



Управляващи оператори в C++

Оператор за присвояване

- `<променлива> = <израз>;`
- `<lvalue> = <rvalue>;`
- `<lvalue>` означава място в паметта със стойност, която може да се променя
- пример: променлива
- `<rvalue>` означава временна стойност, без специално място в паметта
- стандартно преобразуване на типовете

ОСНОВНИ ПОНЯТИЯ

- Операция (operator)
- Израз (expression)
- Оператор (statement)
- $\langle \text{израз} \rangle ::= \langle \text{константа} \rangle \mid$
 $\langle \text{променлива} \rangle \mid$
 $\langle \text{унарна_операция} \rangle \langle \text{израз} \rangle \mid$
 $\langle \text{израз} \rangle \langle \text{бинарна_операция} \rangle \langle \text{израз} \rangle$
- $\langle \text{оператор} \rangle ::= \langle \text{израз} \rangle ;$

Присвояването като операция

- **дясноасоциативна** операция
- $a = b = c = 2;$
- $a = (b = (c = 2));$
- **НЕ:** $((a = b) = c) = 2);$
- $\text{cout} \ll x + (b = 2);$

Операция за изброяване

- $\langle \text{израз}_1 \rangle, \langle \text{израз}_2 \rangle$
- оценява и двата израза, но крайният резултат е оценката на втория израз
- $a, b, c, d \Leftrightarrow (a, (b, (c, d)))$
- **дясно**асоциативна
- използва се рядко

Съкратени оператори за присвояване

- $a = a + 2 \Leftrightarrow a += 2$
- $-=, *=, /=, \%=$
- $a = a + 1 \Leftrightarrow ++a$
- $a = a - 1 \Leftrightarrow --a$
- $a++ \Leftrightarrow a = (b = a) + 1, b$

Оператор за блок

- {<оператор₁><оператор₂>...<оператор_n>}

- { {<оператор> } }

- Вложени блокове

```
{  
    int x = 2;  
    {  
        x += 2;  
        cout << x;  
    }  
}
```

Област на видимост (scope)

- областта на видимост се простира от дефиницията на променливата до края на блока, в който е дефинирана
- дефиниция на променлива със същото име в същия блок е забранена
- дефиниция на променлива във вложен блок покрива всички външни дефиниции със същото име

Празен оператор

- ;
- ; ↔ {}

Условен оператор

- **if** (<израз>) <оператор> [**else** <оператор>]
- if (x < 2) y = 2; if (x > 5) y = 5; else y = 3;
- if (A) X; \Leftrightarrow if (A) X; else;
- if (!A) X; else Y; \Leftrightarrow if (A) Y; else X;
- if (A && B) X; else Y; \Leftrightarrow
if (A) if (B) X; else Y; else Y;
- if (A || B) X; else Y; \Leftrightarrow
if (A) X; else if (B) X; else Y;

Условна операция

- `<булев_израз> ? <израз> : <израз>`
- тернарна операция
- Пример: `x = (y < 2) ? y + 1 : y - 2;`
- `A ⇔ A ? true : false`
- `!A ⇔ A ? false : true`
- `A && B ⇔ A ? B : false`
- `A || B ⇔ A ? true : B`

Оператор за многозначен избор

- **switch** (<израз>) {
 { **case** <конст_израз> : { <оператор> } }
 [**default** : { <оператор> }]
}

- Пример:

```
switch (x) {  
    case 1 : x++;  
    case 2 : x += 2;  
    default : x += 5;  
}
```

Оператор за прекъсване

- **break;**
- ```
switch (x) {
 case 1 : x++; break;
 case 2 : x += 2; break;
 default : x += 5;
}
```

# Задачи за if/else и switch

1. Да се провери дали три числа образуват растяща редица
2. Да се намери най-малкото от три числа
3. Да се подредят три числа в растяща редица
4. Да се провери дали три числа образуват Питагорова тройка
5. Да се транслитерира символ от кирилица на латиница
6. Да се пресметне избрана от потребителя функция

