

```
// Użycie wskaźników

#include <iostream>

using namespace std;

int main()
{

    double zmienna_double=4.6;
    double *pdouble=0;
    pdouble=&zmienna_double;
    cout << pdouble <<endl;
    cout << *pdouble <<endl;
    pdouble=new double(23.5);
    cout << pdouble <<endl;
    cout << *pdouble <<endl;
    double nowa_zmienna=*pdouble;
    cout <<nowa_zmienna<<endl;
    pdouble=new double;
```

```
cout << pdouble <<endl;
    cout << *pdouble <<endl;
    *pdouble= 34.56;
    cout << pdouble <<endl;
    cout << *pdouble <<endl;
    cout<<endl;

    system ("pause");
    return 0;
```

```
}
```

```
0024FB68
4.6
00784FB8
23.5
23.5
00785000
-6.27744e+066
00785000
34.56
```

Press any key to continue . . .

```
// Wskaźniki i tablice

#include <iostream>

using namespace std;

int main()
{

    int tablica[10]=
    {0,2,4,6,8,10,12,14,16,18};
    int *pint=NULL;
    pint=&tablica[3];
    cout << *pint <<endl;
    cout << pint <<endl;
    pint+=1;
    cout << *pint <<endl;
    cout << pint <<endl;
    *pint+=3;
    cout << *pint <<endl;
    cout << pint <<endl;
    pint+=1;
```

```
cout << *pint <<endl;
    cout << pint <<endl;
    pint-=1;
    cout << *pint <<endl;
    cout << pint <<endl;
```

```
    system ("pause");
    return 0;
```

```
}
```

```
6
```

```
0012FF48
```

```
8
```

```
0012FF4C
```

```
11
```

```
0012FF4C
```

```
10
```

```
0012FF50
```

```
11
```

```
0012FF4C
```

```
Aby kontynuować, naciśnij dowolny klawisz . . .
```

```

// Wskazniki

#include<iostream>
using namespace std;

void main()
{
    int tablica[10]={0,3,6,9,12,15,18,21};
    int *ptr_1=&tablica[0];
    int *ptr_2=&tablica[2];
    int *ptr_3=&tablica[9];

    cout<<"Adres : "<<ptr_1<<"      Wartosc : "<<*ptr_1<<endl;
    cout<<"Adres : "<<ptr_2<<"      Wartosc : "<<*ptr_2<<endl;
    cout<<"Adres : "<<ptr_3<<"      Wartosc :
"<<*ptr_3<<endl<<"\n";

    ptr_1++;
    cout<<"Adres : "<<ptr_1<<"      Wartosc : "<<*ptr_1<<endl;
    cout<<"Adres : "<<ptr_2<<"      Wartosc : "<<*ptr_2<<endl;
    cout<<"Adres : "<<ptr_3<<"      Wartosc :
"<<*ptr_3<<endl<<"\n";
}

```

```

ptr_1+=7;
cout<<"Adres : "<<ptr_1<<"      Wartosc : "<<*ptr_1<<endl;
cout<<"Adres : "<<ptr_2<<"      Wartosc : "<<*ptr_2<<endl;
cout<<"Adres : "<<ptr_3<<"      Wartosc : "<<*ptr_3<<endl<<"\n";

*ptr_1+=12;
cout<<"Adres : "<<ptr_1<<"      Wartosc : "<<*ptr_1<<endl;
cout<<"Adres : "<<ptr_2<<"      Wartosc : "<<*ptr_2<<endl;
cout<<"Adres : "<<ptr_3<<"      Wartosc : "<<*ptr_3<<endl<<"\n";

ptr_1-=5;
cout<<"Adres : "<<ptr_1<<"      Wartosc : "<<*ptr_1<<endl;
cout<<"Adres : "<<ptr_2<<"      Wartosc : "<<*ptr_2<<endl;
cout<<"Adres : "<<ptr_3<<"      Wartosc : "<<*ptr_3<<endl<<"\n";

//ptr_3=ptr_1+ptr_2;
*ptr_3=*ptr_1+*ptr_2;
cout<<"Adres : "<<ptr_1<<"      Wartosc : "<<*ptr_1<<endl;
cout<<"Adres : "<<ptr_2<<"      Wartosc : "<<*ptr_2<<endl;
cout<<"Adres : "<<ptr_3<<"      Wartosc : "<<*ptr_3<<endl<<"\n";

system("pause");
return;
}

