

Управляващи оператори в C++

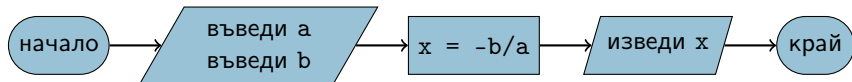
Трифон Трифонов

Увод в програмирането,
спец. Компютърни науки, 1 поток,
спец. Софтуерно инженерство,
2016/17 г.

19 октомври 2016 г.

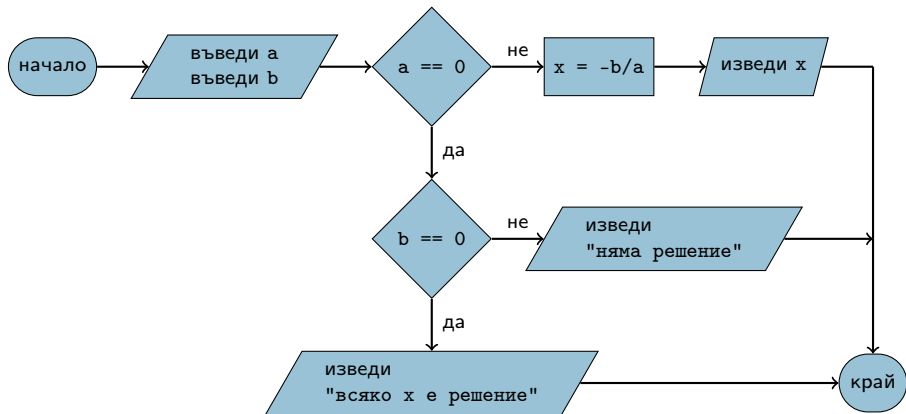
Изчислителни процеси

- Алгоритъм: последователност от стъпки за извършване на пресмятане
- Блок-схема

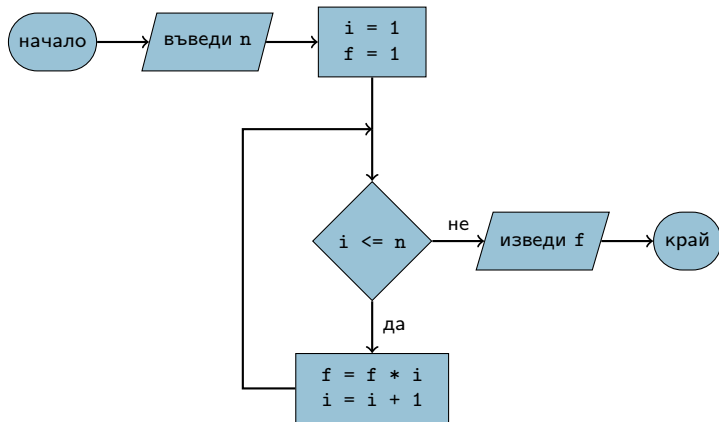


Пример за линеен процес

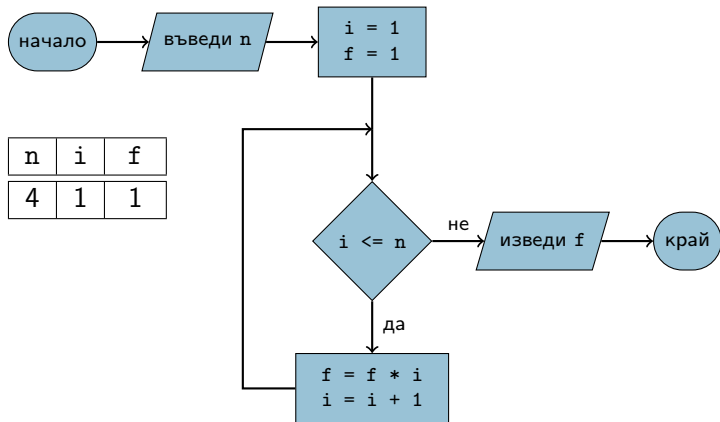
Разклоняващи се процеси



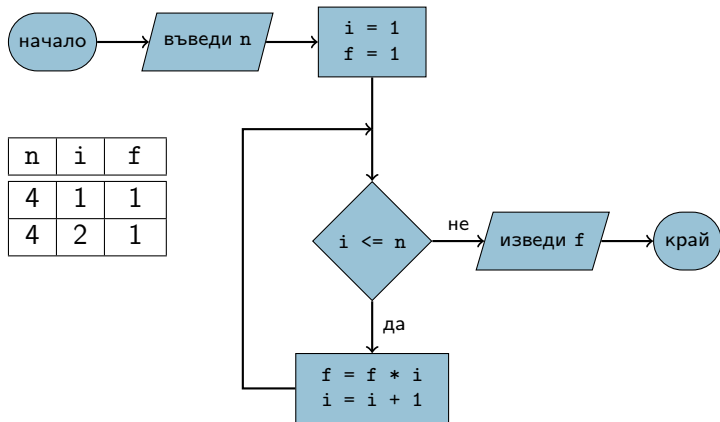
Индуктивни циклични процеси



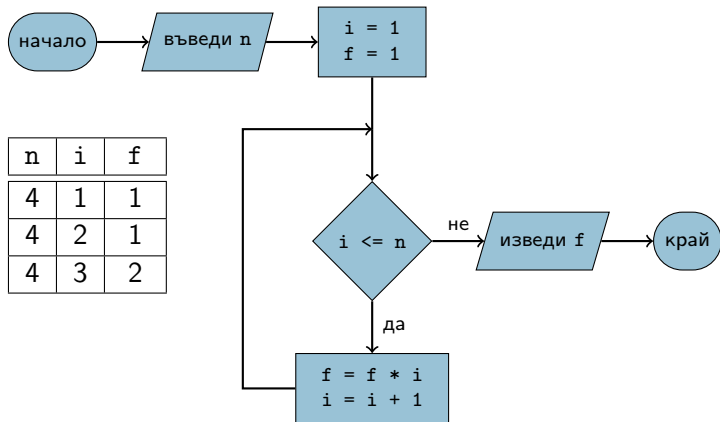
Индуктивни циклични процеси



