

Our client wants us to design a configurator for it's car company. They are able to implement several additions according to their client's requests. They have 4 different kind of car types which are Sedan, Coupe, Hatchback and Suv. And their clients may add some features like sunroof, self-parking, steel wheels, surround sound system, navigation and leather seats. Features are optional therefore i need to use **Decorator Pattern** to let my client produce empty cars which have nothing additional or depends on the requests they may produce cars with some features.

So that we have abstract Car class which has getDescription method returns description and abstract price method. Sedan, Coupe, HatchBack, Suv are concrete classes which extend Car. Configurer extends abstract car class and it has 6 of classes decorate 4 different type of cars.