```

Adres : 00C0FE04	Wartosc : 0
Adres : 00C0FE0C	Wartosc : 6
Adres : 00C0FE28	Wartosc : 0
Adres : 00C0FE08	Wartosc : 3
Adres : 00C0FE0C	Wartosc : 6
Adres : 00C0FE28	Wartosc : 0
Adres : 00C0FE24	Wartosc : 0
Adres : 00C0FE0C	Wartosc : 6
Adres : 00C0FE28	Wartosc : 0
Adres : 00C0FE24	Wartosc : 12
Adres : 00C0FE0C	Wartosc : 6
Adres : 00C0FE28	Wartosc : 0
Adres : 00C0FE10	Wartosc : 9
Adres : 00C0FE0C	Wartosc : 6
Adres : 00C0FE28	Wartosc : 0
Adres : 00C0FE10	Wartosc : 9
Adres : 00C0FE0C	Wartosc : 6
Adres : 00C0FE28	Wartosc : 15

Press any key to continue . . .

```
//reinterpret_cast

#include<iostream>
using namespace std;

void main()
{
    int a=3;
    int *wsk_int=&a;
    cout<<wsk_int<<endl;
    double *wsk_d=0;
    //wsk_d=wsk_int;
    wsk_d=reinterpret_cast<double*>(wsk_int);
    cout<<wsk_int<<endl;

    system("pause");
    return;
}
```

0012FF60

0012FF60

Aby kontynuować, naciśnij dowolny klawisz . . .

Wskaźniki typu const

1. `const int pointer` - wskaźnik do wielkości całkowitej, która nie może być zmieniana
2. `int * const pointer` - stały wskaźnik do wartości całkowite - wartość może być zmieniana, wskaźnik nie może pokazywać niczego innego
3. `const int * const pointer` - połączenie 1 i 2

Utracone wskaźniki

Po usunięciu wskaźnika za pomocą `delete` nie wolno odwoływać się do niego - grozi to zawieszeniem komputera

```

// Wskazniki_const

#include<iostream>
using namespace std;

void main()
{
    int tablica[10]={0,3,6,9,12,15,18,21};
    int *ptr_1=&tablica[0];
    const int *ptr_2=&tablica[1];
    int* const ptr_3=&tablica[2];
    const int* const ptr_4=&tablica[3];
    int const *ptr_5=&tablica[9];

    cout<<"Adres : "<<ptr_1<<"      Wartosc : "<<*ptr_1<<endl;
    cout<<"Adres : "<<ptr_2<<"      Wartosc : "<<*ptr_2<<endl;
    cout<<"Adres : "<<ptr_3<<"      Wartosc : "<<*ptr_3<<endl;
    cout<<"Adres : "<<ptr_4<<"      Wartosc : "<<*ptr_4<<endl;
    cout<<"Adres : "<<ptr_5<<"      Wartosc :
"<<*ptr_5<<endl<<"\n";

    ptr_1+=5;
    cout<<"Adres : "<<ptr_1<<"      Wartosc : "<<*ptr_1<<endl;

```

```

*ptr_1+=4;
    cout<<"Adres : "<<ptr_1<<"      Wartosc :
"<<*ptr_1<<endl<<"\n";

    ptr_2+=4;
    cout<<"Adres : "<<ptr_2<<"      Wartosc :
"<<*ptr_2<<endl<<"\n";
    //*ptr_2+=12;
    *ptr_1+=4;
    cout<<"Adres : "<<ptr_1<<"      Wartosc :
"<<*ptr_1<<endl;
    cout<<"Adres : "<<ptr_2<<"      Wartosc :
"<<*ptr_2<<endl<<"\n";

    //ptr_3+=4;
    cout<<"Adres : "<<ptr_2<<"      Wartosc :
"<<*ptr_2<<endl<<"\n";
    *ptr_3+=12;
    cout<<"Adres : "<<ptr_3<<"      Wartosc :
"<<*ptr_3<<endl<<"\n";

```

```
/* ptr_4+=5;
   *ptr_4=*ptr_1+*ptr_2; */
ptr_5-=2;
cout<<"Adres : "<<ptr_5<<"      Wartosc :
"<<*ptr_5<<endl<<"\n";

    system("pause");
    return;
}
```

