

Programownie I

Wykład 2

dr inż. Adam Zielonka

Instytut Matematyki,
Politechnika Śląska

Gliwice 12.10.2018

Operatory

Operatory relacyjne

- `==` równe $a == b$
- `!=` nierówne $a != b$
- `>` większy $a > b$
- `>=` większy lub równy $a >= b$
- `<` mniejszy $a < b$
- `<=` mniejszy lub równy $a <= b$

przykład

```
cout<<(3>2)<<endl;
cout<<boolalpha<<(3>2)<<" "<<1<<endl;
```

```
1
true 1
```

Operatory

Operatory logiczne

- `||` - suma logiczna np. `(a>b)|| (a>c)`
- `&&` - iloczyn logiczny np. `true&&false`
- `!` - negacja logiczna np. `!true`

Operatory

Operatory inkrementacji i dekrementacji

- postinkrementacja $a++$ *wartość a zostaje zwiększona o 1 po wykonaniu operacji*
- preinkrementacja $++a$ *wartość a zostaje zwiększona o 1 przed wykonaniem operacji*
- prostdekrementacji $a--$ *wartość a zostaje zmniejszona o 1 po wykonaniu operacji*
- predekrementacji $--a$ *wartość a zostaje zmniejszona o 1 przed wykonaniem operacji*

Instrukcja warunkowa

warunek – dowolne wyrażenie posiadające wartość logiczną

Schemat

```
if(warunek)
    jedno_polecenie;
```

```
if(warunek)
{
    blok poleceń
}
```

Przykład

```
int a=0,b=0;
cin>>a>>b;
if(a==b)
    cout<<"Te same liczby";
```

Instrukcja warunkowa

```
if(warunek)
{
    //blok poleceń dla true
}
else
{
    //blok poleceń dla false
}
```

Instrukcja warunkowa

```
if(war1)
{
    //blok poleceń dla war1=true
}
else if(war2)
{
    //blok poleceń dla war1=false i war2=true
}
else
{
    //blok poleceń dla war1=war2=false
}
```

Instrukcja warunkowa

- W najstarszych wersjach C++ nie było zmiennej typu bool i wówczas zmienna równa zero oznaczała stan: false pozostałe wartości rozumiane były jak obecnie stan: true
- Nadal istnieje ta możliwość
- Warunek może być bardzo skomplikowanym wyrażeniem, które po obliczeniu da wartość liczbową

Przykład

```
int a=20,b=16,d=2;
if(a%d+a%d)
{
    //
}
else
{
    cout<<d<<"/"<<a;
    cout<<" i "<<d<<"/"b;
}
```


UWAGA

```
int i=-12;
if(i=0)
{
    cout<<"Zero";
}
else if(i<0)
{
    cout<<"Ujemne";
}
else
{
    cout<<"Dodatnie";
}
cout<<endl;
```

Operator warunkowy

```
warunek ? polecenieT : polecenieF ;
```

Przykład 1

```
double x=0.0,y=0.0;  
cin >> x >> y;  
cout << x << " jest "  
cout << ( x>y ? "wieksze " : "niewieksze " );  
cout << "od " << y << endl;
```

Operator warunkowy

Przykład 2

```
double x=0.0,y=0.0;
cin>>x>>y;
cout<<x<<" jest ";
cout<< ( x>y ? "wieksze " : ( x<y ? "mniejszy " : "rowny " ) );
cout<<"od "<<y<<endl;
```

Pętla do-while

Schemat

```
do
    polecenie;
while( warunek );

do
{
    //blok poleceń
}while( warunek );
```

Przykład

```
int i=0;
int max_iter=10;
do
    cout<<i<<" ";
while(i++<max_iter);
cout<<endl;
do
{
    cout<<i<<" ";
    i-=2;
}while( i>0 );
```

Pętla do-while

Schemat

```
do
    polecenie;
while( warunek );

do
{
    //blok poleceń
}while( warunek );
```

Przykład

```
int i=0;
int max_iter=10;
do
    cout<<i<<" ";
while(i++<max_iter);
cout<<endl;
do
{
    cout<<i<<" ";
    i-=2;
}while( i>0 );
```

```
1 2 3 4 5 6 7 8 9 10
11 9 7 5 3 1
```

Pętla while

Schemat

```
while( warunek )
    polecenie;

while( warunek )
{
    //blok poleceń;
}
```

Przykład

```
int i=0;
int max_iter=10;
while(i++<max_iter)
    cout<<i<<" ";
cout<<endl;
while( i>0 )
{
    cout<<i<<" ";
    i-=2;
}
```

Pętla while

Schemat

```
while( warunek )  
    polecenie;  
  
while( warunek )  
{  
    //blok poleceń;  
}
```

Przykład

```
int i=0;  
int max_iter=10;  
while(i++<max_iter)  
    cout<<i<<" ";  
cout<<endl;  
while( i>0 )  
{  
    cout<<i<<" ";  
    i-=2;  
}
```

```
1 2 3 4 5 6 7 8 10  
11 9 7 5 3 1
```

