

Wskaźniki ogólnie

<code>int szczeniwa = 7;</code>	Deklaracja zmiennej "szczeniwa" typu <code>int</code> (całkowitego)
<code>int *wskaznik;</code>	Deklaracja wskaźnika na typ <code>int</code> (całkowitego)
<code>wskaznik = &szczeniwa</code>	Przypisanie zmiennej "wskaznik" adresu zmiennej "szczeniwa"
<code>int **wskaznik;</code>	Deklaracja wskaźnika na wskaźnik typu <code>int</code> (całkowitego)
<code>void *wskaznik;</code>	Deklaracja wskaźnika który może wskazywać na każdy typ.

Wskaźniki Funkcje

<code>T* mojaFunkcja();</code>	Deklaracja funkcji "mojaFunkcja" zwraca wskaźnik na typ <code>T</code>
<code>T (*f_wskaznik)() = &mojaFunkcja();</code>	Ustawienie wskaźnika "f_wskaznik" na nazwę funkcji o nazwie "mojaFunkcja"
<code>void() (void*);</code>	Deklaracja wskaźnika na funkcję zwracającą wskaźnik na każdy typ danych i przyjmującą wskaźnik na każdy typ

Wskaźniki Tablice

<code>int *wskaznikTablica = mojaTablica;</code>	Wskaźnik na tablice "mojaTablica", na jej pierwszy element
<code>int *wskaznikTablica = &mojaTablica[0];</code>	Wskaźnik na tablice "mojaTablica", na jej pierwszy element
<code>int (*wskaznikTablica)[5];</code>	Wskaźnik na tablice "mojaTablica" 5 elementów
<code>int *mojaTablica[5];</code>	Tablica "mojaTablica" pięciu wskaźników typu <code>int</code>
<code>const char* napis = "Hello";</code>	Deklaracja wskaźnika na pierwszy znak łańcucha znaków typu "char"

Przykłady dla `int tablica[5] = { 1, 2, 3, 4, 5 };`

Referencje

<code>int szczeniwa = 7;</code>	Deklaracja zmiennej "szczeniwa" typu <code>int</code> (całkowitego)
<code>int &referencja = szczeniwa;</code>	Przypisanie zmiennej "referencja", zmiennej "szczeniwa"
<code>int &referencja;</code>	Błąd ! Puste referencje nie są możliwe.

Referencje vs Wskaźnik (Referencje)

```
int szczeniwa = 7;
int &referencja = szczeniwa;
cout << szczeniwa; // Wypisze 7
referencja = 10;
cout << szczeniwa; // Wypisze 10
cout << referencja; // Wypisze 10
```

Zmiana zmiennej "referencja" odpowiada zmianie zmiennej "szczeniwa"



Wskaźniki i struktury

<code>str mojaStruktura;</code>	Deklaracja struktury "mojaStruktura" o typie <code>str</code>
<code>str *mojaStruktura;</code>	Deklaracja wskaźnika <code>str</code> na strukturę "mojaStruktura"
<code>mojaStruktura.a = 10;</code>	Dostęp do pola "a" struktury "mojaStruktura"
<code>mojaStruktura->a = 10;</code>	Dostęp do pola "a" przez wskaźnik na strukturę "mojaStruktura"

```
struct str {
int a;
}
```

Wskaźniki wyświetlanie

<code>cout << szczesliwa;</code>	Wyświetlenie wartości "szczesliwa"
<code>cout << &szczesliwa;</code>	Wyświetlenie adresu zmiennej "szczesliwa"
<code>cout << wskaznik;</code>	Wyświetlenie adresu zmiennej "szczesliwa"
<code>cout << *wskaznik;</code>	Wyświetlenie wartości "szczesliwa"
<code>cout << &wskaznik;</code>	Wyświetlenie adresu wskaźnika "wskaznik"

Wskaźniki Funkcje - wywołanie

<code>mojaFunkcja ();</code>	Wołanie funkcji <code>mojaFunkcja ()</code> za pomocą nazwy (klasycznie)
<code>*f_wskaznik) ();</code>	Wołanie funkcji przez wskaźnik <code>f_wskaznik</code>

Wskaźniki i modyfikator const

<code>const int* wskaznik;</code>	Deklaracja wskaźnika "wskaznik" który może wskazywać tylko na stałą typu <code>int</code>
<code>int * const wskaznik;</code>	Deklaracja stałego wskaźnika na typ <code>int</code>

Wskaźniki - const właściwości

Składnia	Charakter	Ponowne przypisanie ?	Zmiana wartości na którą pokazuje ?
<code>const int *wskaznik;</code>	Wskaźnik na <code>const</code>	TAK	NIE
<code>int const *wskaznik;</code>	Wskaźnik na <code>const</code>	TAK	NIE
<code>int* const wskaznik;</code>	Wskaźnik <code>const</code>	NIE	TAK
<code>const int *const wskaznik;</code>	Wskaźnik <code>const</code> na <code>const</code>	NIE	NIE
<code>const int const *const wskaznik;</code>	Wskaźnik <code>const</code> na <code>const</code>	NIE	NIE

Referencje vs Wskaźnik (Wskaźnik)

```
int szczesliwa = 7;
int *wskaznik = &szczesliwa;
cout << szczesliwa; // Wypisze 7
*wskaznik = 10;
cout << szczesliwa; // Wypisze 10
cout << *wskaznik; // Wypisze 10
```

Zmiana wskaźnika "wskaznik" odpowiada zmianie zmiennej "szczesliwa"

