

Należy stworzyć zestaw klas, mogących utworzyć bazę danych pojazdów.

1. Zaimplementować klasę Pojazd, będącą klasą bazową dla różnych walut w programie. Klasa Pojazd, jest klasą **abstrakcyjną** i przechowuje następujące pola:

nazwa - użyć typu **string**, (np. „Opel”)
cena (cena pojazdu w PLN),

Pola dostępne są tylko dla klas potomnych.

W części publicznej klasa zawiera deklarację **wirtualnych metod** Wypisz(), Wczytaj(char*, int) oraz Zapisz(char*). Konstruktor/destruktor w zależności od potrzeb.

2. Jako przykład klasy odpowiadającej danemu pojazdowi zaimplementować klasę Osobowy, będącą klasą potomną klasy Pojazd, zawierającą dodatkowe pola danych danego typu pojazdów (liczba miejsc, masa) oraz odpowiednie metody. Zdefiniować konstruktor domyślny (wszystkie wartości domyślnie ustawione na 0, konstruktor ten powinien również wywoływać konstruktor klasy nadrzędnej), jak również konstruktor inicjalizujący wszystkie pola klasy Pojazd (wraz z nazwą i ceną pojazdu).

- Metoda Wypisz powinna wypisać na ekran informację "Samochod osobowy marki Opel, 5 miejsc, masa 1200 kg, cena 8000 PLN".
- Metoda Wczytaj(char* a, int b) powinna wczytać z pliku, którego nazwa podana jest jako jej parametr (a), z liniiki o numerze podanym jako drugi parametr (b), dane pojazdu oraz przypisać je odpowiednim składnikom klasy Osobowy. Przykład pliku: SamochodyOsobowe.txt:
Opel 5 1200 8000
Ford 4 800 10000
Volkswagen 2 1000 30000 , gdzie kolejno mamy nazwę, liczbę miejsc, masę oraz cenę.
- Metoda Zapisz powinna zapisać do pliku, którego nazwa podana jest jako jej parametr, wszystkie parametry pojazdu osobowego

3. Napisać program, testujący działanie klasy:

- Stworzyć obiekt o1 klasy Osobowy (używając konstruktora domyślnego). Wypisać (używając metody Wypisz).
- Stworzyć wskaźnik (wraz z obiektem) o2 na klasę Osobowy (ustawić jakieś rozsądne wartości parametrów). Wypisać.
- Wczytać do obiektu o1 dane z drugiej liniiki z pliku którego nazwa podana jest jako **pierwszy parametr wywołania programu**. Wypisać. Przykładowy plik należy stworzyć samodzielnie.
- Zapisać dane obiektu o2 do pliku podanego jako **drugi parametr wywołania programu**.

4. W funkcji głównej należy stworzyć dwuwymiarową tablicę pojazdów osobowych domyślnych (tablica powinna mieć wymiary podane jako trzeci i czwarty parametr wywołania programu). Następnie należy zmienić cenę elementu [0,0] na 8 i wypisać ceny wszystkich pojazdów na ekran, uwzględniając odpowiednie formatowanie - powinno skutkować wypisaniem macierzy złożonej głównie z zer, np. dla tablicy 2 x 3 powinniśmy uzyskać na ekranie:

```
8 0 0
0 0 0
```

Dodatkowe: Na kolokwium mają sporą szansę pojawić się również **tablice dwuwymiarowe jako składnik klasy** (w celu przeciwiczenia można w tym zadaniu dodać dwuwymiarową – 3 x 3 – dynamicznie alokowaną tablicę tab wypełnioną 7-kami, do klasy Pojazd) oraz **losowanie** (liczb całkowitych i zmiennoprzecinkowych, by przeciwiczyć można w konstruktorze domyślnym losowanie z rozkładu jednorodnego z przedziału [3,15]).

5. Należy stworzyć Makefile, umożliwiający kompilację napisanego programu, definiując zmienną CXX definiującą kompilator (g++) oraz CXXFLAGS definiującą flagę (-Wall).

Cały program powinien być napisany zgodnie z obowiązującą dobrą praktyką programowania: klasy dzielimy na pliki *.cpp i *.h, powinny być otoczone strukturami ifndef, odpowiednie **wcięcia** powinny być zachowane, (itp). Program musi się kompilować bez żadnych błędów i ostrzeżeń (za niekompilujący się program przewidziana jest kara: tylko 40% zdobytych punktów – maksymalnie można zdobyć 8 p).