

## Zestaw 5 – struktury, unie, inne typy, klasy

- Zad. 1. Zdefiniuj strukturę trojkat przechowującą długości boków trójkąta. Napisz funkcję, która otrzymuje jako argument zmienną typu struct trojkat, i zwraca jako wartość obwód trójkąta przekazanego w argumencie.
- Zad. 2. Zdefiniuj strukturę lista posiadającą dwa pola: jedno typu int oraz drugie będące wskaźnikiem do definiowanego typu. Zaprezentuj przykład wykorzystania
- Zad. 3. Zdefiniuj strukturę figura przechowującą wymiary figur geometrycznych niezbędne do obliczenia pola. Struktura powinna mieć możliwość przechowywania wymiarów takich figur, jak: trójkąt, prostokąt, równoległobok i trapez. Rodzaj przechowywanej figury powinien być zakodowany w wartości pola fig typu int. Definiując strukturę, staraj się zużyć jak najmniej pamięci. Napisz funkcję pole, która dostaje jako argument zmienną f typu struct figura i zwraca jako wartość pole figury której wymiary przechowuje zmienna f.
- Zad. 4. Zdefiniuj strukturę dane osobowe zawierającą pola: imie, nazwisko, plec, stan\_cywilny. W zależności od płci pole stan\_cywilny powinno mieć jedną z dwóch wartości wolny lub zony dla mężczyzn i wolna lub mężatka dla kobiet. Napisz funkcję wczytaj o dwóch argumentach: tablicy tab o elementach typu stan\_cywilny i jej rozmiarze. Funkcja powinna wczytywać do komórek tablicy tab wartości podane na standardowym wejściu
- Zad. 5. Napisz klasę **liczba** służącą do przechowywania liczb całkowitych. Klasa powinna udostępniać następujące metody publiczne:
- wczytaj** wczytującą wartość liczby ze standardowego wejścia,
  - wypisz** wypisującą wartość liczby na standardowe wyjście,
  - nadaj\_w** nadającą przechowywanej liczbie wartość podana w argumencie metody,
  - wartosc** zwracającą wartość przechowywanej liczby,
  - abs** zwracającą wartość bezwzględną przechowywanej liczby.
- Napisz klasę liczba w taki sposób, żeby dostęp do zawartych w niej danych był możliwy tylko za pośrednictwem metod tej klasy.
- Zad. 6. Napisz klasę punkt służącą do przechowywania współrzędnych punktu w dwuwymiarowym kartezjańskim układzie współrzędnych. Napisz metody do wczytywania i wypisywania współrzędnych. Zadeklaruj wszystkie pola klasy jako prywatne.
- Zad. 7. Napisz klasę punkt3 służącą do przechowywania współrzędnych punktu w trójwymiarowym kartezjańskim układzie współrzędnych. Napisz metody do wczytywania i wypisywania współrzędnych. Zadeklaruj wszystkie pola klasy jako prywatne.
- Zad. 8. Napisz funkcję rzutuj, która otrzymuje jako argument obiekt typu punkt3 z zadania 7 i zwraca jako wartość obiekt typu punkt z zadania 6 będący prostym rzutem punktu otrzymanego w argumencie na płaszczyznę wyznaczoną przez dwie pierwsze współrzędne
- Zad. 9. Do klasy punkt z zadania 6 dopisz metodę rzutuj, która otrzymuje jako argument obiekt typu punkt3 z zadania 7 i przypisuje polom obiektu, na rzecz którego została wywołana, współrzędne prostokątnego rzutu punktu otrzymanego w argumencie na płaszczyznę wyznaczoną przez dwie pierwsze współrzędne
- Zad. 10. Napisz klasę **wskaznik** zawierającą jedno pole prywatne **wsk** typu **wskaznik** do zmiennej typu **int**. Klasa **wskaznik** powinna udostępniać następujące publiczne metody:
- utworz**, która dla otrzymanej w argumencie dodatniej liczby całkowitej n, rezerwuje pamięć dla n-elementowej tablicy o elementach typu **int** i zapisuje w polu **wsk** wskaźnik do nowo utworzonej tablicy,
  - zwroc** zwracającą jako wartość wskaźnik przechowywany w polu **wsk**,
  - zwolnij** zwalniającą obszar pamięci wskazywany przez pole **wsk** i nadającą temu polu wartość NULL,

## Zestaw 5 – struktury, unie, inne typy, klasy

- d. **kopiuuj**, która otrzymuje jako argument referencje ref do zmiennej typu wskaźnik i dokonuje przypisania **ref.wsk=wsk**.

Zad. 11. Do klasy **wskaznik** z zadania 10 dopisz:

- a. bezargumentowy konstruktor, przypisujący do pola **wsk** wartość NULL,

Zad. 12. Napisz klasę kolejka będącą implementacją klasycznej kolejki przechowującej liczby całkowite. Klasa kolejka powinna udostępniać następujące publiczne metody:

- a. bezargumentowy konstruktor tworzący pustą kolejkę,
- b. konstruktor kopiujący,
- c. destruktory zwalniający pamięć zaalokowaną przez obiekt,
- d. **pierwszy** zwracającą jako swoją wartość pierwszy element kolejki,
- e. **usun\_pierwszy** usuwającą pierwszy element kolejki,
- f. **dodaj\_na\_koniec**, dodającą na koniec kolejki liczbę całkowitą otrzymaną w argumencie,
- g. pusta zwracającą **true** jeżeli kolejka jest pusta i **false** w przeciwnym wypadku.

Zad. 13. Napisz klasę stos będącą implementacją klasycznego stosu przechowującego liczby całkowite. Klasa stos powinna udostępniać następujące publiczne metody:

- a. bezargumentowy konstruktor tworzący pusty stos,
- b. konstruktor kopiujący,
- c. destruktory zwalniający pamięć zaalokowaną przez obiekt,
- d. **z\_wierzchu** zwracającą go jako swoją wartość element z wierzchu stosu,
- e. **usun\_z\_wierzchu** usuwającą element znajdujący się na wierzchu stosu,
- f. **poloz\_na\_stos** kładący na wierzchu stosu liczbę całkowitą otrzymaną w argumencie,
- g. pusty zwracającą **true** jeżeli stos jest pusty i **false** w przeciwnym wypadku.