Pętla for

Schemat pętli for

```
for(wyrażenie inicjalizujące ; warunek ; wyrażenie)
    polecenie;
```

```
for(wyrażenie inicjalizujące ; warunek ; wyrażenie)
{
    \\ blok poleceń;
}
```

przykład n!

```
int silnia=1,n=6;
for(int i=2;i<=n;i++)
    silnia*=i;
cout<<n<<"!="<<silnia<<endl;
```


Instrukcja break

break

Wywołanie instrukcji break powoduje natychmiastowe przerwanie instrukcji pętli.

```
int i=0;
for( ; ; )
{
    //instrukcje;
    if( i++==10 ) break;
}
```

Instrukcja break

Przykład

```
for (int i = 0; i < 4; i++)
{
    for (int j = 0; j < 4; j++)
    {
        if ( i == j ) break;
        cout << "(" << i << "," << j << " ) ";
    }
    cout << endl;
}
```

Instrukcja break

Przykład

```
for (int i = 0; i < 4; i++)
{
    for (int j = 0; j < 4; j++)
    {
        if ( i == j ) break;
        cout << "(" << i << "," << j << " ) ";
    }
    cout << endl;
}
```

Instrukcja break

Przykład

```
for (int i = 0; i < 4; i++)
{
    for (int j = 0; j < 4; j++)
    {
        if ( i == j ) break;
        cout << "(" << i << "," << j << " ) ";
    }
    cout << endl;
}
```

(1,0)

Instrukcja break

Przykład

```
for (int i = 0; i < 4; i++)
{
    for (int j = 0; j < 4; j++)
    {
        if ( i == j ) break;
        cout << "(" << i << ", " << j << " ) ";
    }
    cout << endl;
}
```

(1,0)

(2,0) (2,1)

Instrukcja break

Przykład

```
for (int i = 0; i < 4; i++)
{
    for (int j = 0; j < 4; j++)
    {
        if ( i == j ) break;
        cout << "(" << i << "," << j << " ) ";
    }
    cout << endl;
}
```

(1,0)
(2,0) (2,1)
(3,0) (3,1) (3,2)

Instrukcja continue

break

Wywołanie instrukcji `continue` powoduje natychmiastowe przerwanie bieżącego przebiegu pętli, ale wykonywany są jej kolejne przebiegi.

```
int n=4,n0=2;
for(int i=0 ; i<n ; i++)
{
    if( i==n0 ) continue;
    //instrukcje;
}
```

Instrukcja continue

Przykład

```
for (int i = 0; i < 4; i++)
{
    for (int j = 0; j < 4; j++)
    {
        if ( i < j ) continue;
        cout << "(" << i << "," << j << " ) ";
    }
    cout << endl;
}
```


Instrukcja continue

Przykład

```
for (int i = 0; i < 4; i++)
{
    for (int j = 0; j < 4; j++)
    {
        if ( i < j ) continue;
        cout << "(" << i << "," << j << " ) ";
    }
    cout << endl;
}
```

(0,0)

Instrukcja continue

Przykład

```
for (int i = 0; i < 4; i++)
{
    for (int j = 0; j < 4; j++)
    {
        if ( i < j ) continue;
        cout << "(" << i << "," << j << " ) ";
    }
    cout << endl;
}
```

(0,0)

(1,0), (1,1)

Instrukcja continue

Przykład

```
for (int i = 0; i < 4; i++)
{
    for (int j = 0; j < 4; j++)
    {
        if ( i < j ) continue;
        cout << "(" << i << ", " << j << " ) ";
    }
    cout << endl;
}
```

(0,0)
(1,0), (1,1)
(2,0) (2,1), (2,2)

Instrukcja continue

Przykład

```
for (int i = 0; i < 4; i++)
{
    for (int j = 0; j < 4; j++)
    {
        if ( i < j ) continue;
        cout << "(" << i << ", " << j << " ) ";
    }
    cout << endl;
}
```

(0,0)
(1,0), (1,1)
(2,0) (2,1), (2,2)
(3,0) (3,1) (3,2), (3,3)

Instrukcja switch

Instrukcja wielokrotnego wyboru, gdzie wyrażenie jest zmienną całkowitoliczbową

Schemat

```
switch(wyrażenie)
{
    case n1:
        instrukcja1a;
        ...
        break;
    case n2:
        instrukcja2a;
        ...
        break;
    ...
    default:
        instrukcjaNa;
        ...
        break;
}
```

Instrukcja switch

Przykład

```
cout << "Podaj cyfry od 0 do 3" << endl;
char z = 0;
cin >> z;
switch(z)
{
    case 48: cout << "zero";
             break;
    case 49: cout << "jeden";
             break;
    case 50: cout << "dwa";
             break;
    case 51: cout << "trzy";
             break;
    default:
             cout << "Nie podałeś cyfry od 0 do 3";
             break;
}
```

Zadanie

- ```
int a = 3 , b=0, c=0;
b=++a*3+++a;
cout << "a=" << a << " b=" << b <<endl;
a=3;
c=++a*3+a++;
cout << "a=" << a << " c=" << c <<endl;
```