
Zadania nr 17

Zad 17.1 Przepisz program wykorzystujący klasę **Tpralka**. Skompiluj kod. Następnie wykonaj polecenia:

- dodaj 2 pola z atrybutami klasy **Tpralka**
 - ✚ ładowność pralki w litrach – jako liczbę typu **double**
 - ✚ wysokość pralki w cm – jako liczbę typu **int**
- stwórz 3 obiekty klasy **Tpralka**
- w pierwszym stworzonym obiekcie ustaw `nr_programu` na 1 posługując się nazwą obiektu
- w drugim stworzonym obiekcie ustaw wartość ładowności pralki na 5.25 posługując się wskaźnikiem
- w trzecim stworzonym obiekcie ustaw nazwę pralki na „Frania” posługując się referencją
- w przypadku wcześniejszego zakończenia zadania spróbuj poustawiać inne parametry pralek wg własnego uznania przy pomocy dowolnie wybranej metody

Zad 17.2 Utwórz nowy projekt. Napisz kod, który będzie reprezentował Twoją wybraną klasę, którą wymyśliłeś na poprzednich zajęciach. W ramach tego zadania wykonaj następujące „kropki”:





- niech Twoja klasa ma chociaż 3 atrybuty, najlepiej niech będą różnych typów
- niech Twoja klasa ma 2 dowolne metody. Mogą one być typu **void**, jeśli nie mają zwracać żadnej wartości. Nie muszą mieć żadnych argumentów (chyba, że autor twierdzi inaczej). Jako implementację metody dla uproszczenia wyświetl na ekranie konsoli co ta metoda powinna robić. Np. jeśli masz klasę **Samochod** a w nim metodę **jedz()** to przykładowa implementacja może wyglądać następująco:

```
void jedz()
{
    cout << "Jade!" << endl;
}
```

- stwórz 2 obiekty Twojej klasy
- niech klasa posiada metody **wczytaj()** oraz **wypisz()**, gdzie za pomocą `cout` i `cin` wczytasz wartości atrybutów oraz wypiszesz je na ekranie

Implementację klasy wykonaj w jednym pliku źródłowym `main.cpp`.

Zad 17.3 Utwórz nowy projekt. Napisz kod, który będzie reprezentował klasę Kalkulator:

- klasa ta nie będzie posiadała żadnych atrybutów, będzie miała tylko publiczne metody
- zaimplementuj metody:
 -  dodawanie
 -  odejmowanie
 -  mnożenie
 -  dzielenie
- metody, które zaimplementujesz muszą mieć zdefiniowane parametry domyślne (ustawione na wartość 1 lub 0, zgodnie z matematycznym sensem).

Zad 17.4 Utwórz nowy projekt. Napisz program, który posłuży do sterowania samochodem i motocyklem:

- Klasa **Samochod** powinna mieć takie atrybuty jak
 - mocSilnika – liczba całkowita nieujemna. Domyślnie jest to 125KM.
 - pojemnoscSilnika – liczba całkowita nieujemna. Domyślnie jest to 1500 cm³.
 - silnik – wartość true or false informująca czy auto ma uruchomiony silnik. Domyślnie jest to wartość false
 - jazda – wartość true or false informująca czy auto jedzie. Domyślnie jest to wartość false.
- Klasa **Samochod** powinna mieć takie metody jak:
 - Konstruktor inicjujący domyślnie wartości;
 - void hamuj() – metoda wyświetla informację na ekranie, że auto hamuje tylko wtedy, gdy wykryje uruchomiony silnik w samochodzie. W przeciwnym wypadku wyświetla informację, że samochód nie jedzie;
 - void uruchomSilnik() – wyświetla informację na ekranie, że uruchomiono silnik oraz uruchamia silnik (zmienia wartość pola silnik na true);
 - void jedz() – jeśli jest włączony silnik to wypisuje na ekranie informację o tym, że auto się porusza oraz umożliwia jazdę (zmienia wartość pola jazda na true);
 - void wyświetl() – wyświetla informacje o danych samochodu (jego moc oraz pojemność silnika).
- Klasa **Motocykl** powinna mieć takie atrybuty jak
 - mocSilnika – liczba całkowita nieujemna. Domyślnie jest to 20KM.
 - pojemnoscSilnika – liczba całkowita nieujemna. Domyślnie jest to 700 cm³.
- Klasa **Motocykl** powinna mieć takie metody jak:
 - Konstruktor inicjujący domyślnie wartości;
 - void wyświetl() – wyświetla informacje o danych samochodu (jego moc oraz pojemność silnika).
- Po uruchomieniu programu wyświetl użytkownikowi menu główne, w którym będzie on mógł uruchomić silnik samochodu, jechać autem, wyświetlić informacje o samochodzie, wyświetlić informacje o motocyklu lub wyjść z programu.

- Utwórz po jednym egzemplarzu obiektu. Na podstawie wyboru użytkownika niech zostaną wywoływane odpowiednie metody klas **Samochod** oraz **Motocykl**. Możesz wykorzystać instrukcję wyboru `switch`.
- Całość instrukcji `switch` (lub serii `if else`) umieść w jednej pętli nieskończonej `while`, aby program działał tak długo, jak tego chce użytkownik.
- Spróbuj samemu rozwiązać problem wyjścia z nieskończonej pętli!