

Temat: Wskaźniki, referencje i funkcje

Definicja wskaźnika jest podobna do definicji zmiennej:

```
int *wskaznik;
```

Jest to wskaźnik o nazwie `wskaznik`, który wskazuje na adresy komórek w których są zapisane zmienne typu `int`. Różnica polega na tym, że przy nazwie wskaźnika jest gwiazdka `*`. Ona odróżnia definicję zmiennej od nazwy wskaźnika. Zdefiniujmy sobie teraz zmienną typu `int` o nazwie `numer`:

```
int numer;
```

Ta zmienna ma zarezerwowany w pamięci pewien obszar, posiadający unikalny adres. Na razie nie znamy tego adresu, ale dzięki wskaźnikowi już niedługo go poznamy ;-)
Ze wstępu do tej lekcji już wiesz, że wskaźniki przechowują adresy, czyli do wskaźnika należy przypisać adres zmiennej `numer`. Robimy to tak:

```
wskaznik=&numer; // ustawiamy wskaźnik na komórkę ze zmienną numer
```

W tym miejscu kończy się notatka do zeszytu!!!

Za wskaźnik o nazwie `wskaznik` podstaw ADRES zmiennej o nazwie `numer`. Jeżeli się dobrze wpatrzymy to znak `&` będzie nam przypominał literę `a` jak adres ;-)
Teraz nasz wskaźnik wskazuje na komórkę pamięci, w której zapisana jest zmienna `numer`. Wypiszmy ten adres:

```
int numer;  
int *wskaznik;
```

```
wskaznik=&numer; // ustawiamy wskaźnik na komórkę ze zmienną numer  
cout<<"Adres zmiennej numer to "<<wskaznik; // wypisujemy adres komórki  
getch();
```

Spróbuj samodzielnie zdefiniować wskaźnik, zmienną i wypisać na ekranie adres tej zmiennej.

Tak właśnie wygląda adres komórki pamięci. I teraz w Twojej głowie pojawia się myśl: no super, znam adres komórki pamięci, w której jest zapisana zmienna `numer`...tylko po co mi to wiedzieć... Jeżeli znamy adres komórki, to możemy wykonywać operacje na tej zmiennej. Aby dowiedzieć się ile wynosi zmienna, zapisana pod danym adresem wystarczy napisać:

```
numer=5;  
cout<<"Wartosc zmiennej wynosi "<<*wskaznik; // przed nazwą wskaźnika dajemy gwiazdkę *  
getch();
```

Aby poznać wartość zmiennej kryjącej się pod adresem na który wskazuje wskaźnik, należy użyć gwiazdki `*`.

```

int numer;
int *wskaznik;

cout<<"Podaj wartosc zmiennej numer: ";
cin>>numer;

wskaznik=&numer; // ustawiamy wskaźnik na komórkę ze zmienną numer
cout<<"Adres zmiennej numer to "<<wskaznik<<endl; // wypisujemy adres komórki
cout<<"Wartosc zmiennej wynosi "<<*wskaznik; // przed nazwą wskaźnika dajemy gwiazdkę.
getch();

```

Spróbuj samodzielnie zdefiniować wskaźnik, zmienną, przypisać jej wartość i wypisać na ekranie adres zmiennej oraz jej wartość.

Na pewno w dalszym ciągu czujesz, że wskaźniki to bardziej zabawka niż przydatne narzędzie ;-). Masz do tego prawo. A tym czasem zbadamy co się stanie ze zmienną, jeżeli do wyrażenia `*wskaznik` dodamy 1.

```

int numer;
int *wskaznik;

cout<<"Podaj wartosc zmiennej numer: ";
cin>>numer;

wskaznik=&numer; // ustawiamy wskaźnik na komórkę ze zmienną numer

cout<<"Adres zmiennej numer to "<<wskaznik<<endl; // wypisujemy adres komórki
cout<<"Wartosc zmiennej wynosi "<<*wskaznik; // przed nazwą wskaźnika dajemy gwiazdkę.
cout<<"Teraz dodamy jeden do wyrażenia *wskaznik :"<<endl<<endl;

(*wskaznik)++; // całość musi być w nawiasie

cout<<"Adres zmiennej numer to "<<wskaznik<<endl; // wypisujemy adres komórki
cout<<"Wartosc zmiennej wynosi "<<*wskaznik; // przed nazwą wskaźnika dajemy gwiazdkę.
cout<<"Wartosc zmiennej wynosi "<<numer; // wypisujemy zmienną numer.

getch();

```

Adres zmiennej numer nie zmienił się, ale jej wartość tak. Dlaczego? Ponieważ wskaźnik wskazuje na komórkę pamięci, a my zwiększyliśmy o jeden wartość zmiennej na której adres wskazuje wskaźnik.

Poprzez wskaźnik możemy modyfikować wartość zmiennej, która jest zapisana pod adresem wskazywanym przez wskaźnik.

