

## Programowanie w języku C++

### Laboratorium nr 5 i 6

*składniki statyczne / definiowanie funkcji poza klasą / statyczne funkcje składowe /  
konstruktor, destruktor / lokalne zmienne automatyczne / funkcje zaprzyjaźnione /  
tablice obiektów / konstruktor kopiujący / dziedziczenie*

#### Zadanie nr 1

Zaprojektuj klasę **Towar** do przechowywania informacji o danych towarach.

Definicje funkcji składowych umieszczaj poza klasą. W klasie pozostaw tylko prototyp funkcji składowych;

Klasa powinna przechowywać następujące informacje:

- prywatne pola **nazwa** i **cenaNetto** towaru,
- **konstruktor** wypełniający pola klasy(nazwę i cenę) oraz wypisujący komunikat o tworzeniu towaru
- **destruktor** wypisujący komunikat o usuwaniu towaru
- funkcję **drukuj** (typu const) wyświetlającą dane towaru
- prywatną statyczną składową **licznik** do przechowywania liczby towarów
- zainicjalizuj(np. liczbą 0) składową przechowującą liczbę towarów
- zmodyfikuj klasę tak, aby przy tworzeniu i usuwaniu towaru zmieniała się wartość zmiennej przechowującej liczbę towarów
- statyczną funkcję **LiczbaTowarow** zwracającą liczbę towarów
- utwórz akcesory dla pól składowych **nazwa** i **cenaNetto**

W programie głównym

- Wypisz liczbę towarów (na początku powinno być zero);
- Utwórz obiekt typu Towar (np. zeszyt, 5.50 zł)
- Wypisz dane zeszytu(korzystając z funkcji składowej)
- Wypisz ile jest towarów(skorzystaj z funkcji statycznej)
- Utwórz obiekt lokalny automatyczny typu towar(np. długopis, 8 zł)
- Wypisz ile jest towarów
- Utwórz jeszcze jeden towar za pomocą operatora new i wypisz jego dane

Przetestuj działanie programu najlepiej z linii komend. Zaobserwuj kiedy wywoływany jest konstruktor, a kiedy destruktor.

**Zadanie nr 2**

- Zdefiniuj klasę `Osoba` zawierającą dwa pola: imię oraz nazwisko.
- Utwórz konstruktor dla klasy `Osoba` zarówno bezargumentowy, jak i konstruktor wywoływany z dwoma argumentami pozwalającymi zainicjalizować pola: imię i nazwisko.
- Przygotuj metodę `Prezentuj()` wyświetlającą zawartość pól klasy `osoba`. Zadbaj o odpowiednie formatowanie danych tak, by imię było oddzielone od nazwiska spacją.
- Przygotuj funkcję zaprzyjaźnioną `ZmienNazwisko()` pozwalającą na zmianę nazwiska już utworzonego obiektu.
- Zdefiniuj kilka obiektów klasy `osoba`. Sprawdź działanie metod `Prezentuj()` oraz `ZmienNazwisko()`.

**Zadanie nr 3**

- Zdefiniuj klasę `Pracownik` dziedziczącą po klasie `Osoba` (z zadanie 2), zawierającą dodatkowe pola `stanowisko` oraz `wynagrodzenie`.
- Zdefiniuj konstruktor dla klasy `Pracownik`. Przygotuj również konstruktor kopiujący umożliwiający utworzenie pracownika na podstawie już istniejących obiektów klasy `Osoba`.
- Zdefiniuj metodę `PrezentujPracownika()` prezentującą wszystkie dane pracownika (imię, nazwisko, stanowisko, wynagrodzenie). Wykorzystaj w nowozdefiniowanej metodzie `PrezentujPracownika()` metodę `Prezentuj()` z klasy bazowej.
- Utwórz kilka obiektów klasy `pracownik`.
- Sprawdź działanie metody `PrezentujPracownika()` dla obiektów klasy `Pracownik`.

**Zadanie nr 4**

- W klasie `Pracownik` (z zadania 3) zmień modyfikatory dostępu dla pól `stanowisko` oraz `wynagrodzenie` na `private`.
- Zdefiniuj gettery i settery umożliwiające dostęp i modyfikowanie pól z kwantyfikatorami `private`.
- Sprawdź działanie powyższych akcesorów.

**Zadanie nr 5**

- W programie z zadanie 3 zdefiniuj nową klasę Student dziedziczącą po klasie Osoba.
- Nowym polem w klasie Student będzie pole o nazwie nrIndeksu.
- Przygotuj odpowiednie konstruktory, domyślny oraz wieloargumentowy (w tym pozwalający na tworzenie obiektów klasy Student na podstawie już utworzonych obiektów klasy Osoba).
- Do każdej z utworzonych klas dodaj pole statyczne pozwalające zliczać ilość obiektów danej klasy. Nazwij je adekwatnie do nazwy klasy tj. licznikOsob, licznikPracownikow, licznikStudentow.
- Do powyższych klas dodaj również pole przechowujące numer obiekty (obiekty numerowane są wg hierarchii ich powstawania). Nazwij je adekwatnie do nazwy klasy tj. nrOsoby, nrPracownika, nrStudenta.
- Przetestuj działanie pola zliczającego oraz pola numerującego obiekty klasy Pracownik oraz Student.

**Zadanie nr 6**

- Kontynuując program z zadania 4 utwórz klasę Grupa zawierającą pole nazwa oraz tablicę obiektów klasy Student o nazwie skladGrupy[]. Wyznacz stałą definiującą maksymalną liczbę kursantów w grupie. Stała ta będzie jednocześnie rozmiarem tablicy.
- Przygotuj definicje metody DodajDoGrupy() pozwalającej na wprowadzenie kolejnego obiektu klasy Student do tablicy skladGrupy[].
- Przygotuj również metodę UsunZGrupy() usuwającą obiekty klasy Student z tablicy skladGrupy[].

**Zadanie nr 7 [ZADANIE DOMOWE]**

- Zaplanuj nowy zestaw trzech klas zachowując podobne zależności tj. klasa bazowa, z której dziedziczą dwie kolejne klasy.
- Zdefiniuj powyższe klasy. Wykorzystaj następujące elementy:
  - konstruktory argumentowe i bezargumentowe
  - konstruktor kopiujący
  - funkcje zaprzyjaźnione
  - gettery i settery
  - licznik obiektów
  - numerowanie obiektów
  - tablicę obiektów
- Przygotuj odpowiednie przykłady użycia powyższych elementów.