Adres	:	006DFA28	Wartosc	:	0
Adres	:	006DFA2C	Wartosc	:	3
Adres	:	006DFA30	Wartosc	:	6
Adres	:	006DFA34	Wartosc	:	9
Adres	:	006DFA4C	Wartosc	:	0
Adres	:	006DFA3C	Wartosc	:	15
Adres	:	006DFA3C	Wartosc	:	19
Adres	:	006DFA3C	Wartosc	:	19
Adres	:	006DFA3C	Wartosc	:	23
Adres	:	006DFA3C	Wartosc	:	23
Adres	:	006DFA3C	Wartosc	:	23
Adres	:	006DFA30	Wartosc	:	18
Adres	:	006DFA44	Wartosc	:	21

Press any key to continue . . .

```

// Rzutowanie const_cast

#include<iostream>
using namespace std;

int main()
{
    double A[4]={0.0,1.1,2.2,3.3};
    double const *wsk_1=&A[0];
    cout<<*wsk_1<<endl;

    //double *wsk_2=wsk_1;
    double *wsk_2=const_cast<double*>(wsk_1);
    wsk_2++;      cout<<*wsk_2<<endl;

    double *wsk_3=&A[2];
    const double *wsk_4=const_cast<const double*>(wsk_3);
    //*wsk_4-=0.5;
    wsk_4++;
    cout<<*wsk_4<<endl;
    system("pause");
    return 0;
}

```

```

0
1.1
3.3
Press any key to continue . . .

```

```

// tablica wskaznikow
#include<iostream>
using namespace std;

int main()
{
    char *Napisy[6]=
    {
        "Napis_0",
        "Napis_1",
        "Napis_2",
        "Napis_3",
        "Napis_4"
    };
    for (int i=0; i<5 ; ++i)
    {cout<<Napisy[i]<<"\t"<<&Napisy[i]<<"\t"<<*Napisy[i]<<
endl;}
    cout<<endl;

    Napisy[5]="Napis_5";
    for (int i=0; i<6 ; ++i)
    {cout<<Napisy[i]<<"\t"<<&Napisy[i]<<"\t"<<*Napisy[i]<<
endl;}

```

```
        system("pause");        return 0;
    }
```

```
Napis_0 0087FD94      N
Napis_1 0087FD98      N
Napis_2 0087FD9C      N
Napis_3 0087FDA0      N
Napis_4 0087FDA4      N
```

```
Napis_0 0087FD94      N
Napis_1 0087FD98      N
Napis_2 0087FD9C      N
Napis_3 0087FDA0      N
Napis_4 0087FDA4      N
Napis_5 0087FDA8      N
Press any key to continue . . .
```


Referencje

Referencja jest aliasem - inną nazwą.

& - operator referencji

```
int &rReferencja = zmienna; // referencja do  
zmienna
```

```
// Referencje

#include <iostream>

using namespace std;

int main()
{
    int zmienna_1=2;
    int &rZmienna=zmienna_1;
    cout << "zmienna_1 \t" << zmienna_1<<endl;
    cout << "rZmienna \t" << rZmienna<<endl;
    cout << "&zmienna_1 \t"<<&zmienna_1<<endl;
    cout << "&rZmienna \t"<<&rZmienna<<endl;

    int zmienna_2= 5;
    rZmienna=zmienna_2;
    cout << "zmienna_1 \t" << zmienna_1<<endl;
```

```

cout << "zmienna_2 \t" << zmienna_2<<endl;
    cout << "rZmienna \t" << rZmienna<<endl;
    cout << "&zmienna_1 \t"<<&zmienna_1<<endl;
    cout << "&zmienna_2 \t"<<&zmienna_2<<endl;
    cout << "&rZmienna \t"<<&rZmienna<<endl;
    cout <<endl;

    system("pause");
    return 0;
}

```

```

zmienna_1      2
rZmienna      2
&zmienna_1    005CFA98
&rZmienna     005CFA98
zmienna_1     5
zmienna_2     5
rZmienna      5
&zmienna_1    005CFA98
&zmienna_2    005CFA80
&rZmienna     005CFA98

```

Press any key to continue . . .

