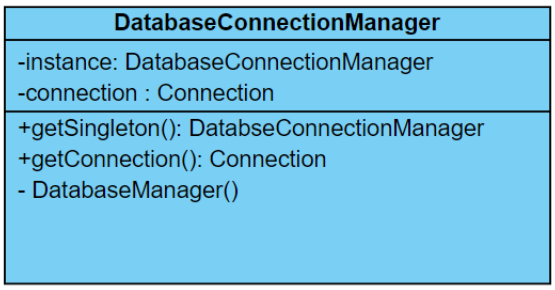
Kreacijski paterni

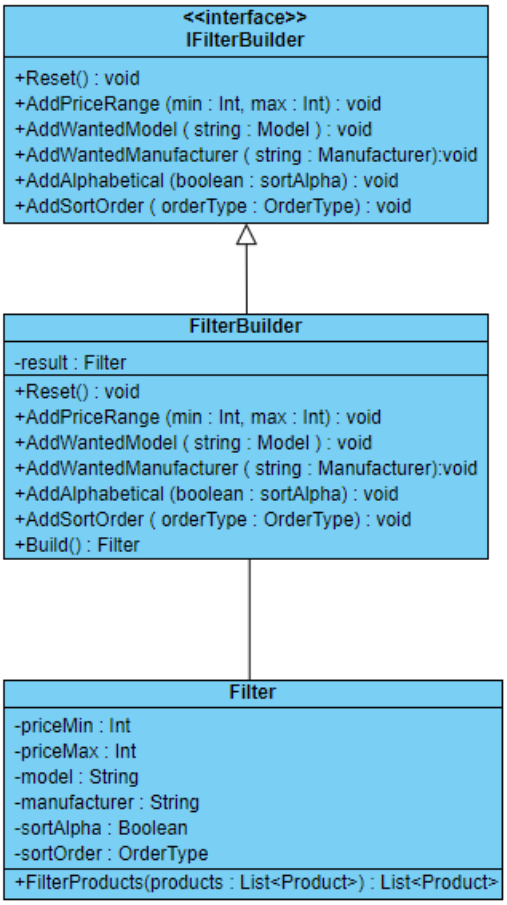
Singleton

Database Connection Manager. Možemo koristiti singleton kreacijski patern za učinkovito upravljanje vezama baze podataka, osiguravajući da samo jedna instanca rukuje svim interakcijama baze podataka. Stvorit ćemo klasu DatabaseManager s privatnim konstruktorom i statičkom metodom za pružanje jedne instance. Ova klasa će upravljati svim vezama baze podataka.



Builder

Builder pattern ćemo iskoristiti sa klasom Filter. Ova klasa ima mnoge atribute poput načina sortiranja, price range, specifični zahtjevi za određenim modelima i sl. Da bi olakšali kreiranje ovih objekata te da ne bi imali nepotrebne inicijalizacije (jer neki atributi neće biti iskorišteni npr. korisnik želi sve BMW proizvode ali nije ga briga za price-range) koristimo builder pattern.



Prototype

Potencijalni način upotrebe prototype paterna bi bio kada kloniramo objekat tipa Auto kada želimo kreiratu kopiju već postojećeg automobila sa sitnim izmjenama (Npr. Boja automobila)

Factory

Definirali bismo sučelje za stvaranje objekta, ali dopustili potklasama da mijenjaju tip objekata koji će biti kreirani. Za rukovanje stvaranja različitih tipova automobila (npr. limuzina, SUV, kamion) bez navođenja tačne klase.

Abstract Factory

Ideja za implementaciju abtract factory paterna bi bila stvaranje obitelji povezanih ili ovisnih objekata, kao što su različite vrste automobilskih komponenti (npr. motor, kotač) za različite modele automobila. A mplementirali bi tako što bi definirali sučelja za svaku vrstu komponente i konkretne tvornice za različite obitelji automobila.