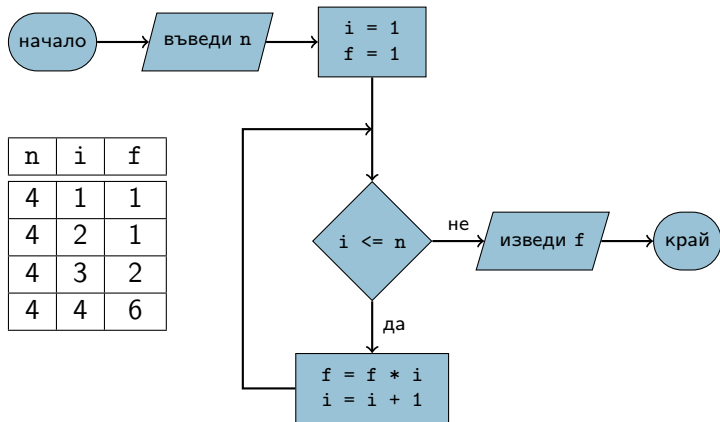
Индуктивни циклични процеси



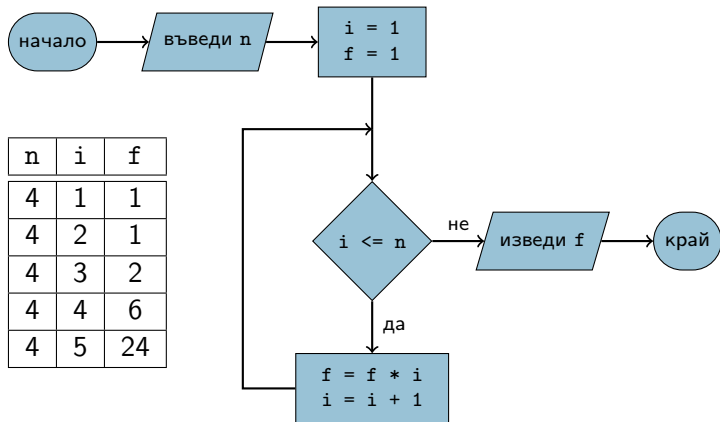
Индуктивни циклични процеси



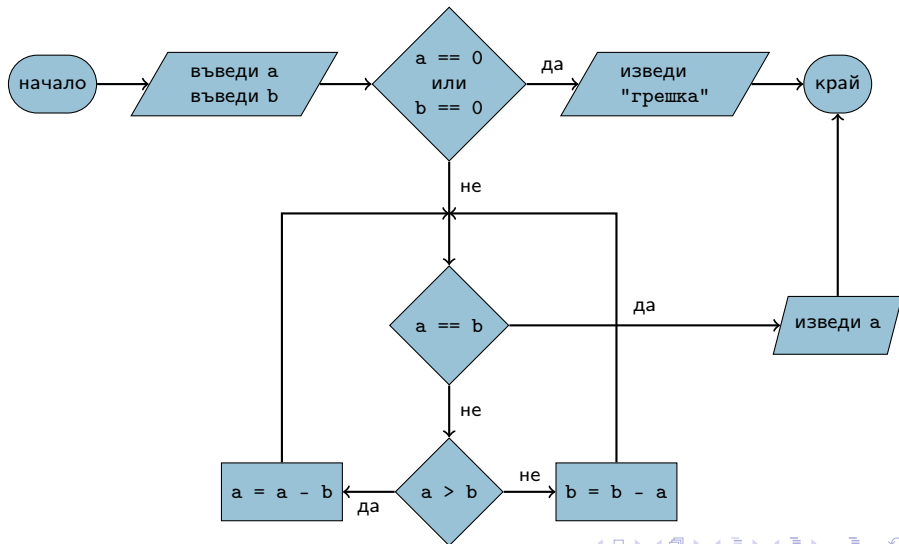
Индуктивни циклични процеси



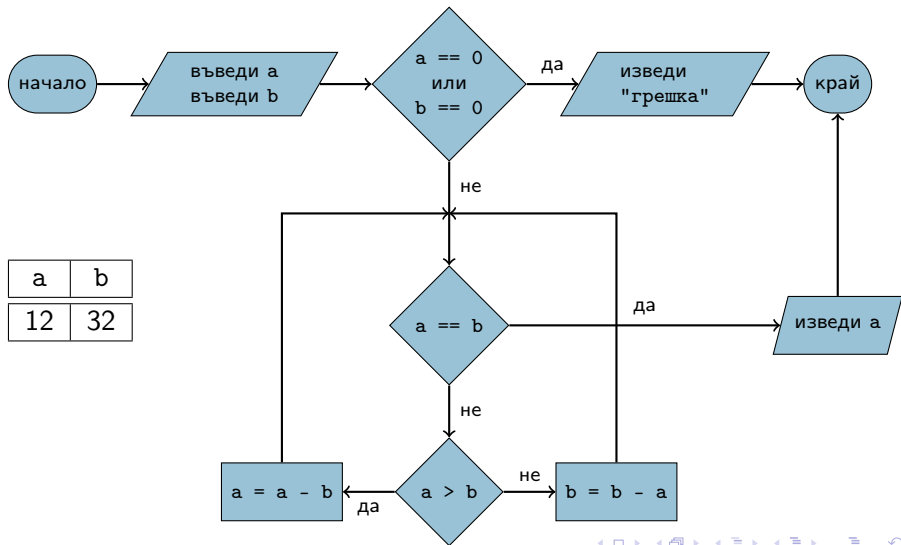
Индуктивни циклични процеси



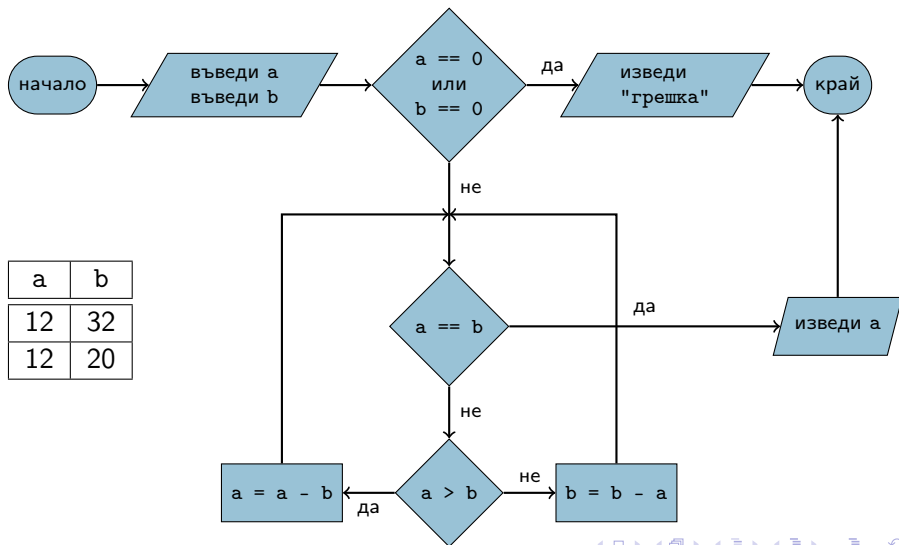
Итеративни циклични процеси



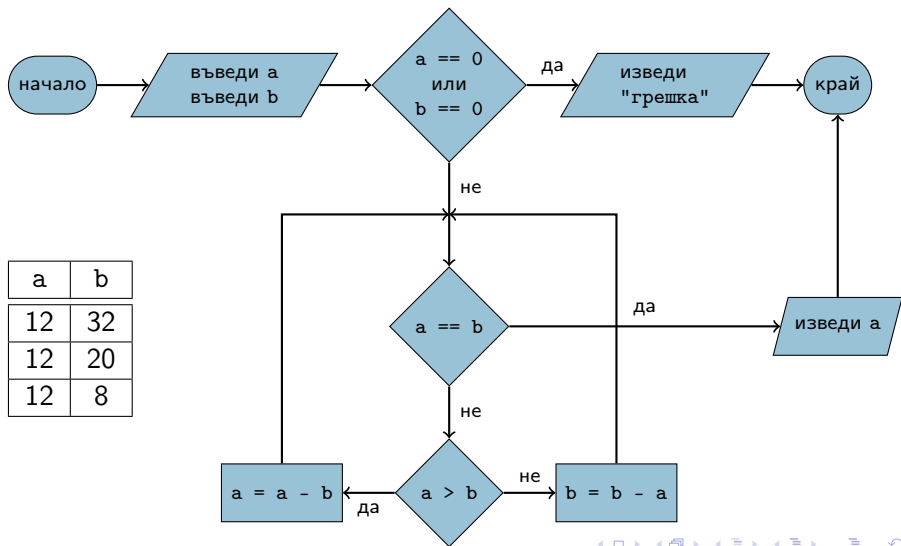
Итеративни циклични процеси



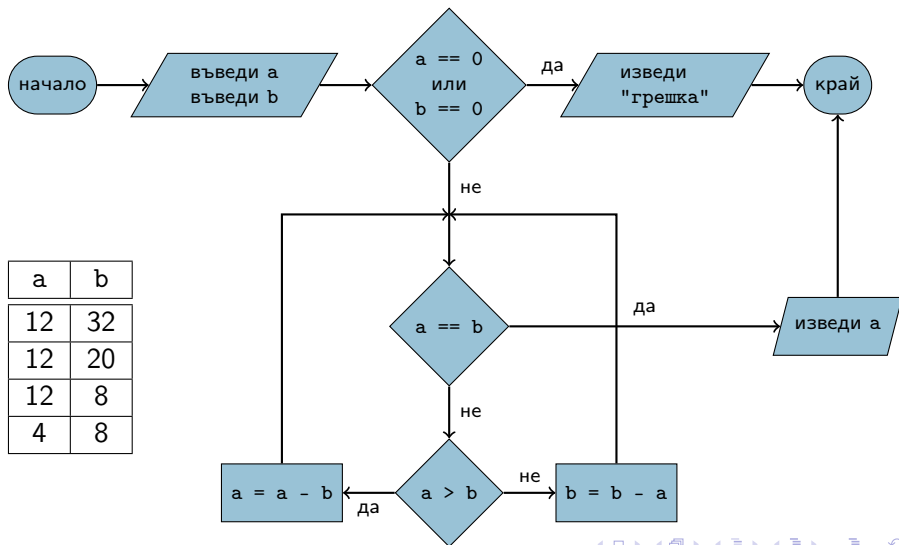
Итеративни циклични процеси



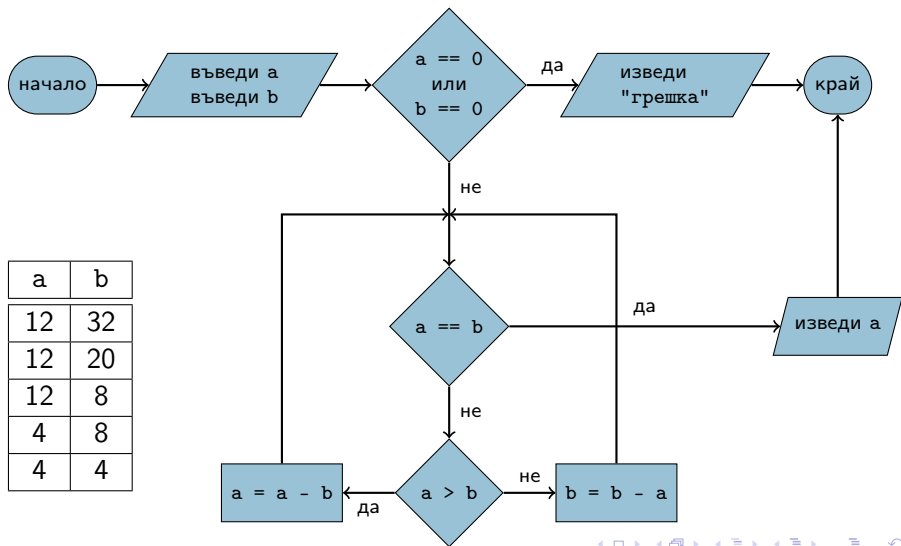
Итеративни циклични процеси



Итеративни циклични процеси



Итеративни циклични процеси



Структурни езици — разклонение

