

## Języki Programowania, Zadanie 3

Ideą programu jest stworzenie „bazy danych” w C++ dla sklepu z używanymi skuterami. Sklep ma miejsce na 40 skuterów (można zadeklarować tablicę statyczną na [40] skuterów), a każdy skuter ma: kolor, wiek oraz cenę. Można nowe skutery dodawać, oraz wypisywać wszystko, co sklep ma na składzie.

1) Stwórz klasę Skuter zawierającą trzy składniki prywatne: kolor (std::string), wiek (int) oraz cenę (double). Klasa powinna być podzielona na dwa oddzielne pliki: skuter.h i skuter.cpp. W trzecim pliku powinna znajdować się funkcja główna.

Klasa powinna mieć dwie funkcje składowe (tzw. metody):

- `SetCena(double cost)` - zapisująca daną cenę do odpowiedniego składnika klasy
- `Wypisz()` - wypisująca na ekran informacje o danym skuterze (po spacjach: kolor, wiek oraz cenę)

2) Należy również zaimplementować konstruktor domyślny, główny, kopiujący oraz destruktor. Destructork należy opatrzyć komentarzem - wypisywaniem na domyślne wyjście „Został użyty destruktor”.

3) W programie głównym, należy:

- stworzyć obiekty napisanej klasy używając konstruktora domyślnego, głównego i kopiującego, wypisać je na ekran (1,5 p.)

**1: (zielony, 6 letni, 3000.5 zł) 2: (niebieski, 2 letni, 4000 zł) 3: (niebieski, 2 letni, 4000 zł)**

- stworzyć wskaźniki do obiektów (operator new) napisanej klasy używając konstruktora domyślnego, głównego i kopiującego, wypisać je na ekran (0,5 p.)
- użyć operatora delete do usunięcia wskaźnika na obiekty klasy Skuter (0.5 p.)

4) Należy przygotować „bazę danych” z możliwością dodawania, zmiany ceny oraz wypisywania skuterów używając w tym celu tablicy na 40 skuterów (ich maksymalna ilość). W głównym programie (funkcji main) wymagana jest instrukcja typu switch - case, pytająca w pętli o podanie liczby od 1-3 lub 0. **Należy również stworzyć zmienną odpowiadającą za zliczanie ilości skuterów w sklepie (licznik).**

- Wpisanie liczby innej niż dozwolone - program zakomunikuje, iż została wprowadzona błędna wartość. (0.5p)
- Wpisanie 0 - powinno przerwać pętlę oraz spowodować opuszczenie programu. (0.5p)
- Wpisanie 1 - wypisanie wszystkich zapisanych skuterów na ekran. (0.5p)
- Wpisanie 2 - dodanie nowego skutera do „bazy danych”. (0.5 p) (pola podawane z klawiatury!)
- Wpisanie 3 - zmiana ceny skutera o konkretnym indeksie z „bazy danych” (0.5p)

Przydatne nagłówki (biblioteki):

```
#include <iostream>
```

```
g++ -Wall skuter.cpp program.cpp -o program
```

**Zadania dodatkowe:**

- Wpisanie 4 - usuwanie skutera o konkretnym indeksie z „bazy danych” (należy pamiętać o przesuwaniu kolejnych skuterów w tablicy).
- Należy zmienić string na char\*. Tablica znaków przeznaczona na nazwę koloru w klasie powinna być alokowana dynamicznie (operator **new**). || #include <cstring>
- Dynamiczna „baza danych”: użycie klasy vector lub queue.