Description of application

Decorator pattern

Dimitar Markov – ei4s1

**Intro**

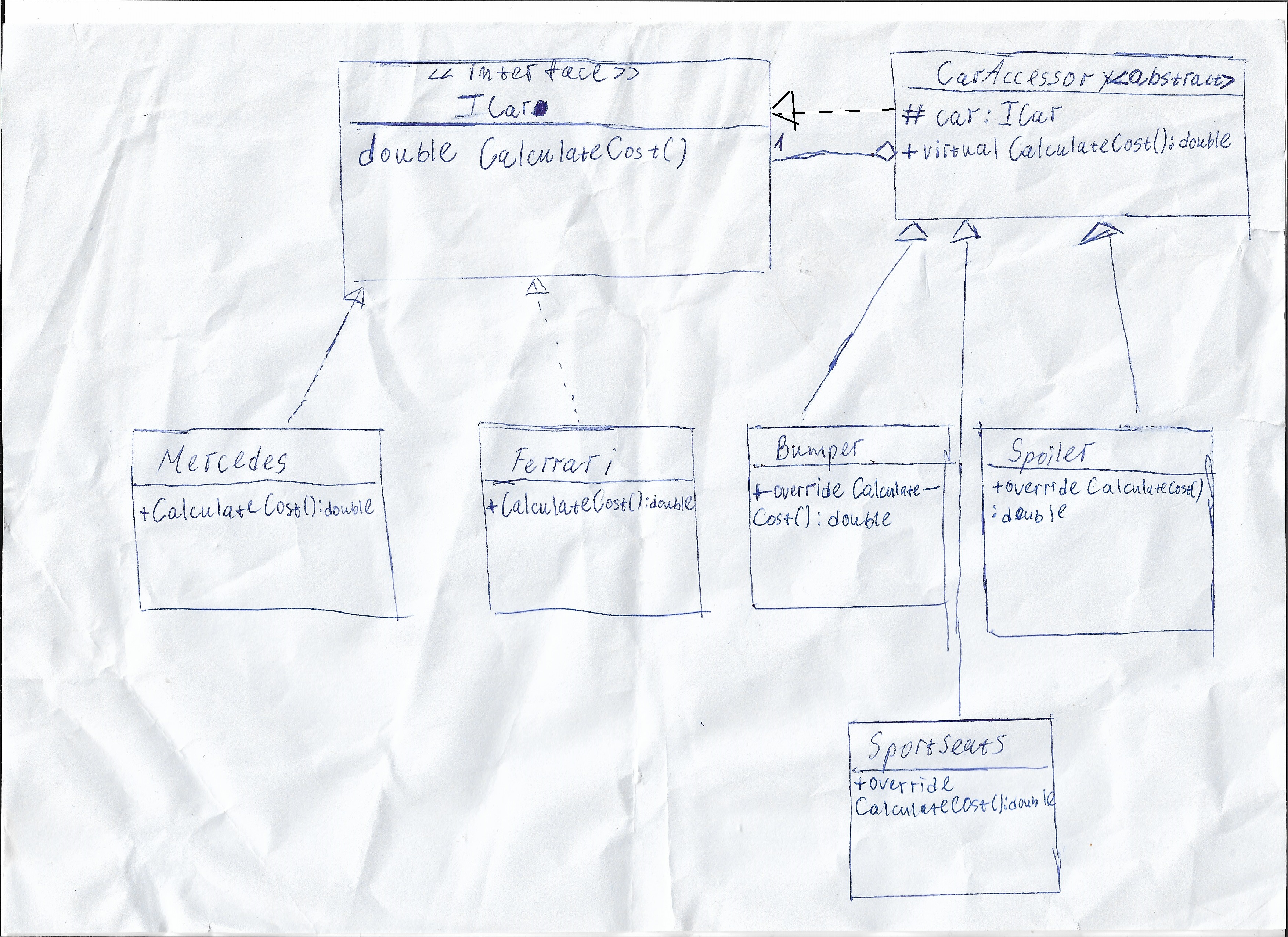
The functionality of the application is regarding to cars and car accessories. The GUI gives you the possibility to buy a car by pressing a button and selecting accessories from the list box and at the end the total price is displayed. The implementation is minimalistic and its only purpose is to display usage of the decorator pattern.

**Architecture and organization**

The project consists of 4 divisions (class libraries)

1. Client – The GUI and the starting point of the app
2. Cars – Holding classes related to cars(Components)
3. Accessories – Holding classes related to car accessories(Decorations)
4. Tests

If we summarize all the classes, we get that class diagram.



If we relate the classes to the decorator pattern we can give them the following roles.

* ICar - Component
* Mercedes – Concrete component
* Ferrari – Concrete component
* CarAccessory - Decorator
* Bumber – Concrete decorator
* Spoiler – Concrete decorator
* SportSeats – Concrete decorator

The implementation allows easier update of the catalog with accessories without modifying previous code, this makes it very flexible and reusable. It also allows adding accessories to given a given car in runtime and modifying the object dynamically.