Struktura funkcji w C++

- Nagłówek funkcji

`zwracany_typ nazwa_funkcji(lista_parametrów);`

- Ciało funkcji
- Instrukcja `return wyrażenie;`

```
// funkcja

#include<iostream>
using namespace std;

double potega(double podstawa, int wykladnik);
//prototyp funkcji

void main()
{
    int i, petle, wykl;
    double podst;
    cout<<"Liczba petli = ";
    cin>>petle;
    for (i=0; i<petle; i++)
    {
        cout<<endl<<"Podstawa = ";
        cin>>podst;
        cout<<"Wykladnik = ";
        cin>>wykl;
```

```

double wynik=potega (podst,wykl);
        cout<<podst<<" do potegi ";
        cout<<wykl<<" = "<<wynik<<"      Adres :
"<<&wynik<<endl;

    }
    system("pause");
    return;
}

double potega(double podstawa, int wykladnik)
{
    double wynik=1.0;
    for (int k=1; k<=wykladnik; k++)
    {wynik*=podstawa;}
    cout<<"Adres lokalny : "<<&wynik<<endl;
    return wynik;
}

```

Liczba petli = 3

Podstawa = 4

Wykladnik = 5

Adres lokalny : 00DDF890

4 do potegi 5 = 1024 Adres : 00DDF984

Podstawa = 6

Wykladnik = 4

Adres lokalny : 00DDF890

6 do potegi 4 = 1296 Adres : 00DDF984

Podstawa = 3.5

Wykladnik = 3

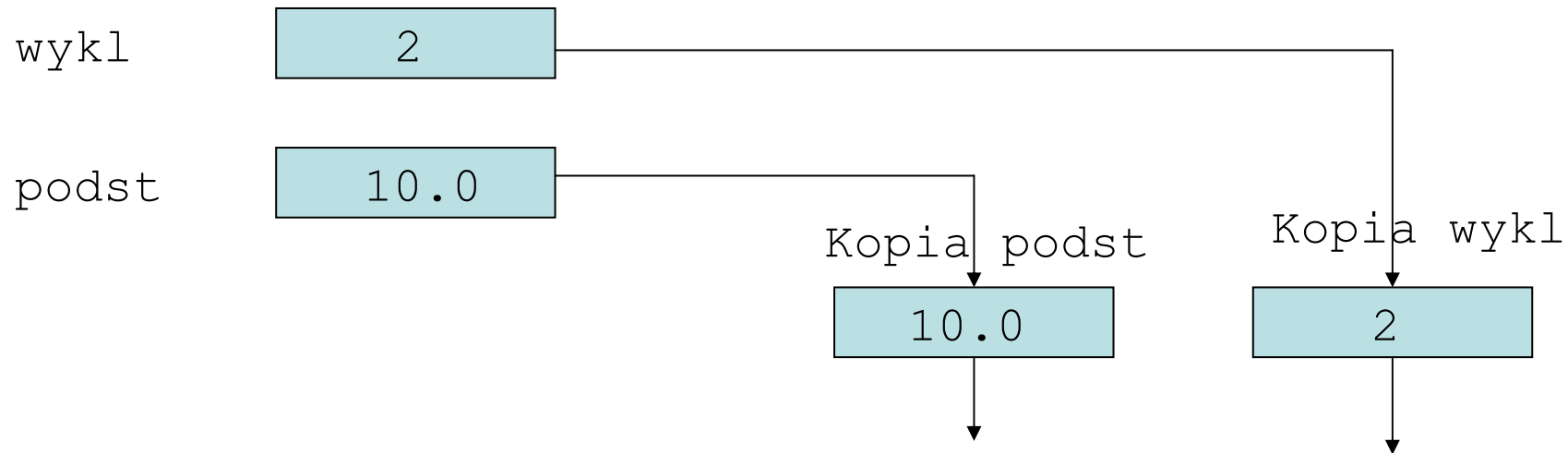
Adres lokalny : 00DDF890

3.5 do potegi 3 = 42.875 Adres :
00DDF984

Press any key to continue . . .

