

Metody przekazywania parametrów

1. Przez wartość

Przykład:

```
void foo(int x)
{
    x = 34;
}

int main()
{
    int x = -1;
    foo(x);
    cout << x << endl;
}
```

Do funkcji foo przekazywana jest kopia zmiennej x, operacje na zmiennej wewnątrz funkcji nie mają wpływu na zmienną poza funkcją, ponieważ zmieniana zostaje lokalna kopia. Dlatego powyższy kod wypisze na wyjście -1.

2. Przez wskaźnik

Przykład:

```
void foo(int* x)
{
    *x = 34;
}
```

```
int main()
{
    int x = -1;
    foo(&x);
    cout << x << endl;
}
```

```
int main()
{
    int x = -1;
    int* p = &x;
    foo(p);
    cout << x << endl;
}
```

W tym przypadku do funkcji foo zostaje przekazany adres zmiennej, a dokładnie w funkcji jest tworzona lokalna kopia wskaźnika, wskazująca na podaną przez nas zmienną x. Powyższy kod wypisze na wyjście 34, ponieważ odwołujemy się po przez adres do zmiennej poza funkcją.

3. Przez referencje

Przykład:

```
void foo(int& x)
{
    x = 34;
}

int main()
{
    int x = -1;
    foo(x);
    cout << x << endl;
}
```

Ten program wypisze na wyjście 34. Referencja działa tak jak wskaźnik ale jest bezpieczniejsza i wygodniejsza dla programisty. Raz przypisanej referencji nie można zmienić więc mamy pewność, że cały czas operujemy na tej samej zmiennej, co zwiększa czytelność kodu.

Wskaźniki

Wskaźnik to rodzaj zmiennej, w której przechowywany jest adres pamięci komputera. Może przechowywać adres zarówno do zmiennych jak i klas oraz funkcji.

symbol	znaczenie	użycie
*	weź wartość x	*x
*	deklaracja wskaźnika do wartości	int* x;
&	weź adres	&x

Wiszący wskaźnik (dangling pointer)- wskaźnik który nie został zainicjalizowany wskazuje na losowe miejsce w pamięci, próba dostępu do wskazanego obiektu kończy się niezdefiniowanym zachowaniem lub zakończeniem programu, dlatego do „pustych” wskaźników należy przypisywać nullptr.

Arytmetyka wskaźników – na wskaźnikach możemy wykonywać operacje dodawania i odejmowania. Dodanie do wskaźnika pewnej liczby spowoduje przesunięcie się o wielokrotność rozmiaru typu zmiennej.

Tablice a wskaźniki – tablic i wskaźników można używać zamiennie

Dynamiczna alokacja pamięci – dzięki wskaźnikom mamy dostęp do mechanizmu dynamicznej alokacji pamięci. Tworzymy wskaźnik a następnie za pomocą operatora new rezerwujemy odpowiednią ilość pamięci np.: int* tab = new int[5], zawsze należy pamiętać o zwalnianiu pamięci za pomocą operatora delete (w przypadku tablic delete[])

Źródła:

<https://pl.wikibooks.org/wiki/C%2B%2B>