# Цикъл

- $x += 1; x += 2; x += 3; x += 4; x += 5;$
- $x += i; \text{ за } i = 1, 2, 3, 4, 5$
- индуктивен цикличен процес
- $\text{if} ( x \geq 10) x /= 10; \text{if} ( x \geq 10) x /= 10; \dots$
- $x /= 10; \text{ докато е вярно, че } x \geq 10$
- итеративен цикличен процес

# Оператор for

- **for** ( <израз> ; <израз> ; <израз> ) <оператор>
- **for** ( <инициализация> ; <условие > ; <коррекция> )  
<тяло>
- <инициализация>;  
if (<условие>) { <тяло> <коррекция>; }  
if (<условие>) { <тяло> <коррекция>; }  
...
- Изключение: <инициализация> може да е дефиниция на променлива.



# Оператор for

- Пример:

```
for (int i = 1, x = 0, y = 1; i < 5; i++) {
 x += i;
 y *= x;
}
```

# Задачи за for

1. Пресметнете  $n!$

2. Пресметнете сумата  $\sum_{i=0}^n \frac{x^i}{i!}$

3. Намерете броя на тези от числата  $x_i = n^3 + 5i^2n - 8i$ , които са кратни на 3

4. Намерете най-голямото число от вида  $x_i = n^3 + 5i^2n - 8i$  за  $i = 1, \dots, n$

# Оператор while

- **while** (<израз>) <оператор>
- if (<израз>) <оператор>  
if (<израз>) <оператор>  
...
- for ( ; <израз> ; ) <оператор>
- for ( <инициализация> ; <условие> ; <коррекция> ) <тяло>  
⇔  
<инициализация>; while (<условие>) { <тяло> <коррекция>; }

# Задачи за while

1. Пресметнете  $n!$
2. Намерете средното аритметично на поредица числа
3. Пресметнете сумата  $\sum_{i=0}^n \frac{x^i}{i!}$  с точност  $\varepsilon$
4. Намерете най-значимата цифра на  $n$
5. Проверете дали  $n$  съдържа цифрата 5

# Оператор do/while

- **do** <оператор> **while** (<израз>);

- <оператор>  
while (<израз>) <оператор>

- **Внимание:**

```
int i = 0, y = 0;
do {
 int x = i * i;
 y += x;
 i++;
} while (x < 10);
```

# while или do/while?

- 0 или 1?
- `do <тяло> while (<условие>);`  
има същия ефект като  
`<тяло> while (<условие>) <тяло>`
- `while (<условие>) <тяло>`  
има същия ефект като  
`do if (<условие>) <тяло> while (<условие>);`
  - стига <условие> да няма странични ефекти...

# Задачи за do/while

1. Проверете дали сред числата  $x_i = n^3 + 5i^2n - 8i$  има кратно на 7
2. Намерете най-малката цифра на  $n$
3. Проверете дали  $n$  е просто число
4. Намерете  $\sqrt{x}$  по метода на Нютон

# Оператор за продължаване

- **continue;**

```
i = 0;
while (i < 10) {
 i++;
 if (i % 2 == 0) continue;
 x += i;
 if (i * i > 50) break;
 y *= x;
}
```

- **ЛОШ СТИЛ!**

```
flag = true; i = 0;
while (i < 10 && flag) {
 i++;
 if (i % 2 != 0) {
 x += i;
 if (i * i > 50)
 flag = false;
 else
 y *= x;
 }
}
```



# Вложени цикли

- ```
int i, j, k;
for (i = 0; i < n; i++ )
  for (j = i; j < n; j++ )
    for (k = j; k > i; k-- ) {
      cout << '(' << i;
      cout << ',' << j;
      cout << ',' << k << ')' << endl;
    }
```

Задачи за вложени цикли

1	3	6	10	15
2	5	9	14	
4	8	13		
7	12			
11				

1	4	9	16	25
2	3	8	15	24
5	6	7	14	23
10	11	12	13	22
17	18	19	20	21

1	3	6	10	15
2	5	9	14	19
4	8	13	18	22
7	12	17	21	24
11	16	20	23	25

$$\sum_{i=1}^n \prod_{j=i+1}^{2n-i} \frac{i-j}{i+j}$$