Одесский колледж компьютерных технологий «Сервер»

Отчет по практической работе №6

«Абстрактные классы и механизмы. Механизм обратного вызова»

Выполнил:

студент группы К19.1

Фучило Богдан

Проверил:

Рудниченко Н.Д.

Одесса 2022

Содержание

[ВВЕДЕНИЕ 2](#_3znysh7)

[ТЕОРИТИЧЕСКАЯ ЧАСТЬ 3](#_2et92p0)

[ПРАКТИЧЕСКАЯ ЧАСТЬ 6](#_tyjcwt)

[ВЫВОД 14](#_3dy6vkm)

[ЛИТЕРАТУРА 15](#_1t3h5sf)

ВВЕДЕНИЕ

Цель данной лабораторной работы ознакомиться с с механизмом абстрактных классов и интерфейсов, разобраться с тем, когда и для чего необходимо использовать абстрактные классы и интерфейсы, а так-же изучить использование механизма обратного вызова.

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

**Интерфейс** – это семантическая и синтаксическая конструкция в коде программы, используемая для специфицирования услуг, предоставляемых классом или компонентом. Интерфейс определяет границу взаимодействия между классами или компонентами, специфицируя определенную абстракцию, которую осуществляет реализующая сторона.

**КЛАСС**

Каждый объект определяется общим шаблоном, который называется классом. В рамках класса задается общий шаблон, структура, на основе которой затем создаются объекты. Данные, относящиеся к классу, называются полями класса, а программный код для их обработки – методами класса. Поля и методы иногда называют общим термином – члены класса.

В классе описываются, какого типа данные относятся к классу, а также то, какие методы применяются к этим данным. Затем, в программе на основе того или иного класса создается экземпляр класса (объект), в котором указываются конкретные значения полей и выполняются необходимые действия над ними.

Согласно конвенциям кода Java:

• каждый класс должен содержаться в своем отдельном файле с расширением .java;

• название файла должно совпадать с названием класса;

• класс должен быть именем существительным;

• имя класса должно его описывать;

• имя класса начинается с большой буквы;

• если имя состоит из нескольких слов, то каждое слово начинается с большой буквы.

**Абстрактный класс**

Абстрактный, это базовый класс, который не подразумевает под собой создание экземпляров.

Абстрактные классы реализуют на практике один из принципов ООП — полиморфизм. Абстрактный класс может содержать (и не содержать) абстрактные методы и свойства. Абстрактный метод не реализуется для класса, в котором описан, однако должен быть реализован для его не абстрактных потомков. Абстрактные классы представляют собой наиболее общие абстракции, то есть имеющие наибольший объём и наименьшее содержание.

ПРАКТИЧЕСКАЯ ЧАСТЬ

1. В пакете carshop.cars создать абстрактный класс Саг:

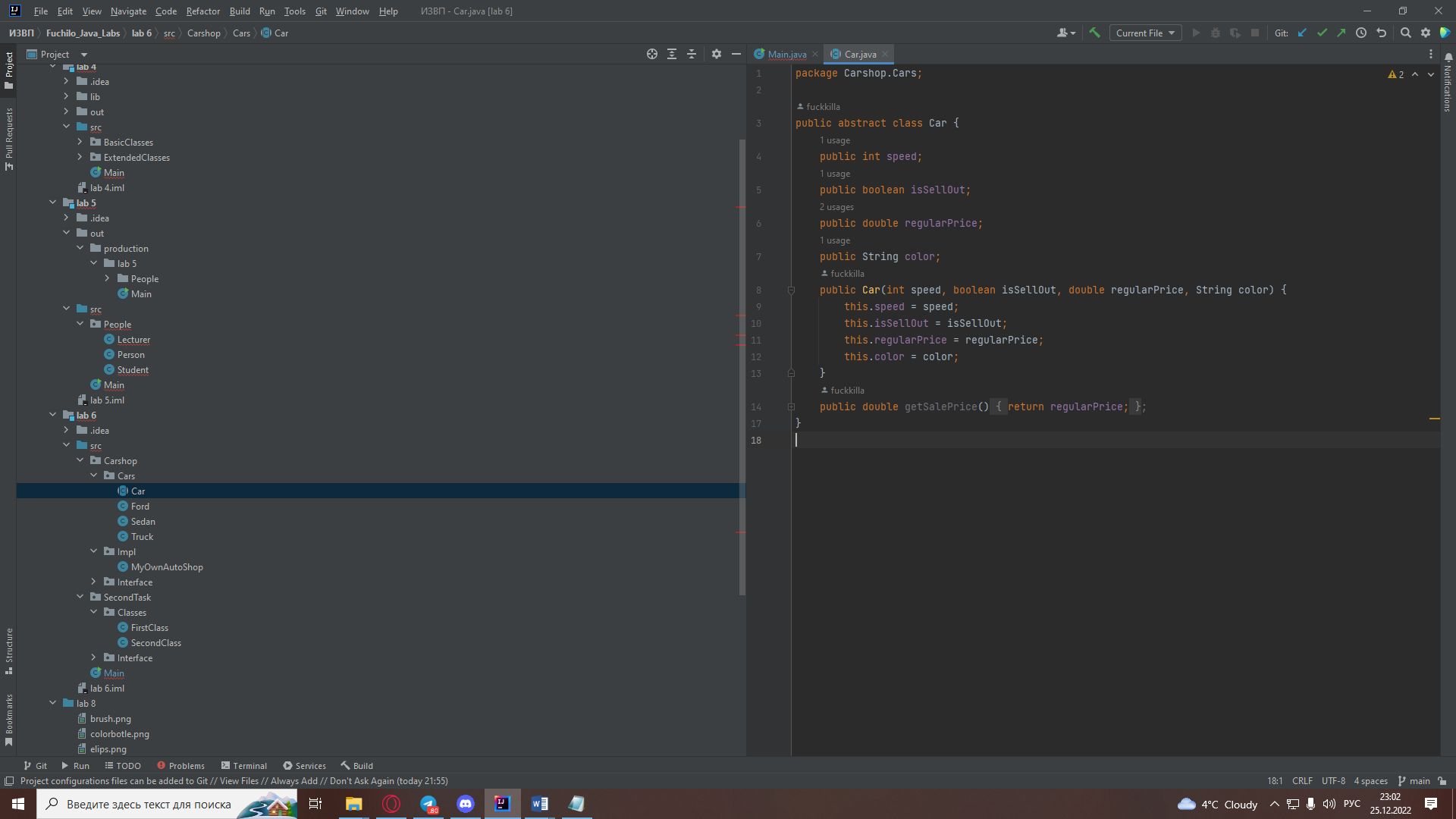


Рисунок 1 – Cоздание абстрактного класса Саг

2. В пакете carshop.cars создать класс Truck, расширяющий класс Саг:

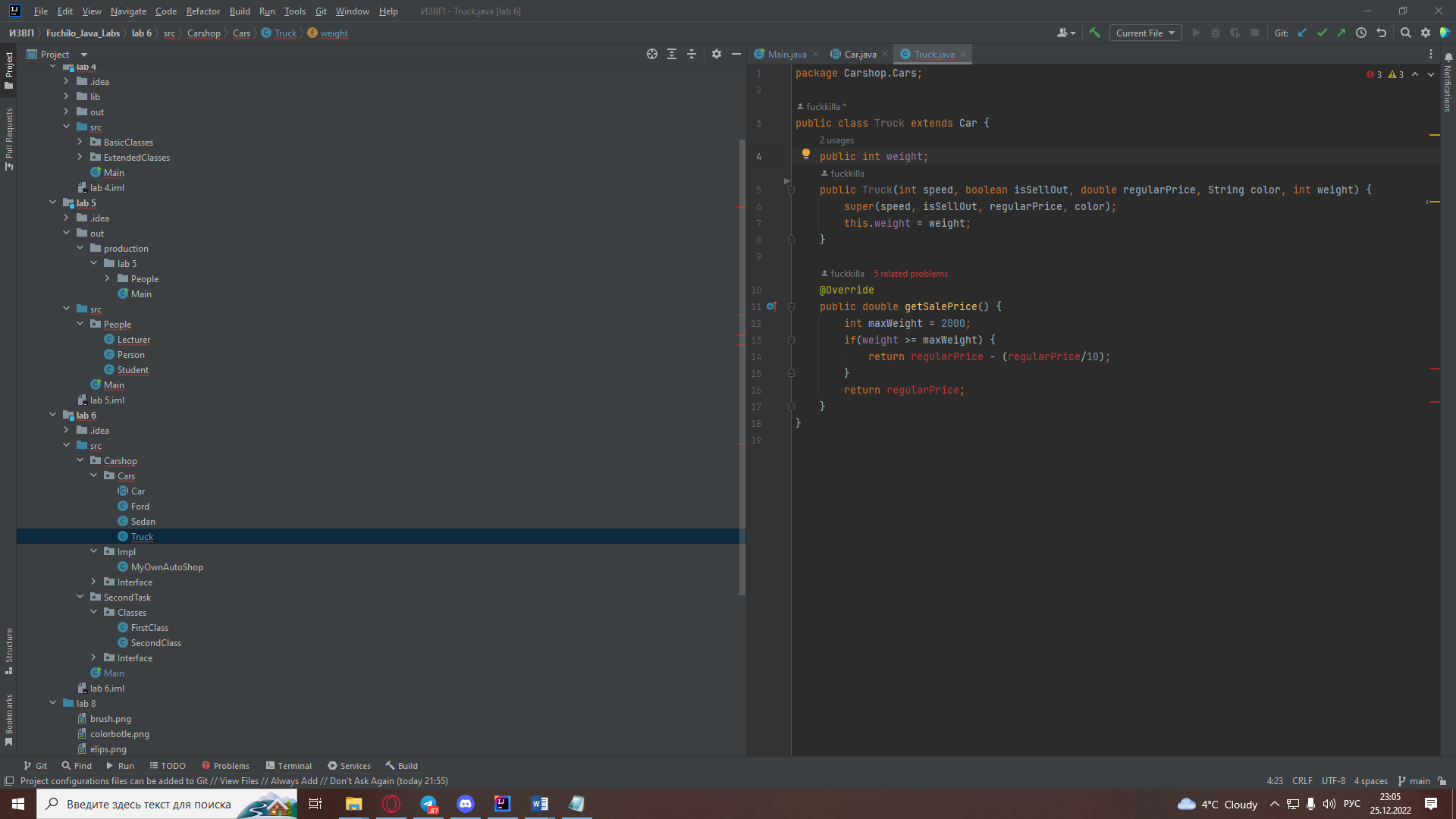


Рисунок 2 – Cоздание класса Truck

3. В пакете carshop.cars создать класс Ford, расширяющий класс Саг:

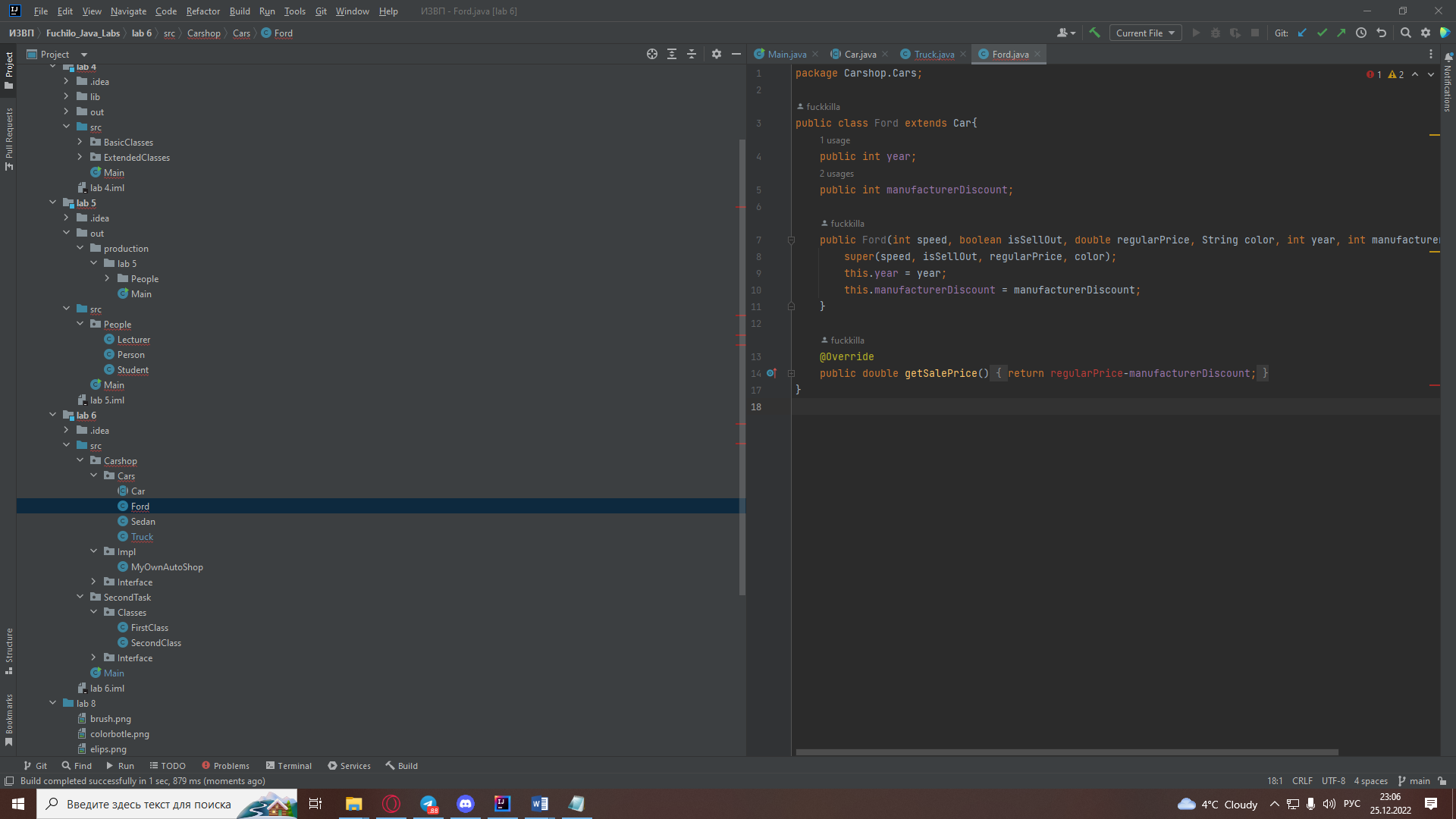


Рисунок 3 – Cоздание класса Ford

4. В пакете carshop.cars создать класс Sedan, расширяющий класс Саг:

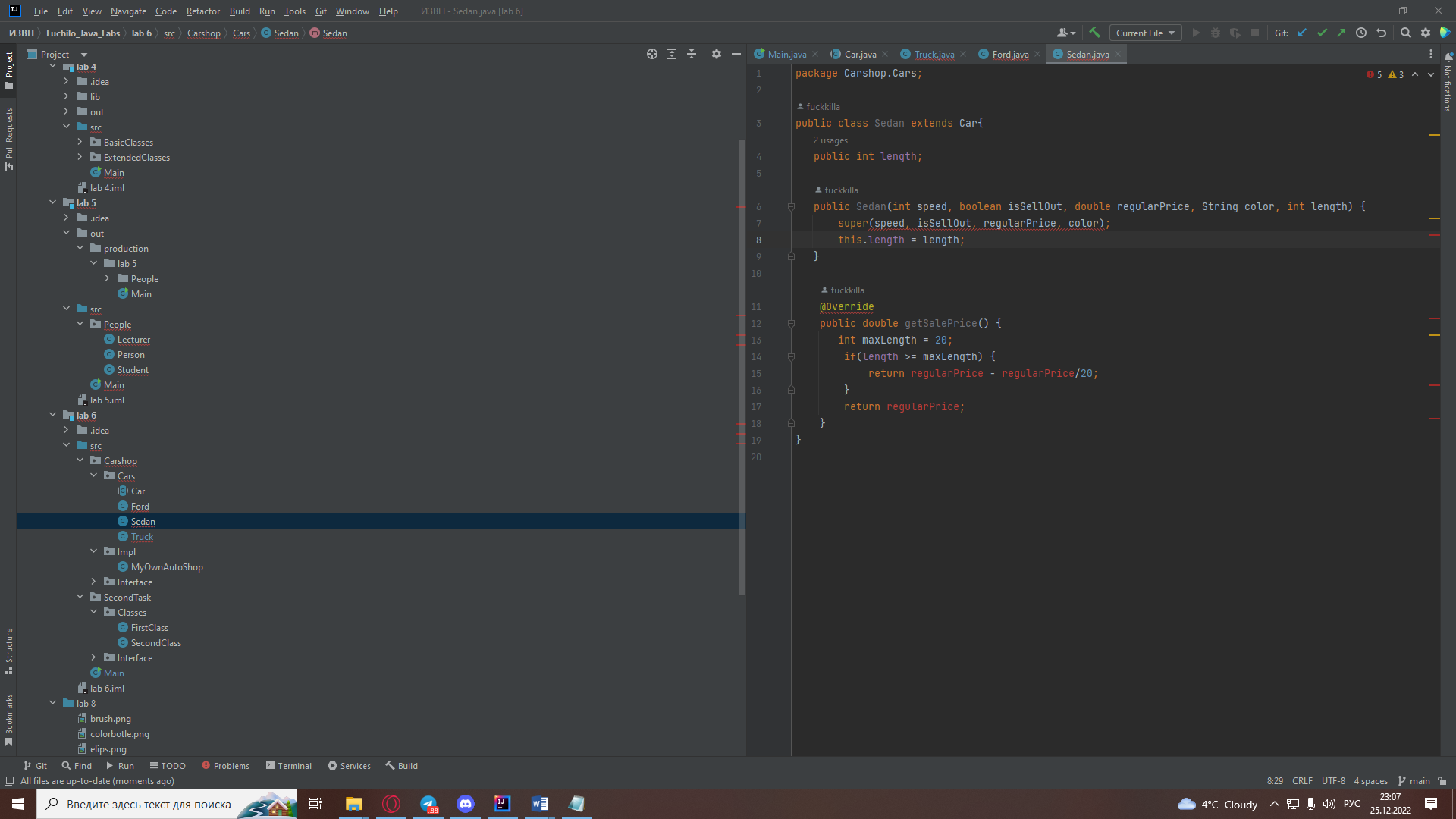


Рисунок 4 – Cоздание класса Sedan

5. В пакете carshop.interface создать два интерфейса Admin и Customer:

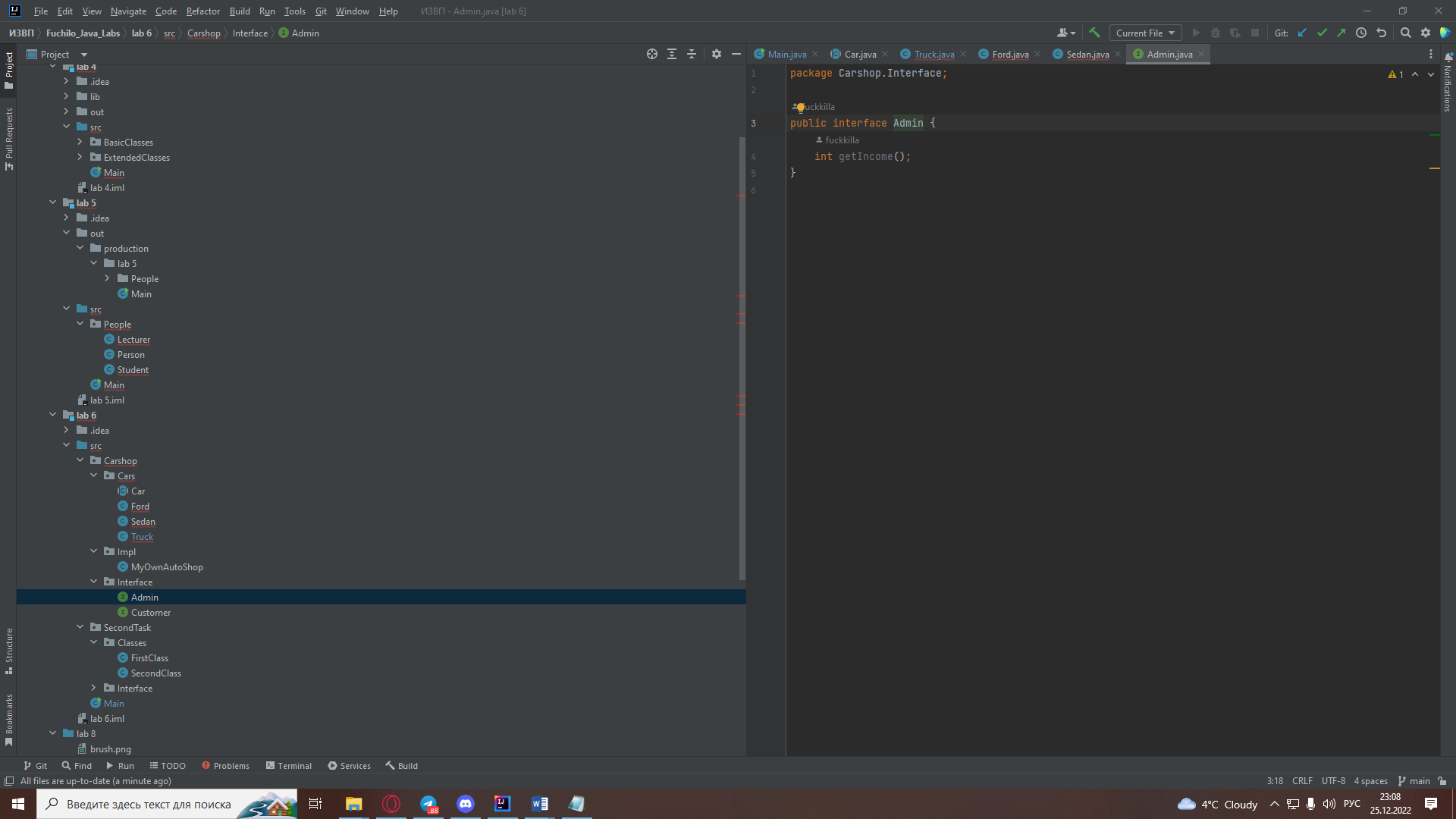


Рисунок 5 – Cоздание интерфейса Admin

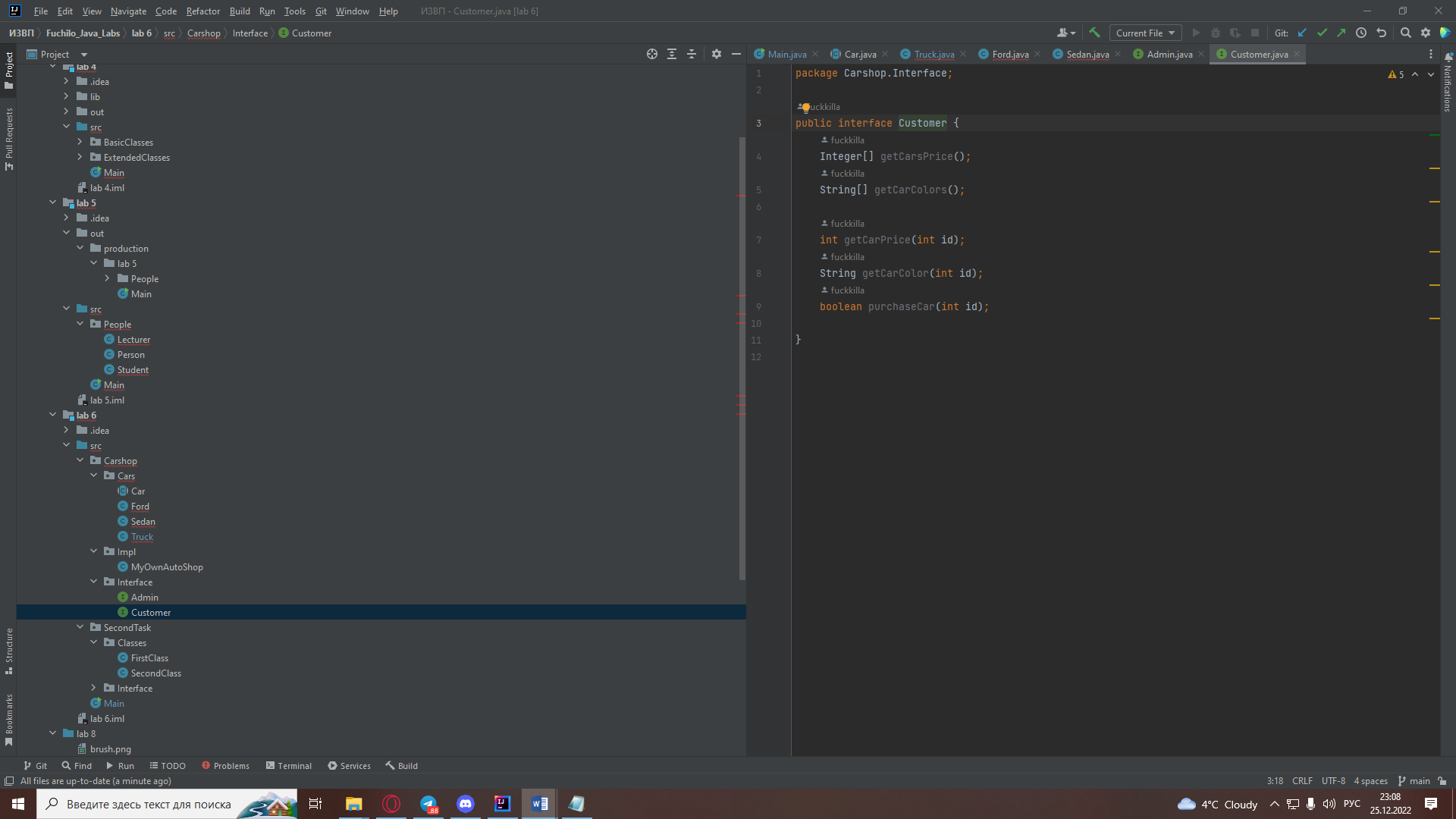


Рисунок 6 – Cоздание интерфейса Customer

6. В пакете carshop.impl создать класс MyOwnAutoShop реализующий интерфейсы Admin и Customer.:

