquilateralTriangle{sideLength=7.5}
Is Triangle 2 valid triangle? true
Are triangles equal? false
Triangle 3: EquilateralTriangle{sideLength=7.5}
Are triangles (2 and 3) equal? true

```

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.

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.

Код программы:

```
public class Car
{
    {...}

    public Car(String fileLine) {
        Scanner scanner = new Scanner(fileLine);

        this.id = scanner.nextInt();
        this.brand = scanner.next();
        this.model = scanner.next();
        this.year = scanner.nextInt();
        this.color = scanner.next();
        this.price = scanner.nextDouble();
        this.registrationNumber = scanner.next();
        this.carNumber = scanner.next();

        if (scanner.hasNext()) {
            this.renterName = scanner.next();
            this.renterPassport = scanner.next();
        }
    }
}
```

```

        scanner.close();
    }

    public String getCarInfo()
    {
        StringBuilder carInfo = new StringBuilder();
        carInfo.append("ID: ").append(id)
            .append("\tBrand: ").append(brand)
            .append("\tModel: ").append(model)
            .append("\tYear: ").append(year)
            .append("\tColor: ").append(color)
            .append("\tPrice: $").append(price)
            .append("\tRegistration Number: ").append(registrationNumber)
            .append("\tCar Number: ").append(carNumber);

        // Добавляем опциональные атрибуты, если они присутствуют
        if (renterName != null)
        {
            carInfo.append("\tRenter Name: ").append(renterName);
        }

        if (renterPassport != null)
        {
            carInfo.append("\tRenter Passport: ").append(renterPassport);
        }

        return carInfo.toString();
    }

    public String getCarsRenterInfo()
    {
        StringBuilder renterInfo = new StringBuilder();

        if (renterName != null)
        {
            renterInfo.append("Renter Name: ").append(renterName);
        }

        if (renterPassport != null)
        {
            renterInfo.append("\tRenter Passport: ").append(renterPassport);
        }

        return renterInfo.toString();
    }

    {...}
}

class CarRentalSystem
{
    private final ArrayList<Car> carsList;

    public CarRentalSystem(String file)
    {
        this.carsList = new ArrayList<>();
        this.importCarsFromFile(file);
    }

    /** Метод для импорта списка автомобилей из файла */
    public void importCarsFromFile(String filePath)
    {
        try (BufferedReader reader = new BufferedReader(new FileReader(filePath)))
        {
            String line;
            while ((line = reader.readLine()) != null)

```

```

        {
            Car newCar = new Car(line);
            carsList.add(newCar);
        }
    } catch (IOException e)
    {
        e.printStackTrace();
    }
}

/** @return список автомобилей как строку */
public String getCars()
{
    return getCars(carsList);
}

/** @return список автомобилей как строку */
public String getCars(ArrayList<Car> carsListL)
{
    StringBuilder carsInfo = new StringBuilder();
    for (Car car : carsListL) {
        carsInfo.append(car.getCarInfo()).append("\n");
    }
    return carsInfo.toString();
}

/** вывод списка автомобилей заданной марки */
public String getCarsByBrand(String brand)
{
    ArrayList<Car> filteredCars = new ArrayList<>();

    for (Car car : carsList) {
        if (car.getBrand().equalsIgnoreCase(brand)) {
            filteredCars.add(car);
        }
    }

    return getCars(filteredCars);
}

/** вывод списка автомобилей заданной модели, которые эксплуатируются больше n лет */
public String getCarsByModelAndMoreThenNYears(String model, int n_years)
{
    ArrayList<Car> filteredCars = new ArrayList<>();
    int year = LocalDateTime.now().getYear();

    for (Car car : carsList) {
        if (car.getModel().equalsIgnoreCase(model) && year - car.getYear() >= n_years) {
            filteredCars.add(car);
        }
    }

    return getCars(filteredCars);
}

/** вывод списка автомобилей заданного года выпуска, цена которых больше указанной */
public String getCarsByYearAndMoreThanNPrice(int year, double n_price)
{
    ArrayList<Car> filteredCars = new ArrayList<>();

    for (Car car : carsList) {
        if (car.getYear() == year && car.getPrice() > n_price) {
            filteredCars.add(car);
        }
    }

    return getCars(filteredCars);
}

```

```

/** вывод списка автомобилей, взятых на прокат */
public String getCarsThatAreRented()
{
    ArrayList<Car> filteredCars = new ArrayList<>();

    for (Car car : carsList) {
        if (car.getRenterName() != null) {
            filteredCars.add(car);
        }
    }

    return getCars(filteredCars);
}

/** вывод списка автомобилей, взятых на прокат с выводом личной информации об арендаторах */
public String getCarsRentersInfo()
{
    ArrayList<Car> filteredCars = new ArrayList<>();

    for (Car car : carsList) {
        if (car.getRenterName() != null) {
            filteredCars.add(car);
        }
    }

