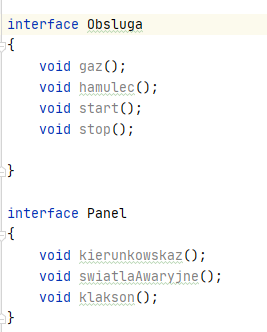
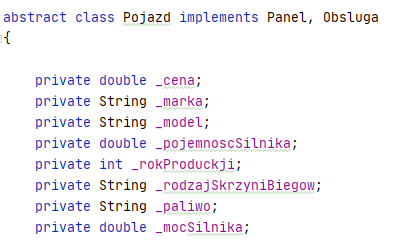
* Interface

Interface jest wykorzystywany do tego aby wymóc na danej klasie nadpisanie funkcji podanych w interfejsie. Zaletą interfejsu jest to ze można go implementować do klasy. Możemy zaimplantować kilka interfejsów jak w poniższym przykładzie.



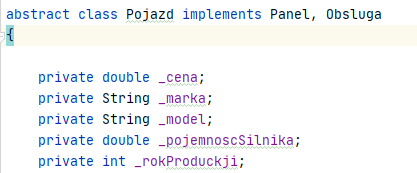


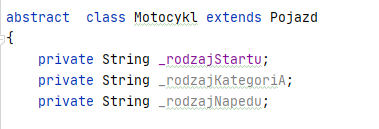
Zdecydowałem się na interfejs w danym przykładzie, ponieważ każdy pojazd może wykonywać takie funkcje jak gaz, hamulec itd. Różnią się one jedynie poszczególnymi cechami które zostały napisane w odpowiednim miejscu w kodzie.

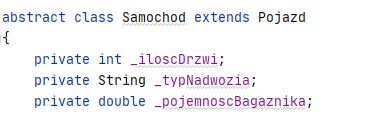
* Klasa Abstrakcyjna

Klasy abstrakcyjne to takie klasy, które nie dopuszczają możliwości stworzenia swojej instancji. Stosujemy je po to aby utworzyć „szkielet”, z którego w późniejszych etapach programu będziemy dziedziczyć cechy do innych klas.

Zdecydowałem się utworzyć abstrakcyjna klasę Pojazdu, Samochodu oraz Motocyklu, ponieważ nie ma potrzeby tworzenia instancji np. Pojazdu bo pełni on tu role wcześniej wspomnianego szkieletu, z którego cechy dziedziczą inne klasy np. Samochód Sportowy.



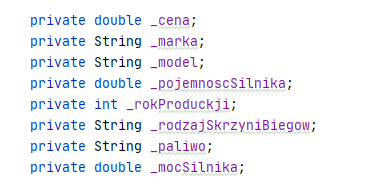




* Hermetyzacja

Hermetyzacja służy do tego aby chronić dane. Nie zawsze chcemy aby dostęp do ważnych danych był z poza klasy, wtedy więc korzystamy z hermetyzacji aby dane te nie mogły ulec zmianie poza klasą. Do ich zmiany stosuje się odpowiednie metody takie jak getery i setery.

W poniższym przykładzie wszystkie zmienne są ustawione na private z powodu, iż nie ma potrzeby dostępu do nich z poza klasy oraz przechowują one ważne i kluczowe dane, które nie mogą w bezpośredni sposób ulegać zmianom.

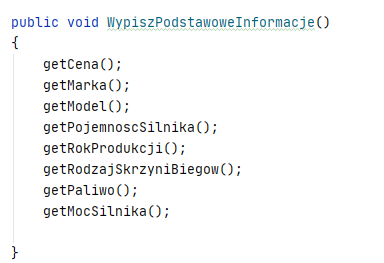


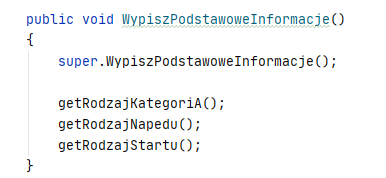
* Polimorfizm

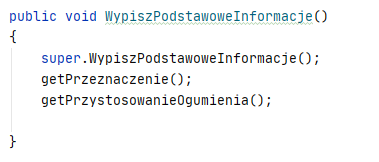
Polimorfizm jest to traktowanie obiektów pewnych klas jako obiekty innych klas z zachowaniem pełnej funkcjonalności. Prezentując to na poniższym przykładzie SamochódSportowy można potraktować jako Pojazd, ponieważ SamochódSportowy dziedziczy od Pojazdu cechy, a więc po części nim jest.

Możliwe także staje się inicjowanie SportowegoSamochod poprzez typ Pojazd co jest przydatne gdy np. mamy wiele klas, które dziedziczą od super klasy i chcemy te klasy przesyłać do funkcji. Wówczas wystarczy stworzyć obiekt klasy Pojazd i przypisywać do tego obiektu dowolny rodzaj klasy czyli np. SamochódSportowy czy SamochodTurysyczny itd.









* Dziedziczenie

Dziedziczenie jest to przekazywanie cech z klasy rodzica do klasy dziecka. Dziedziczenia używamy aby nie dopisywać poszczególnych cech do każdej klasy w przypadku gdy te cechy są wspólne dla danych klas, np. Pojazd ma cenę, model, markę itd, to samo ma samochód lecz cechu te są rozszerzone o np. ilość miejsc siedzących. Nie ma potrzeby przepisywania tych cech wystarczy użyć słowa kluczowego extends i z klasy np. Pojazd do klasy Samochód zostają „przekopiowane” wszystkie cechy z klasy Pojazd. Prezentacja na przykładzie:

