

Programowanie C++

Wykład 4 (23.03.2016)

Uwaga: slajdy w znacznej części bazują
na wykładzie Dariusza Wardeckiego

Powtórzenie

Co zwraca funkcja i jakie wartości zostaną wypisane?

```
#include<iostream>
using namespace std;

void add(int a, int b){
    a += 2;
    b += 2;
}

int main(){
    int a = 10;
    int b = 20;
    add(a, b);
    cout << a << "\t" << b << endl;
    return 0;
}
```

Powtórzenie

Co zwróci funkcja?

```
#include<iostream>
using namespace std;

double suma(int n)
{
    double s = 0;
    double a_n = 1;
    for(int i = 1; i < n; i++)
    {
        a_n /= 2*i; // a_n = 1/n!!
        s += (1/2)*a_n;
    }
    return s;
}
```

Adresowanie pamięci

adres	komórka pamięci	wielkość
0x7fff5b7359d0		1 bajt
0x7fff5b7359d1		1 bajt
0x7fff5b7359d2		1 bajt
0x7fff5b7359d3		1 bajt
0x7fff5b7359d4		1 bajt
0x7fff5b7359d5		1 bajt
0x7fff5b7359d6		1 bajt
0x7fff5b7359d7		1 bajt
...

Adresowanie pamięci

adres	komórka pamięci	wielkość
0x7fff5b7359d0		1 bajt
0x7fff5b7359d1		1 bajt
0x7fff5b7359d2		1 bajt
0x7fff5b7359d3		1 bajt
0x7fff5b7359d4		1 bajt
0x7fff5b7359d5		1 bajt
0x7fff5b7359d6		1 bajt
0x7fff5b7359d7		1 bajt
...

char

Adresowanie pamięci

adres	komórka pamięci	wielkość
0x7fff5b7359d0		1 bajt
0x7fff5b7359d1		1 bajt
0x7fff5b7359d2		1 bajt
0x7fff5b7359d3		1 bajt
0x7fff5b7359d4		1 bajt
0x7fff5b7359d5		1 bajt
0x7fff5b7359d6		1 bajt
0x7fff5b7359d7		1 bajt
...

char

int, float

Adresowanie pamięci

adres	komórka pamięci	wielkość
0x7fff5b7359d0		1 bajt
0x7fff5b7359d1		1 bajt
0x7fff5b7359d2		1 bajt
0x7fff5b7359d3		1 bajt
0x7fff5b7359d4		1 bajt
0x7fff5b7359d5		1 bajt
0x7fff5b7359d6		1 bajt
0x7fff5b7359d7		1 bajt
...

char

int, float

double

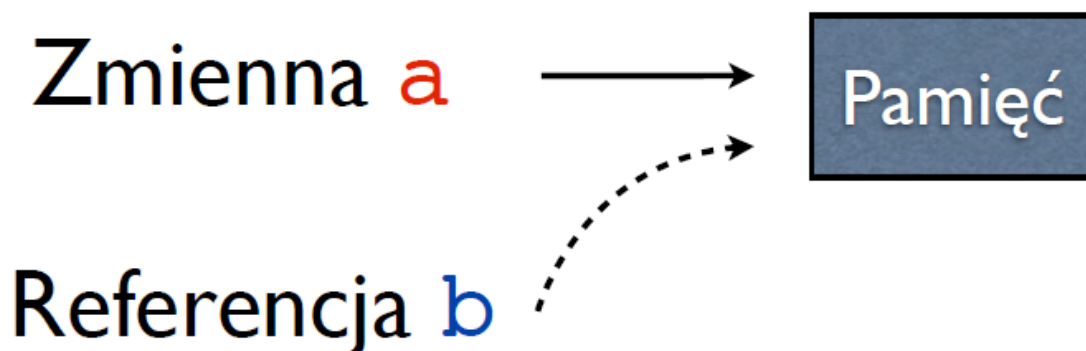
Jak odczytać adres?

operator `&` – zwraca adres zmiennej

```
#include<iostream>
using namespace std;

int main()
{
    double s = 66;
    cout << "Zmienna s jest pod adresem " << &s
    << " i ma wartosc " << s << endl;
    return 0;
}
```


Referencje



```
double a = 5;  
double& b = a;
```

~~double& b;~~

```
b = 7; // => a = 7
```

Przekazywanie referencji do funkcji

```
#include<iostream>
using namespace std;

void add(int a, int b){
    a += 2;
    b += 2; // Funkcja pracuje na kopii!!!
}

int main(){
    int a = 10;
    int b = 20;
    add(a, b);
    cout << a << "\t" << b << endl;
    return 0;
}
```

10 20

Przekazywanie referencji do funkcji

```
#include<iostream>
using namespace std;

void add(int& a, int& b){
    a += 2;
    b += 2; // Funkcja pracuje na oryginale!!!
}

int main(){
    int a = 10;
    int b = 20;
    add(a, b);
    cout << a << "\t" << b << endl;
    return 0;
}
```

Wskaźniki

Wskaźnik jest zmienną, która przechowuje adres innej zmiennej.

```
double a = 5;  
double* b = &a;
```

```
*b = 7; // => a = 7
```

