

Основы алгоритмизации и программирования. Условный оператор и логическое выражение

Лекция 3

Привезенцев Д.Г.

Муромский институт Владимирского государственного университета
Очная форма обучения

30 сентября 2021 г.

Разветвляющийся алгоритм

Разветвляющимся называется такой алгоритм, в котором выбирается один из нескольких возможных вариантов вычислительного процесса. Каждый подобный путь называется ветвью алгоритма.

Признаком разветвляющегося алгоритма является наличие **операций проверки условия**. Чаще всего для проверки условия используется условный оператор `if`.

Условный оператор в Си (полный)

```
if (Условие)
{
    БлокОпераций1;
}
else
{
    БлокОпераций2;
}
```

В случае полной развилки
если **Условие** истинно, то
выполняется **БлокОпераций1**, иначе
выполняется **БлокОпераций2**.



Условный оператор в Си (неполный)

```
if (Условие)
{
    БлокОпераций1;
}
```

В случае неполной развилки
если **Условие** истинно,
то **БлокОпераций1** выполняется,
если **Условие** ложно,
то **БлокОпераций1** не выполняется.



Условный оператор (пример)

```
#include <stdio.h>
int main()
{
    int k; // объявляем целую переменную k
    printf("k= "); // выводим сообщение
    scanf("%d", &k); // вводим переменную k
    if (k >= 5) // если k>5
        printf("%d >= 5", k); // выводим "ЗНАЧЕНИЕ >= 5"
    else // иначе
        printf("%d < 5", k); // выводим "ЗНАЧЕНИЕ < 5"
    getchar();
    getchar();
    return 0;
}
```

Условный оператор (пример 2)

```
#include <stdio.h>
int main() {
    int key; // объявляем целую переменную key
    printf("Введите номер пункта, 1 или 2: ");
    scanf("%d", &key); // вводим значение переменной key
    if (key == 1) // если key = 1
        printf("\n Выбран первый пункт"); // выводим сообщение
    else if (key == 2) // иначе если key = 2
        printf("\n Выбран второй пункт"); // выводим сообщение
    else // иначе
        printf("\n Первый и второй пункты не выбраны"); // выводим
        сообщение
    getchar();
    getchar();
    return 0;
}
```

Условный оператор (пример 3)

```
#include <stdio.h>
int main()
{
    int key; // объявляем целую переменную key
    printf("Введите номер пункта, 1 или 2: ");
    scanf("%d", &key); // вводим значение переменной key
    if (key != 1) { // если key не равен 1
        if (key == 2) // если key равен 2
            printf("\n Выбран второй пункт"); // вывод сообщения
        // если key - не 1 и не 2, то ничего не выводится
    }
    else // иначе, если key равен 1
        printf("\n Выбран первый пункт"); // вывод сообщения
    getchar();
    getchar();
    return 0;
}
```

Составной оператор

Составной оператор

Если возникает необходимость, чтобы при выполнении одного условия выполнялось несколько команд, следует использовать составной оператор.

```
if (условие)
{
    оператор1;
    оператор2;
    ...
}
```


Составной оператор (пример 1)

```
#include <stdio.h>
int main ()
{
    int t=2, b=7, r=3;
    if (t>b)
    {
```

```
        if (b < r)    r=b;
    }
    else r=t;
    return 0;
}
```

Сложное логическое выражение

Для сложных условий, когда необходимо проверить одновременно несколько условий, используются три логических оператора:

Сложное логическое выражение

Для сложных условий, когда необходимо проверить одновременно несколько условий, используются три логических оператора:

- ИЛИ ($||$),

Сложное логическое выражение

Для сложных условий, когда необходимо проверить одновременно несколько условий, используются три логических оператора:

- ИЛИ (||),
- И (&&)

Сложное логическое выражение

Для сложных условий, когда необходимо проверить одновременно несколько условий, используются три логических оператора:

- ИЛИ (`||`),
- И (`&&`)
- и отрицания (`!`).

Сложное логическое выражение

Для сложных условий, когда необходимо проверить одновременно несколько условий, используются три логических оператора:

- ИЛИ (||),
- И (&&)
- и отрицания (!).

Оператор **ИЛИ** означает, что для выполнения оператора if достаточно истинности **одного из двух или одновременно обоих** условий.

Сложное логическое выражение

Для сложных условий, когда необходимо проверить одновременно несколько условий, используются три логических оператора:

- ИЛИ (||),
- И (&&)
- и отрицания (!).

Оператор **ИЛИ** означает, что для выполнения оператора if достаточно истинности **одного из двух или одновременно обоих** условий.

Оператор **И** означает, что должны быть истинными **оба условия одновременно**.

Сложное логическое выражение

Для сложных условий, когда необходимо проверить одновременно несколько условий, используются три логических оператора:

- ИЛИ (||),
- И (&&)
- и отрицания (!).

Оператор **ИЛИ** означает, что для выполнения оператора if достаточно истинности **одного из двух или одновременно обоих** условий.

Оператор **И** означает, что должны быть истинными **оба условия одновременно**.

Оператор **отрицания** означает, что оператор if **выполняется**, если некое **условие оказалось ложным**.