Przekazywanie argumentów w C++

Przekazywanie przez wartość



```
double potega (double podstawa,  
int wykladnik);  
  
{  
  
...//oryginalne argumenty niedostępne  
  
}
```



```

// Przekazywanie argumentu przez wskaźnik

#include<iostream>
using namespace std;

double potega(double *podstawa, int *wykladnik);
//prototyp funkcji

void main()
{
    int i, petle, wyk1;
    double podst;
    double* pd;
    int* pi;
    cout<<"Liczba petli = ";
    cin>>petle;
    for (i=0; i<petle; i++)
    {
        cout<<endl<<"Podstawa = ";
        cin>>podst;
    }
}

```

```

pd=&podst;
    cout<<"      Adres : "<<pd<<endl;
    cout<<"Wykladnik = ";
    cin>>wykl;
    pi=&wykl;
    cout<<"      Adres : "<<pi<<endl;
    double wynik=potega(pd,pi);
    cout<<podst<<" do potegi ";
    cout<<wykl<<" = "<<wynik<<"      Adres :
"<<&wynik<<endl;

    }
    system("pause");
    return;
}

```

```
double potega(double *podstawa, int *wykladnik)
{
    cout<<"Adresy : "<<podstawa<<"
"<<wykladnik<<endl;
    double wynik=1.0;
    for (int k=1; k<=*wykladnik; k++)
    {wynik*=*podstawa;}
    cout<<"Adres lokalny : "<<&wynik<<endl;
    return wynik;
}
```

Liczba petli = 2

Podstawa = 3.5

Adres : 00ECFB08

Wykladnik = 2

Adres : 00ECFB18

Adresy : 00ECFB08 00ECFB18

Adres lokalny : 00ECF9F0

3.5 do potegi 2 = 12.25 Adres : 00ECFAE0

Podstawa = 4

Adres : 00ECFB08

Wykladnik = 3

Adres : 00ECFB18

Adresy : 00ECFB08 00ECFB18

Adres lokalny : 00ECF9F0

4 do potegi 3 = 64 Adres : 00ECFAE0

Press any key to continue . . .

```
// Przekazywanie argumentu przez referencję

#include<iostream>
using namespace std;

double potega(double& podstawa, int& wykladnik);
//prototyp funkcji

void main()
{
    int i, petle, wykl;
    double podst;
    double &rd=podst;
    int &ri=wykl;
    cout<<"Liczba petli = ";
    cin>>petle;
    for (i=0; i<petle; i++)
    {
        cout<<endl<<"Podstawa = ";
        cin>>podst;
```

```

rd=podst;
    cout<<"      Adres : "<<&rd<<endl;
    cout<<"Wykladnik = ";
    cin>>wykl;
    ri=wykl;
    cout<<"      Adres : "<<&ri<<endl;
    double wynik=potega(rd,ri);
    cout<<podst<<" do potegi ";
    cout<<wykl<<" = "<<wynik<<"      Adres :
"<<&wynik<<endl;

    }
    system("pause");
    return;
}

```

```
double potega(double &podstawa, int &wykladnik)
{
    cout<<"Adresy : "<<&podstawa<<"
" <<&wykladnik<<endl;
    double wynik=1.0;
    for (int k=1; k<=wykladnik; k++)
    {wynik*=podstawa;}
    cout<<"Adres lokalny : "<<&wynik<<endl;
    return wynik;
}
```

Liczba petli = 2

Podstawa = 3

Adres : 0045FDD8

Wykladnik = 2

Adres : 0045FDE8

Adresy : 0045FDD8 0045FDE8

Adres lokalny : 0045FCC0

3 do potegi 2 = 9 Adres : 0045FDB0

Podstawa = 4.5

Adres : 0045FDD8

Wykladnik = 3.1

Adres : 0045FDE8

Adresy : 0045FDD8 0045FDE8

Adres lokalny : 0045FCC0

4.5 do potegi 3 = 91.125 Adres : 0045FDB0

Press any key to continue . . .


```

//przekazywanie przez wartosc

#include <iostream>
using namespace std;

int Dodaj_5(int liczba);

int main()
{
    int liczba = 3;
    cout << endl<<"Dodaj_5(liczba) = " <<
Dodaj_5(liczba);
    cout << endl << "liczba = " << liczba << endl;

    cout << endl<<"Dodaj_5(liczba) = " <<
Dodaj_5(liczba);
    liczba=Dodaj_5(liczba);
    cout << endl << "liczba = " << liczba << endl;
    system("pause");
    return 0;
}

```

```
int Dodaj_5(int liczba)
{
liczba+=5;
return liczba;
}
```

```
Dodaj_5(liczba) = 8
liczba = 3
```

```
Dodaj_5(liczba) = 8
liczba = 8
```

```
Press any key to continue . . .
```