

Wstęp do programowania

- wykład 4

dr Piotr Jastrzębski

Stałe zmiennoprzecinkowe

- ▶ zapis 5.2 czy 5.4E3 domyślnie traktowany jest jako double

```
cout << typeid(4.5).name() << endl;
```

- ▶ to znaczenia float używamy f lub F
- ▶ to znaczenia long double używamy l lub L

<https://en.cppreference.com/w/c/types/limits>

https:

[//gist.github.com/pjastr/c3827960238a24ed4cebb54181c30aad](https://gist.github.com/pjastr/c3827960238a24ed4cebb54181c30aad)

https:

[//gist.github.com/pjastr/d406716d11291dcf09201a8eb222b34f](https://gist.github.com/pjastr/d406716d11291dcf09201a8eb222b34f)

Problem 0.1?

Binarnie to:

0.0001100110011001100110011001100110011001100110011001100110011001100

Jako float

0.000110011001100110011001101

Jako double

0.0001100110011001100110011001100110011001100110011001100110011010

A dziesiętnie

float

0.10000002384185791015625

double

0.1000000000000000088817841970012523233890533447265625

Typ char - znaki

Typ do przechowywania znaków.

<https://en.cppreference.com/w/cpp/language/types>

Tablica ASCII

<https://pl.wikipedia.org/wiki/ASCII>

<https://upload.wikimedia.org/wikipedia/commons/d/dd/ASCII-Table.svg>

<https://www.youtube.com/watch?v=k-GH3mbvUro>

<https://sci-fi.narkive.pl/DkYfU5Dm/dlaczego-mark-u-ywaszesnastkowego-do-komunikacji>

https:

//gist.github.com/pjastr/5603573f1ca71e9b25ad7bc1df8f8359

https:

//gist.github.com/pjastr/fab8f8691b2a85686423841bbaf485a6

Stałe znakowe

Wpisujemy w apostrofach

'A'

'\n' - nowy wiersz

'\t' - tabulator

Typ bool - logiczny

```
bool a = true;  
bool b = false;  
cout << a <<endl;  
cout << b <<endl;
```


Rzutowanie

```
(typ) wartosc  
typ (wartosc)
```

https:

[//gist.github.com/pjastr/bd047a47b19cdd1a6504bb9e84cf01f6](https://gist.github.com/pjastr/bd047a47b19cdd1a6504bb9e84cf01f6)

Operacje arytmetyczne

Operatory arytmetyczne

Operator	Działanie
$a * b$	Mnożenie
a / b	Dzielenie
$a + b$	Dodawanie
$a - b$	Odejmowanie
$a \% b$	Reszta z dzielenia (modulo)
$++a$	Preinkrementacja (zwiększenie o 1)
$a++$	Postinkrementacja (zwiększenie o 1)
$--a$	Predekrementacja (zmniejszenie o 1)
$a--$	Postdekrementacja (zmniejszenie o 1)

Kolejność działań

https:

[//en.cppreference.com/w/cpp/language/operator_precedence](https://en.cppreference.com/w/cpp/language/operator_precedence)

Różnica między pre a post?

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

int main()
{
    int x = 10, a;
    a = ++x;
    cout << "a = " << a <<endl;
    cout << "x = " << x <<endl;
    return 0;
}
```

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

int main()
{
    int x = 10, a;
    a = x++;
    cout << "a = " << a <<endl;
    cout << "x = " << x <<endl;
    return 0;
}
```

Ciekawostka:

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

int main()
{
    int x = 10;
    cout << "Przed:" <<endl;
    cout << "x = " << x <<endl;
    x = x++;
    cout << "Po:" <<endl;
    cout << "x = " << x <<endl;
    return 0;
}
```

Warianty dzielenia

https:

[//gist.github.com/pjastr/92912986bb4e24c6529849f17cfcbbc2](https://gist.github.com/pjastr/92912986bb4e24c6529849f17cfcbbc2)

Złożone operatory przypisania

Operator	Działanie
<code>a = b = c</code>	Przypisanie, także w formie ciągu
<code>a *= b</code>	<code>a = a * b</code>
<code>a /= b</code>	<code>a = a / b</code>
<code>a += b</code>	<code>a = a + b</code>
<code>a -= b</code>	<code>a = a - b</code>
<code>a %= n</code>	<code>a = a % b</code>

https:

[//en.cppreference.com/w/cpp/language/operator_assignment](https://en.cppreference.com/w/cpp/language/operator_assignment)

Operatory porównania

Operator	Znaczenie	Przykład
>	Większe niż	$x > y$
<	Mniejsze niż	$x < y$
==	Równe	$x == y$
!=	Nie równa się	$x != y$
>=	Większe lub równe	$x >= y$
<=	Mniejsze lub równe	$x <= y$

https:

[//en.cppreference.com/w/cpp/language/operator_comparison](https://en.cppreference.com/w/cpp/language/operator_comparison)

Operatory logiczne

Operator	Znaczenie	Przykład
<code>&&</code>	i	<code>x && y</code>
<code> </code>	lub	<code>x y</code>
<code>!</code>	negacja	<code>!x</code>

https://en.cppreference.com/w/cpp/language/operator_logical

Operacje na bitach

https:

[//gist.github.com/pjastr/2ec140374fac47944fded18a355a79ea](https://gist.github.com/pjastr/2ec140374fac47944fded18a355a79ea)

AND &

[https://pl.wikipedia.org/wiki/Koniunkcja_\(logika\)](https://pl.wikipedia.org/wiki/Koniunkcja_(logika))

OR |

<https://pl.wikipedia.org/wiki/Alternatywa>

XOR ^

https:

[//pl.wikipedia.org/wiki/Alternatywa_roz%C5%82%C4%85czna](https://pl.wikipedia.org/wiki/Alternatywa_roz%C5%82%C4%85czna)

NOT ~

<https://pl.wikipedia.org/wiki/Negacja>

Przesunięcia

https://pl.wikipedia.org/wiki/Przesuni%C4%99cie_bitowe

<< >>