

Wstęp do programowania

- wykład 5

dr Piotr Jastrzębski

Warianty dzielenia

https:

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

Uzupełnienie:

Dzielenie z resztą

https:

[//gist.github.com/pjastr/3e584ad17829580ad33f0bf3572a5d31](https://gist.github.com/pjastr/3e584ad17829580ad33f0bf3572a5d31)

Działa aby było spełnione:

$$(a / b) * b + a \% b == a$$

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

<< >>

Problem:

Jak dokonać zamiany wartości dwóch zmiennych typu `int`?

Instrukcje warunkowe

```
if (wyrażenie) {  
    /* blok wykonany, jeśli wyrażenie jest prawdziwe */  
}
```

```
if (wyrażenie) {  
    /* blok wykonany, jeśli wyrażenie jest prawdziwe */  
} else {  
    /* blok wykonany, jeśli wyrażenie jest nieprawdziwe */  
}
```

```
switch (wyrażenie) {  
  case wartość1:  
    break;  
  case wartość2:  
    break;  
  /* ... */  
  default:  
    break;  
}
```

Przykłady:

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

int main()
{
    int a =5;
    if (a>0)
    {
        cout << "Liczba dodatnia" <<endl;
    }
    else
    {
        cout << "Liczba nie jest dodatnia" <<endl;
    }
    return 0;
}
```

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

int main()
{
    int a =-1;
    if (a)
    {
        cout << "Prawda" <<endl;
    }
    else
    {
        cout << "falsz" <<endl;
    }
    return 0;
}
```

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

int main()
{
    int a =-1;
    if (a)
        cout << "Prawda" <<endl;
    cout << "Falsz" <<endl;
    return 0;
}
```

Zagnieżdżony if

https:

[//gist.github.com/pjastr/3369f5eabc2ec329142b09f580580250](https://gist.github.com/pjastr/3369f5eabc2ec329142b09f580580250)

Pytanie?

https:

[//gist.github.com/pjastr/6dbf49092489e5c7d45ef41a030fea14](https://gist.github.com/pjastr/6dbf49092489e5c7d45ef41a030fea14)

Problemy

- ▶ szukanie rozwiązań równania kwadratowego
- ▶ szukanie najmniejszej/największej wartości

```
#include <iostream>
using namespace std;
int main()
{
    int liczba;
    cout << "Podaj liczbe: ";
    cin >> liczba;
    switch( liczba )
    {
        case 2:
            cout << "dwa" << endl;
            break;
        case 1:
            cout << "jeden" << endl;
            break;
        default:
            cout << "ani jeden, ani dwa" << endl;
            break;
    }
    return 0;
}
```