Przy użyciu wskaźników możemy przeprowadzać takie same operacje jak przy użyciu zmiennych

```
float liczba1, liczba2, suma;
float *wsk1, *wsk2; // tym razem wskaźniki typu float

wsk1=&liczba1; // ustawiamy wskaźnik wsk1 na adres zmiennej liczba1
wsk2=&liczba2; // ustawiamy wskaźnik wsk2 na adres zmiennej liczba2
cout<<"Podaj pierwsza liczbe: "; // prosimy o liczby
cin>>liczba1;
cout<<"Podaj druga liczbe: ";
cin>>liczba2;

suma=(*wsk1)+(*wsk2); // obliczamy sumę
cout<<"Suma podanych liczb wynosi "<<suma;
getch();
```

Ta właściwość będzie przydatna przy korzystaniu z funkcji

Ćwiczenie

Napisz program, który pobierze 2 liczby, ustawi wskaźniki na ich adresach i obliczy średnią tych liczb. Skorzystaj ze wskaźników.

Wyrażenia `*wsk1` i `*wsk2` są w nawiasach. Gdybyśmy nie dali nawiasów to nie zwiększylibyśmy wartości zmiennej lecz adres, czyli przesunęlibyśmy się do kolejnej komórki pamięci. Bez nawiasów, wyrażenie `*wsk1++` traktowane jest jako `wsk1++`. Jak widać, zwiększamy adres a nie wartość.

```
float *wsk;
float liczba;

wsk=&liczba;
cout<<"Adres komórki to "<<wsk<<endl;
*wsk++; // dokładnie to samo co wsk++
cout<<"Adres komórki to "<<wsk<<endl;
wsk++; // to samo co *wsk++
cout<<"Adres komórki to "<<wsk<<endl;

getch();
```

Napiszmy program, w którym zadeklarujemy tablicę i spróbujemy wypisać na ekran jej nazwę:

```
float liczby[10];
```

```
cout<<liczby;
getch();
```

Coś nam to przypomina. Jest to adres tablicy. Z tego wnioskujemy, że **nazwa tablicy jest jednocześnie wskaźnikiem do pierwszego elementu tej tablicy. Jest to wskaźnik stały**. Przekonamy się o tym, wypisując wartość zmiennej, na którą wskazuje wskaźnik. **Jest to wskaźnik stały**. To znaczy, że nie można na wskaźnik liczby podstawić wartości liczby+1.

```
float liczby[10];
```

```
liczby[0]=5; // za pierwszy element podstawiamy 5
cout<<*liczby; // piszemy wartość zmiennej, na którą wskazuje wskaźnik liczby.
getch();
```

Zapiszmy do tablicy kilka elementów i zwiększmy o i adres na który wskazuje wskaźnik liczby.

```
float liczby[10];
int i;

for(i=0; i<5; i++) // wczytujemy 5 liczb
{
    cout<<"podaj liczbe: ";
    cin>>liczby[i];
}

cout<<"adresy:"<<endl<<endl;
for(i=0; i<5; i++) // wypisujemy 5 adresów
{
    cout<<liczby+i<<endl; // bo liczby jest wskaźnikiem stałym. Wypisujemy adresy
}

getch();
```

Wyświetliły się kolejne adresy komórek pamięci. Są to komórki, w których zapisane są kolejne elementy tablicy. Aby się o tym przekonać, przeanalizuj przykład:

```
float liczby[10];
int i;

for(i=0; i<5; i++) // wczytujemy 5 liczb
{
    cout<<"podaj liczbe: ";
    cin>>liczby[i];
```

```
}

cout<<"wartosci:"<<endl<<endl;
for(i=0; i<5; i++) // wypisujemy 5 adresów
{
cout<<*(liczby+i)<<endl; // bo liczby jest wskaźnikiem stałym. Wypisujemy wartości
}

getch();
```

Zadania

(w trakcie lekcji proszę zrobić dwa wybrane zadania i przesłać na pocztę: marek@zstio-elektronika.pl Do niedzieli do godz. 12:00 proszę zrobić jeszcze dwa inne zadania i również wysłać na wskazany adres)

1. Napisz program, który wypisze na ekran adres zadeklarowanej zmiennej.
2. Napisz program, który pobierze wartość zmiennej, wypisze na ekran jej adres oraz wartość.
3. Napisz program, który przy użyciu wskaźników obliczy różnicę dwóch liczb.
4. Napisz program, który przy użyciu wskaźników obliczy średnią trzech liczb.
5. Napisz program, w którym zadeklarujesz tablicę. Wypisz na ekran adres jej pierwszego elementu.
6. Napisz program, w którym zadeklarujesz tablicę. Wypisz na ekran adres jej czwartego elementu.
7. Napisz program, w którym wczytasz wartości do tablicy i wypiszesz ich adresy.
8. Napisz program, w którym wczytasz i wypiszesz na ekran elementy tablicy przy użyciu wskaźników
9. Napisz program, w którym wczytasz i wypiszesz na ekran elementy tablicy większe od 0 przy użyciu wskaźników.
10. Napisz program, w którym wczytasz i wypiszesz na ekran elementy tablicy przy użyciu wskaźników oraz obliczysz ich średnią.
11. Napisz program, w którym wczytasz elementy tablicy, obliczysz ich średnią oraz wpiszesz elementy większe od średniej. Wszystko przy pomocy wskaźników.