Zmienna **a**



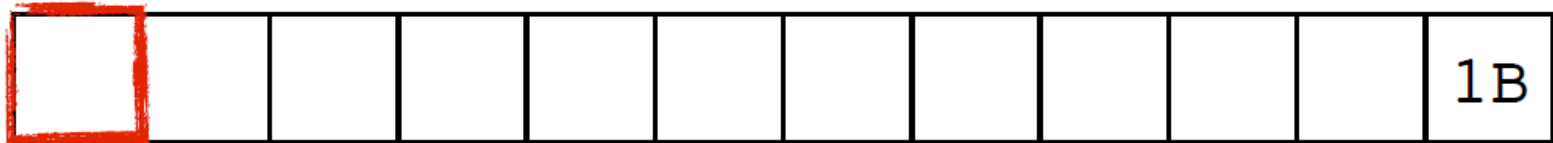
Adres: 0xffaa
Wartość: 5

Wskaźnik **b**



Adres: 0xffbb
Wartość: 0xffaa

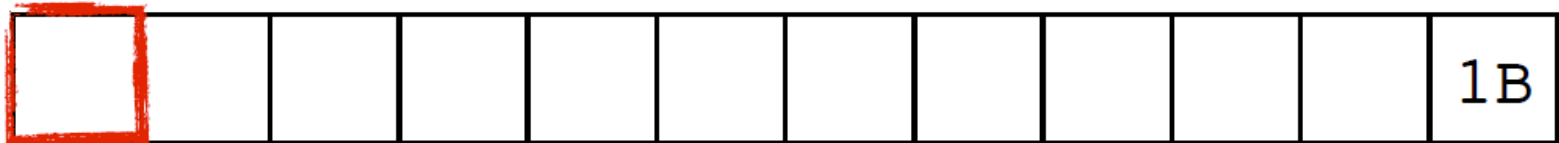
Wskaźniki



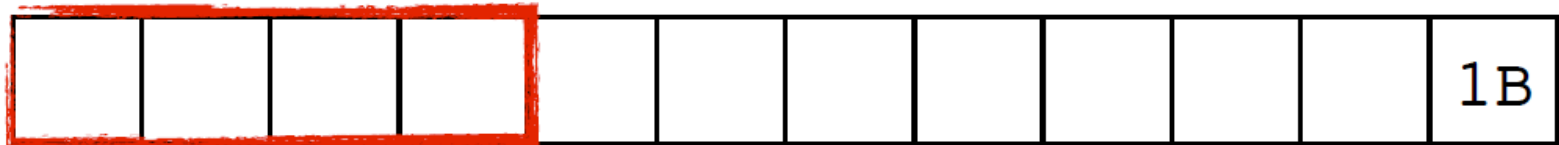
↑
`char* wsk;`



Wskaźniki



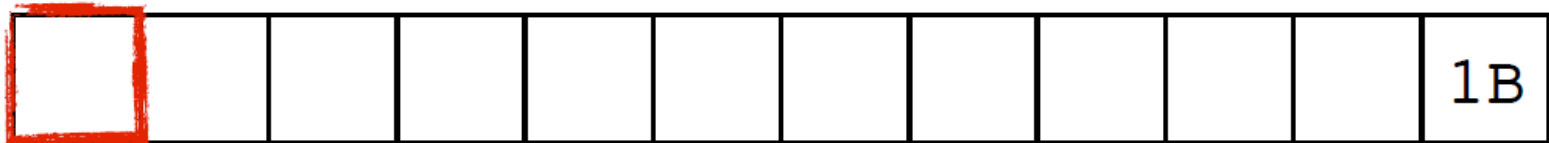
↑
`char* wsk;`



↑
`int* wsk;`



Wskaźniki



↑
`char* wsk;`



↑
`int* wsk;`



↑
`double* wsk;`

Przekazywanie wskaźnika do funkcji

```
#include<iostream>
using namespace std;

void add(int a, int b){
    a += 2;
    b += 2; // Funkcja pracuje na kopii!!!
}

int main(){
    int a = 10;
    int b = 20;
    add(a, b);
    cout << a << "\t" << b << endl;
    return 0;
}
```

10 20

Przekazywanie wskaźnika do funkcji

```
#include<iostream>
using namespace std;

void add(int* a, int* b){
    a += 2;
    b += 2; // Funkcja pracuje na oryginalne!!!
}

int main(){
    int a = 10;
    int b = 20;
    add(&a, &b);
    cout << a << "\t" << b << endl;
    return 0;
}
```

Przekazywanie funkcji do funkcji

typ funkcja(typ (*fun)(typ, typ, ...), ...)

```
#include <cmath>
#include <iostream>
using namespace std;

double trapezoid(double (*f)(double), double a, double b, int n)
{
    double h=(b-a)/n, s=0;
    for (int i=0; i<n; i++)
        s += h*(f(a+i*h) + f(a+(i+1)*h))/2;
    return s;
}

int main ()
{
    cout << trapezoid(sin,0,M_PI,100) << endl;
    return 0;
}
```

Struktury

```
struct moja_struktura
{
    typ zmienna1;
    typ zmienna2;
    ...
};
main()
{
    moja_struktura obiekt;
    ...
}
```

Struktury

```
struct osoba
{
    string imie;
    int wiek;
};
main()
{
    osoba student;
    student.imie = "Pawel";
    student.wiek = 20;
}
```

```
main()
{
    osoba student;
    osoba *wsk = &student;
    wsk->imie = "Pawel";
    wsk->wiek = 20;
}
```

Struktury

```
struct osoba
{ ... };
main()
{
    osoba *wsk = new osoba;

    wsk->imie = "Pawel";
    wsk->wiek = 20;
}
```