    StringBuilder carsInfo = new StringBuilder();
    for (Car car : filteredCars) {
        carsInfo.append(car.getCarsRenterInfo()).append("\n");
    }

    return carsInfo.toString();
}
}

```

Пример

```

>java.exe Main
ID: 1  Brand: Toyota      Model: Camry Year: 2018  Color: Silver      Price:
$25000.0  Registration Number: ABC123  Car Number: 12345
ID: 2  Brand: Honda Model: Accord      Year: 2019  Color: Blue Price: $28000.0
Registration Number: XYZ456  Car Number: 67890
ID: 3  Brand: Ford Model: Mustang      Year: 2017  Color: Red  Price: $35000.0
Registration Number: DEF789  Car Number: 54321 Renter Name: Bob_Jones  Renter
Passport: EF8765432
ID: 4  Brand: Chevrolet  Model: Malibu      Year: 2020  Color: Black Price:
$30000.0  Registration Number: GHI012  Car Number: 67890 Renter Name: Emily_Brown
Renter Passport: GH7654321
ID: 5  Brand: BMW  Model: X5  Year: 2019  Color: White Price: $50000.0
Registration Number: JKL345  Car Number: 98765 Renter Name: Michael_Williams  Renter
Passport: IJ6543210
ID: 6  Brand: Mercedes-Benz  Model: C-Class      Year: 2018  Color: Gray Price:
$45000.0  Registration Number: MNO678  Car Number: 43210
ID: 7  Brand: Nissan      Model: Altima      Year: 2016  Color: Silver      Price:
$20000.0  Registration Number: PQR901  Car Number: 13579 Renter      Name:
Sophia_Jackson  Renter Passport: OP4321098
ID: 8  Brand: Volkswagen  Model: Golf Year: 2021  Color: Blue Price: $28000.0
Registration Number: STU234  Car Number: 24680 Renter Name: Liam_Wilson Renter
Passport: QR3210987

```

ID: 9 Brand: Hyundai Model: Sonata Year: 2017 Color: Green Price:
\$23000.0 Registration Number: VWX567 Car Number: 98765
ID: 10 Brand: Tesla Model: Model_3 Year: 2022 Color: Silver Price:
\$60000.0 Registration Number: YZA890 Car Number: 34567 Renter Name: Noah_Taylor
Renter Passport: ZY1098765

ID: 3 Brand: Ford Model: Mustang Year: 2017 Color: Red Price: \$35000.0
Registration Number: DEF789 Car Number: 54321 Renter Name: Bob_Jones Renter
Passport: EF8765432

ID: 8 Brand: Volkswagen Model: Golf Year: 2021 Color: Blue Price: \$28000.0
Registration Number: STU234 Car Number: 24680 Renter Name: Liam_Wilson Renter
Passport: QR3210987

ID: 10 Brand: Tesla Model: Model_3 Year: 2022 Color: Silver Price:
\$60000.0 Registration Number: YZA890 Car Number: 34567 Renter Name: Noah_Taylor
Renter Passport: ZY1098765

ID: 3 Brand: Ford Model: Mustang Year: 2017 Color: Red Price: \$35000.0
Registration Number: DEF789 Car Number: 54321 Renter Name: Bob_Jones Renter
Passport: EF8765432

ID: 4 Brand: Chevrolet Model: Malibu Year: 2020 Color: Black Price:
\$30000.0 Registration Number: GHI012 Car Number: 67890 Renter Name: Emily_Brown
Renter Passport: GH7654321

ID: 5 Brand: BMW Model: X5 Year: 2019 Color: White Price: \$50000.0
Registration Number: JKL345 Car Number: 98765 Renter Name: Michael_Williams Renter
Passport: IJ6543210

ID: 7 Brand: Nissan Model: Altima Year: 2016 Color: Silver Price:
\$20000.0 Registration Number: PQR901 Car Number: 13579 Renter Name:
Sophia_Jackson Renter Passport: OP4321098

ID: 8 Brand: Volkswagen Model: Golf Year: 2021 Color: Blue Price: \$28000.0
Registration Number: STU234 Car Number: 24680 Renter Name: Liam_Wilson Renter
Passport: QR3210987

ID: 10 Brand: Tesla Model: Model_3 Year: 2022 Color: Silver Price:
\$60000.0 Registration Number: YZA890 Car Number: 34567 Renter Name: Noah_Taylor
Renter Passport: ZY1098765

Renter Name: Bob_Jones Renter Passport: EF8765432
Renter Name: Emily_Brown Renter Passport: GH7654321
Renter Name: Michael_Williams Renter Passport: IJ6543210
Renter Name: Sophia_Jackson Renter Passport: OP4321098
Renter Name: Liam_Wilson Renter Passport: QR3210987
Renter Name: Noah_Taylor Renter Passport: ZY1098765

Process finished with exit code 0

Рисунки с результатами работы программы

```
Task1 X
C:\Users\misij\.jdk\openjdk-21.0.2\bin\java.exe
Triangle 1: EquilateralTriangle{sideLength=5.0}
Perimeter: 15.0
Area: 10.825317547305483
Triangle 2: EquilateralTriangle{sideLength=7.5}
Is Triangle 2 valid triangle? true
Are triangles equal? false
Triangle 3: EquilateralTriangle{sideLength=7.5}
Are triangles (2 and 3) equal? true

Process finished with exit code 0
```

```
Task2 X
C:\Users\misij\.jdk\openjdk-21.0.2\bin\java.exe --javaagent:D:\JetBrains\IntelliJ IDEA Community Edition 2023.1.2\lib\idea_rt.jar=23457:D:\JetBrains\IntelliJ ID
ID: 1 Brand: Toyota Model: Camry Year: 2018 Color: Silver Price: $25000.0 Registration Number: ABC123 Car Number: 12345
ID: 2 Brand: Honda Model: Accord Year: 2019 Color: Blue Price: $28000.0 Registration Number: XYZ456 Car Number: 67890
ID: 3 Brand: Ford Model: Mustang Year: 2017 Color: Red Price: $35000.0 Registration Number: DEF789 Car Number: 54321 Renter Name: Bob_Jones Renter Passport: EF8765432
ID: 4 Brand: Chevrolet Model: Malibu Year: 2020 Color: Black Price: $30000.0 Registration Number: GHI012 Car Number: 67890 Renter Name: Emily_Brown Renter Passport: GH7654321
ID: 5 Brand: BMW Model: X5 Year: 2019 Color: White Price: $50000.0 Registration Number: JKL345 Car Number: 98765 Renter Name: Michael_Williams Renter Passport: IJK432109
ID: 6 Brand: Mercedes-Benz Model: C-Class Year: 2018 Color: Gray Price: $45000.0 Registration Number: MNO678 Car Number: 43210
ID: 7 Brand: Nissan Model: Altima Year: 2016 Color: Silver Price: $20000.0 Registration Number: PQR901 Car Number: 13579 Renter Name: Sophia_Jackson Renter Passport: LMN567890
ID: 8 Brand: Volkswagen Model: Golf Year: 2021 Color: Blue Price: $28000.0 Registration Number: STU234 Car Number: 24680 Renter Name: Liam_Wilson Renter Passport: OPQ123456
ID: 9 Brand: Hyundai Model: Sonata Year: 2017 Color: Green Price: $23000.0 Registration Number: VWX567 Car Number: 98765
ID: 10 Brand: Tesla Model: Model_3 Year: 2022 Color: Silver Price: $60000.0 Registration Number: YZA890 Car Number: 34567 Renter Name: Noah_Taylor Renter Passport: RST901234

ID: 3 Brand: Ford Model: Mustang Year: 2017 Color: Red Price: $35000.0 Registration Number: DEF789 Car Number: 54321 Renter Name: Bob_Jones Renter Passport: EF8765432

ID: 8 Brand: Volkswagen Model: Golf Year: 2021 Color: Blue Price: $28000.0 Registration Number: STU234 Car Number: 24680 Renter Name: Liam_Wilson Renter Passport: OPQ123456

ID: 10 Brand: Tesla Model: Model_3 Year: 2022 Color: Silver Price: $60000.0 Registration Number: YZA890 Car Number: 34567 Renter Name: Noah_Taylor Renter Passport: RST901234

ID: 3 Brand: Ford Model: Mustang Year: 2017 Color: Red Price: $35000.0 Registration Number: DEF789 Car Number: 54321 Renter Name: Bob_Jones Renter Passport: EF8765432
ID: 4 Brand: Chevrolet Model: Malibu Year: 2020 Color: Black Price: $30000.0 Registration Number: GHI012 Car Number: 67890 Renter Name: Emily_Brown Renter Passport: GH7654321
ID: 5 Brand: BMW Model: X5 Year: 2019 Color: White Price: $50000.0 Registration Number: JKL345 Car Number: 98765 Renter Name: Michael_Williams Renter Passport: IJK432109
ID: 7 Brand: Nissan Model: Altima Year: 2016 Color: Silver Price: $20000.0 Registration Number: PQR901 Car Number: 13579 Renter Name: Sophia_Jackson Renter Passport: LMN567890
ID: 8 Brand: Volkswagen Model: Golf Year: 2021 Color: Blue Price: $28000.0 Registration Number: STU234 Car Number: 24680 Renter Name: Liam_Wilson Renter Passport: OPQ123456
ID: 10 Brand: Tesla Model: Model_3 Year: 2022 Color: Silver Price: $60000.0 Registration Number: YZA890 Car Number: 34567 Renter Name: Noah_Taylor Renter Passport: RST901234

Renter Name: Bob_Jones Renter Passport: EF8765432
Renter Name: Emily_Brown Renter Passport: GH7654321
Run TODO Problems Terminal Services Build
are up-to-date (moments ago) 40:1 CRLF UTF-8 4 spaces master
```

Вывод: исследовал создание классов и их использование на языке программирования Java.