- 1 Въведи a , b
 - 2 Ако $a == 0$, към 5
 - 3 $x = -b / a$
 - 4 Премини към 9
 - 5 Ако $b = 0$, към 8
 - 6 “Няма решения”
 - 7 Премини към 9
 - 8 “Всяко x е решение”
 - 9 Край
- Въведи a , b
 - Ако $a == 0$
 - Ако $b == 0$
 - “Всяко x е решение”
 - Иначе
 - “Няма решения”
 - Иначе
 - $x = -b / a$

Структурни езици — индуктивен цикъл

- 1 Въведи n
- 2 $i = 1$
- 3 $f = 1$
- 4 Ако $i > n$, към 8
- 5 $f = f * i$
- 6 $i = i + 1$
- 7 Премини към 4
- 8 Изведи f
- 9 Край

- Въведи n
- $i = 1$
- $f = 1$
- Повтаряй n пъти
 - $f = f * i$
 - $i = i + 1$
- Изведи f

Структурни езици — итеративен цикъл

- 1 Въведи a , b
 - 2 Ако $a == b$, към 6.
 - 3 Ако $a > b$, към 5.
 - 4 $b = b - a$; към 2.
 - 5 $a = a - b$; към 2.
 - 6 Изведи a
 - 7 Край
- Въведи a , b
 - Докато $a != b$
 - Ако $a > b$
 - $a = a - b$
 - В противен случай
 - $b = b - a$
 - Изведи a

Основни понятия

- **Операция** (operator)
- **Израз** (expression)
- **Оператор/команда** (statement)
- $\langle \text{израз} \rangle ::= \langle \text{константа} \rangle \mid \langle \text{променлива} \rangle \mid$
 $\langle \text{едноместна_операция} \rangle \langle \text{израз} \rangle \mid$
 $\langle \text{израз} \rangle \langle \text{двуместна_операция} \rangle \langle \text{израз} \rangle$
- $\langle \text{оператор} \rangle ::= \langle \text{израз} \rangle ;$

Оператор за присвояване

- `<променлива> = <израз>;`
- `<lvalue> = <rvalue>;`
- `<lvalue>` — място в паметта със стойност, която може да се променя
- Пример: променлива
- `<rvalue>` — временна стойност, без специално място в паметта
- Пример: константа, литерал, резултат от пресмятане
- стандартно преобразуване на типовете:
 `<rvalue>` се преобразува до типа на `<lvalue>`

Присвояването като операция

- **дясноасоциативна** операция

Присвояването като операция

- **дясноасоциативна** операция
- `a = b = c = 2;`

Присвояването като операция

- **дясноасоциативна** операция
- $a = (b = (c = 2));$
- ~~$((a = b) = c) = 2);$~~

Присвояването като операция

- **дясноасоциативна** операция

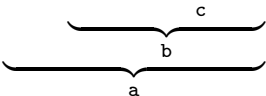
- $a = (b = (c = 2))$

The diagram illustrates the right-associative nature of the assignment operator. It shows the expression $a = (b = (c = 2))$ with three levels of curly braces underneath. The innermost brace is under $(c = 2)$ and labeled 'c'. The middle brace is under $(b = (c = 2))$ and labeled 'b'. The outermost brace is under the entire expression $a = (b = (c = 2))$ and labeled 'a'. This visualizes the order of evaluation from right to left.

Присвояването като операция

- **дясноасоциативна** операция

- $a = (b = (c = 2))$



- Пример: `cout << x + (b = 2);`

Присвояването като операция

- **дясноасоциативна** операция

- $a = (b = (c = 2))$

The diagram illustrates the right-associative evaluation of the expression $a = (b = (c = 2))$. It shows three nested curly braces: the innermost brace is under $(c = 2)$ and labeled 'c'; the middle brace is under $(b = (c = 2))$ and labeled 'b'; and the outermost brace is under the entire expression $a = (b = (c = 2))$ and labeled 'a'.

- Пример: `cout << x + (b = 2);`
- Пример: `(a = b) = a + 3;`

Операция за изброяване

- `<израз1>, <израз2>`
- оценява и двата израза, но крайният резултат е оценката на втория израз
- $a, b, c, d \Leftrightarrow (a, (b, (c, d)))$
- **дясноасоциативна**
- използва се рядко
- Пример: `a = (cout << x, x);`

Съкратени оператори за присвояване

- $a = a + 2 \Leftrightarrow a += 2$

Съкратени оператори за присвояване

- $a = a + 2 \Leftrightarrow a += 2$
- $--, *=, /=, \% =$

Съкратени оператори за присвояване

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

Съкратени оператори за присвояване

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

Съкратени оператори за присвояване

- $a = a + 2 \Leftrightarrow a += 2$
- $--, *=, /=, %=$
- $a = a + 1 \Leftrightarrow ++a$
- $a = a - 1 \Leftrightarrow --a$
- $a++$ увеличава a с 1, но връща предишната стойност на a

Съкратени оператори за присвояване

- $a = a + 2 \Leftrightarrow a += 2$
- $-=$, $*=$, $/=$, $\%=$
- $a = a + 1 \Leftrightarrow ++a$
- $a = a - 1 \Leftrightarrow --a$
- $a++$ увеличава a с 1, но връща предишната стойност на a
 - $a++ \Leftrightarrow (a = (tmp = a) + 1, tmp)$

Съкратени оператори за присвояване

- $a = a + 2 \Leftrightarrow a += 2$
- $-=$, $*=$, $/=$, $\%=$
- $a = a + 1 \Leftrightarrow ++a$
- $a = a - 1 \Leftrightarrow --a$
- $a++$ увеличава a с 1, но връща предишната стойност на a
 - $a++ \Leftrightarrow (a = (tmp = a) + 1, tmp)$
- $a--$ действа аналогично

Съкратени оператори за присвояване

- $a = a + 2 \Leftrightarrow a += 2$
- $-=$, $*=$, $/=$, $\%=$
- $a = a + 1 \Leftrightarrow ++a$
- $a = a - 1 \Leftrightarrow --a$
- $a++$ увеличава a с 1, но връща предишната стойност на a
 - $a++ \Leftrightarrow (a = (tmp = a) + 1, tmp)$
- $a--$ действа аналогично
- $a++$ връща a , което е `<lvalue>`

Съкратени оператори за присвояване

- $a = a + 2 \Leftrightarrow a += 2$
- $-=$, $*=$, $/=$, $\%=$
- $a = a + 1 \Leftrightarrow ++a$
- $a = a - 1 \Leftrightarrow --a$
- $a++$ увеличава a с 1, но връща предишната стойност на a
 - $a++ \Leftrightarrow (a = (tmp = a) + 1, tmp)$
- $a--$ действа аналогично
- $a++$ връща a , което е **<lvalue>**
 - Пример: $++a += 5;$

Съкратени оператори за присвояване

- $a = a + 2 \Leftrightarrow a += 2$
- $-=$, $*=$, $/=$, $\%=$
- $a = a + 1 \Leftrightarrow ++a$
- $a = a - 1 \Leftrightarrow --a$
- $a++$ увеличава a с 1, но връща предишната стойност на a
 - $a++ \Leftrightarrow (a = (tmp = a) + 1, tmp)$
- $a--$ действа аналогично
- $a++$ връща a , което е **<lvalue>**
 - Пример: $++a += 5;$
- $a--$ връща предишната стойност на a , което е **<rvalue>**

Съкратени оператори за присвояване

- $a = a + 2 \Leftrightarrow a += 2$
- $-=$, $*=$, $/=$, $\%=$
- $a = a + 1 \Leftrightarrow ++a$
- $a = a - 1 \Leftrightarrow --a$
- $a++$ увеличава a с 1, но връща предишната стойност на a
 - $a++ \Leftrightarrow (a = (tmp = a) + 1, tmp)$
- $a--$ действа аналогично
- $a++$ връща a , което е **<lvalue>**
 - Пример: `++a += 5;`
- $a--$ връща предишната стойност на a , което е **<rvalue>**
 - Пример: `x = a++ * b;` ~~`a++ += 5;`~~

Оператор за блок

- { { <оператор> } }

Оператор за блок

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

Оператор за блок

- { { <оператор> } }
- { <оператор₁> <оператор₂> ... <оператор_n> }
- Вложени блокове

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

Празен оператор

- ;
- ; ⇔ {}
- няма никакъв ефект

Условен оператор

- `if (<израз>) <оператор> [else <оператор>]`

Условен оператор

- `if (<израз>) <оператор> [else <оператор>]`
- Съкратената форма \Leftrightarrow пълна форма с празен оператор

Условен оператор

- `if (<израз>) <оператор> [else <оператор>]`
- Съкратената форма \Leftrightarrow пълна форма с празен оператор
 - `if (A) X; \Leftrightarrow if (A) X; else;`

Условен оператор

- `if (<израз>) <оператор> [else <оператор>]`
- Съкратената форма \Leftrightarrow пълна форма с празен оператор
 - `if (A) X; \Leftrightarrow if (A) X; else;`
- Пример: `if (x < 2) y = 2;`

Условен оператор

- `if (<израз>) <оператор> [else <оператор>]`
- Съкратената форма \Leftrightarrow пълна форма с празен оператор
 - `if (A) X; \Leftrightarrow if (A) X; else;`
- Пример: `if (x < 2) y = 2;`
- Пример: `if (x > 5) y = 5; else y = 3;`

Условен оператор

- `if (<израз>) <оператор> [else <оператор>]`
- Съкратената форма \Leftrightarrow пълна форма с празен оператор
 - `if (A) X; \Leftrightarrow if (A) X; else;`
- Пример: `if (x < 2) y = 2;`
- Пример: `if (x > 5) y = 5; else y = 3;`
- Представяне на логически операции с вложени if:

Условен оператор

- `if (<израз>) <оператор> [else <оператор>]`
- Съкратената форма \Leftrightarrow пълна форма с празен оператор
 - `if (A) X; \Leftrightarrow if (A) X; else;`
- Пример: `if (x < 2) y = 2;`
- Пример: `if (x > 5) y = 5; else y = 3;`
- Представяне на логически операции с вложени if:
 - `if (!A) X; else Y; \Leftrightarrow if (A) Y; else X;`

Условен оператор

- `if (<израз>) <оператор> [else <оператор>]`
- Съкратената форма \Leftrightarrow пълна форма с празен оператор
 - `if (A) X; \Leftrightarrow if (A) X; else;`
- Пример: `if (x < 2) y = 2;`
- Пример: `if (x > 5) y = 5; else y = 3;`
- Представяне на логически операции с вложени if:
 - `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;`

Условен оператор

- `if (<израз>) <оператор> [else <оператор>]`
- Съкратената форма \Leftrightarrow пълна форма с празен оператор
 - `if (A) X; \Leftrightarrow if (A) X; else;`
- Пример: `if (x < 2) y = 2;`
- Пример: `if (x > 5) y = 5; else y = 3;`
- Представяне на логически операции с вложени if:
 - `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;`