

## LABORATORIUM 2.3

### INTERAKCJE OBIEKTOWE

#### Zadanie podstawowe [2,0 pkt.]

---

■ Napisz klasę **Flota**, która będzie agregatem dla maksymalnie 20 instancji klasy **SamochodCiezarowy** z pliku „klasy.h”. Klasa ta powinna mieć tylko konstruktor domyślny ustawiający wszystkie wskaźniki na nullptr i aktualną liczbę pojazdów na 0.

*Wskazówki: Do tego celu oczywiście potrzebna jest tablica wskaźników i licznik przypinanych instancji. Licznik powinien być polem instancyjnym, gdyż każda instancja Floty może mieć inną liczbę przypisanych pojazdów. Wskaźnik uznawana jest za wolny jeśli przypisany jest do niego nullptr. Nie pisz żadnych funkcjonalności, które nie są konieczne do działania klasy.*

■ Dla klasy **Flota** zdefiniuj metody:

- dodającą wskazaną instancję klasy **SamochodCiezarowy** do **Floty**, tak aby wpisać ją na pierwszej wolnej pozycji. Zadbaj, aby nie dało się do przypisać dwa razy tej samej instancji (na różnych pozycjach).
- usuwającą wskazaną instancję klasy **SamochodCiezarowy** z **Floty** (nie podawaj numeru wskazania w tablicy tylko używaj porównania tożsamościowego).

*Wskazówki: Pamiętaj, aby przypisać instancję tylko raz – jeśli robisz to w pętli, to przerwij pętlę po udanym przypisaniu. Na początku sprawdź czy podana instancja nie jest już przypisana. Kolejność wskazań nie ma znaczenia, oznacza to, że przy usuwaniu wskazania ze środka tablicy, można zastąpić je ostatnim wskazaniem, a te z kolei ustawić na nullptr. Takie rozwiązanie sprawia, że nie tylko nie ma „dziur” we wskazaniach, ale też, że licznik wskazań, zawsze indeksuje pierwsze wolne miejsce w tablicy. Nie zapomnij o dekrementacji licznika przy usuwaniu instancji i inkrementacji przy dodawaniu.*

■ Dla klasy **Flota** zdefiniuj metodę zwracającą sumaryczną pojemność wszystkich samochodów do niej przypisanych.

*Wskazówki: Jeśli nie ma żadnego przypisanego samochodu to zwróć 0.*

■ Dla klasy **Flota** zdefiniuj operator << wypisujący wszystkie przypisane do **Floty** samochody podając ich numer rejestracyjny i pojemność.

#### Ocenianie:

Projekt agregatu Flota:	0,4 pkt.
Metoda dodająca:	0,5 pkt.
Metoda usuwająca:	0,5 pkt.
Metoda obliczająca pojemność:	0,2 pkt.
Operator <<:	0,2 pkt.
Ogólna jakość kodu:	0,2 pkt.

#### Zadanie ambitne [2,0 pkt.]:

---

▲ Zmodyfikuj klasy **SamochodOsobowy** i **Kierowca** z pliku „klasy.h” tak, aby możliwe było utworzenie między nimi asocjacji obustronnej.

*Wskazówki: W obu klasach dodaj właściwe wskaźniki.*

▲ Dodaj do tych klas właściwe metody, które dokonują asocjacji obustronnej przy czym przypisanie jest możliwe tylko jeśli **Kierowca** ma prawo jazdy kategorii B. Jeśli inny kierowca był już przypisany, to zastąp stare przypisanie nowym.

*Wskazówki: Jeśli dojdzie do próby przypisania Kierowcy z niewłaściwą kategorią powinien zostać zgłoszony wyjątek. Zauważ, że asocjacja jest tu obustronna, więc jeśli miałyby nastąpić jej zerwanie, to musi to nastąpić w obu powiązanych instancjach. Warto najpierw zrealizować następne polecenie i skorzystać metody zrywającej asocjację, aby dwa razy nie siłować się z tym samym problemem.*

▲ Dodaj do tych klas właściwe metody zrywające asocjację.

*Wskazówki: Zauważ że do zerwania asocjacji nie potrzeba podawać z kim się ją zrywa. Ważne jest jednak aby bez względu na to która strona zainicjuje zerwanie asocjacji, nastąpiło ono w obu. Kluczowe jest tu napisanie takiej logiki, aby nie nastąpiło nieskończone odwołanie cykliczne.*

■ Zdefiniuj dla obu klas destruktory, które dokonają automatycznego zerwania asocjacji w chwili zniszczenia instancji.

*Wskazówki: Zauważ, że to tylko wywołanie gotowej już metody. To zerwanie jest ważne, gdyż jeśli instancja przestanie istnieć, to instancja z nią powiązana nie będzie o tym wiedzieć i będzie miała wskaźnik ustawiony na nieistniejący już obiekt – to bardzo niebezpieczne.*

■ Podziel program tak, aby klasy **SamochodOsobowy** i **Kierowca** znalazły się w oddzielnych jednostkach translacji.

*Wskazówki: Zauważ, że przy asocjacji obustronnej podział na pliki trochę się komplikuje, gdyż każda z tych klas wymaga znajomości drugiej. Na wykładzie (slajd 11) pokazano jak należy rozwiązać ten problem.*

#### Ocenianie:

Asocjacja obustronna:	0,6 pkt.
Zerwanie asocjacji:	0,6 pkt.
Destruktor:	0,2 pkt.
Podział na pliki:	0,4 pkt.
Ogólna jakość kodu:	0,2 pkt.