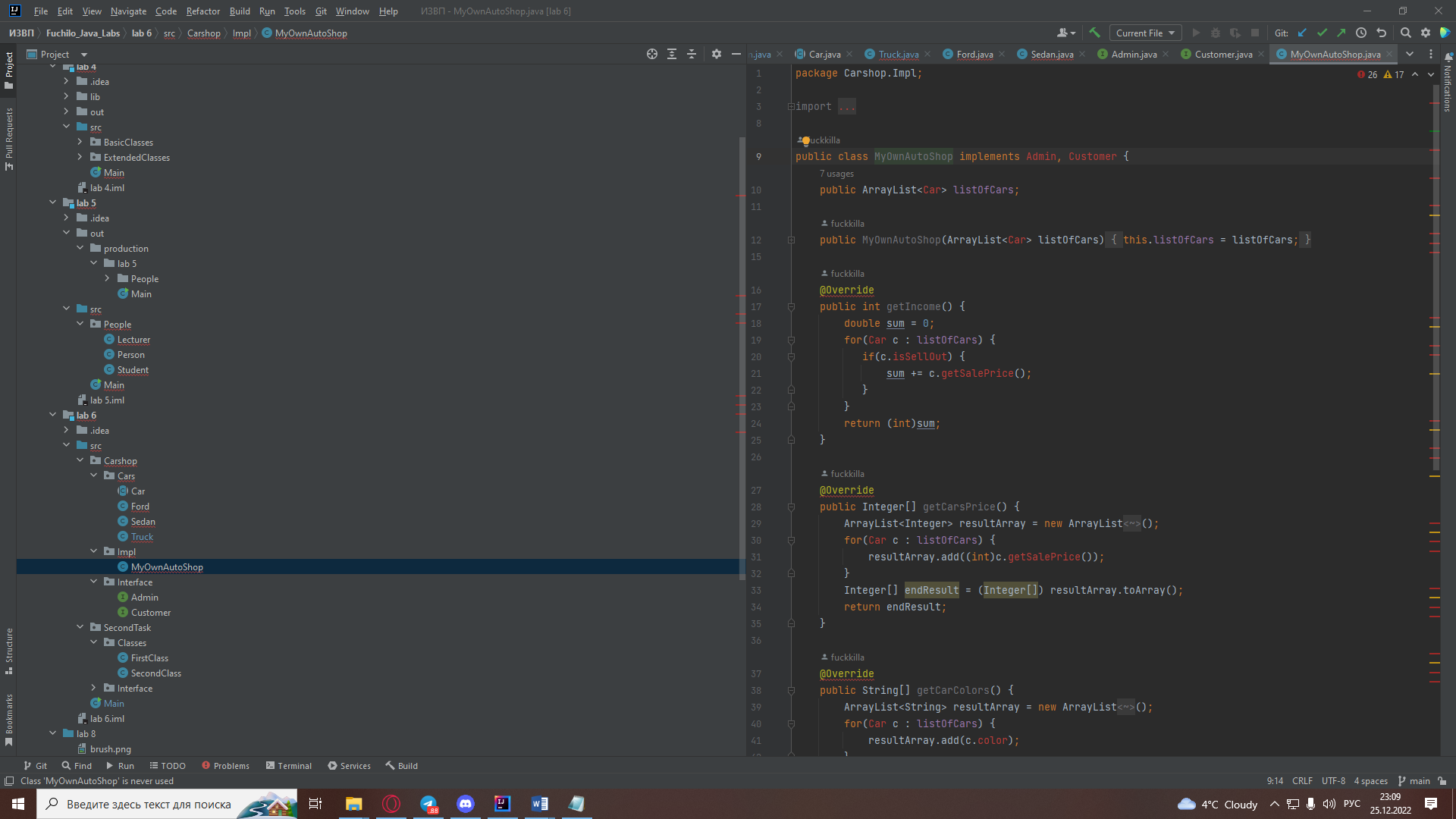


Рисунок 7 – Cоздание класса MyOwnAutoShop

ВЫВОД

В данной лабораторной работе я ознакомился с механизмами абстрактных классов и интерфейсов, а так же разобрался с тем, когда, как и в каких случаях их стоит использовать.

ЛИТЕРАТУРА

1. Множественное наследование, абстрактные классы, интерфейсы <http://kostin.ws/java/java-abstract-and-interfaces.html>

<https://github.com/fuckkilla/Fuchilo_Java_Labs>