

## **Модуль 2.3. Условные операторы**

# Темы модуля

---

- ? Условные операторы if, if–else
- ? Тернарный оператор ?
- ? оператор выбора switch–case–default
- ? Вложенные условные операторы



# Оператор `if`

---

Общая форма оператора `if` следующая

```
if (выражение) оператор;  
else оператор;
```

Примеры:

```
if (x<0) y=-x;
```

```
else y=x;
```

```
if ((x<0) && (y>5)) z=y-x;
```

```
else z=y+x;
```

```
if (x<0) || (y>5) z=y-x;
```

```
else z=y+x;
```

---



---

```
#include <stdio.h>
#include <stdlib.h>
int main(void) {
int magic; /* магическое число */
int guess; /* попытка игрока */
magic = rand(); /* генерация магического числа
*/ printf("Угадай магическое число: ");
scanf("%d", &guess);
if(guess == magic) printf("** Верно **");
else printf("Неверно");
return 0;
}
```

---



# Вложенные условные операторы if

---

- ? Оператор if является вложенным, если он вложен, т.е. находится внутри другого оператора if или else.
- ? Во вложенном условном операторе фраза else всегда ассоциирована с ближайшим if в том же блоке, если этот if не ассоциирован с другой фразой else.

Например:

```
if(i) {  
    if(j) statement 1;  
    if(k) statement 2; /* этот if */  
    else statement 3; /* ассоциирован с этим else  
*/  
}  
else statement 4; /* ассоциирован с if(i) */
```



# Лестница *if-else-if*

---

В программах часто используется конструкция, которую называют *лестницей if-else-if*.

Общая форма лестницы имеет вид

```
if (выражение) оператор;  
else if (выражение) оператор;  
else if (выражение) оператор;  
.  
.  
.  
else оператор;
```

Условные выражения операторов *if* вычисляются сверху вниз. После выполнения некоторого условия, т.е. когда встретится выражение, принимающее значение **ИСТИНА**, выполняется ассоциированный с этим выражением оператор, а оставшаяся часть лестницы пропускается. Если все условия ложны, то выполняется оператор в последней фразе *else*, а если последняя фраза *else* отсутствует, то в этом случае не выполняется ни один оператор

---



# Оператор "?", альтернативный условному

---

Оператор ? можно использовать вместо оператора if-else, записанного в форме

*if (условие) переменная = выражение; else переменная = выражение;*

Оператор ? является **тернарным**, потому что он имеет три операнда.

Его общая форма следующая:

*Выражение1 ? Выражение2 : Выражение3;*

Результат операции ? определяется следующим образом. Сначала вычисляется *Выражение1*. Если оно имеет значение ИСТИНА, вычисляется *Выражение2* и его значение становится результатом операции ?. Если *Выражение1* имеет значение ЛОЖЬ, вычисляется *Выражение3* и его значение становится результатом операции ?.

Например:

`x = 10; y = x > 9 ? 100 : 200;`

---



# Оператор выбора - switch

---

Оператор выбора switch (часто его называют переключателем) предназначен для выбора ветви вычислительного процесса исходя из значения управляющего выражения. (При этом значение управляющего выражения сравнивается со значениями в списке целых или символьных констант. Если будет найдено совпадение, то выполнится ассоциированный с совпавшей константой оператор.)

Общая форма оператора switch следующая:

```
switch (выражение)  
{  
  case постоянная1: последовательность операторов;  
    break;  
  case постоянная2: последовательность операторов;  
    break;  
  case постоянная3: последовательность операторов;  
    break;  
  default: последовательность операторов;  
}
```



---

? Оператор `break` — это один из операторов безусловного перехода. Он может применяться не только в операторе `switch`, но и в циклах. Когда в теле оператора `switch` встречается оператор `break`, программа выходит из оператора `switch` и выполняет оператор, следующий за фигурной скобкой `}` оператора `switch`.

Об операторе `switch` очень важно помнить следующее:

- ? Оператор `switch` отличается от `if` тем, что в нем управляющее выражение проверяется только на равенство с постоянными
- ? В одном и том же операторе `switch` никакие два оператора `case` не могут иметь равных постоянных




# Пример

---

```
void menu(void)
{
char ch; printf("1. Проверка правописания\n");
printf("2. Коррекция ошибок\n");
printf("3. Вывод ошибок\n");
printf("Для пропуска нажмите любую
клавишу\n");
printf(" Введите Ваш выбор: ");
ch = getchar(); /* чтение клавиш */
switch(ch)
{
case '1': check_spelling(); break;
case '2': correct_errors(); break;
case '3': display_errors(); break;
default : printf("Ни выбрана ниодна
опция");
}
}
```

---



# Использование кириллицы

---

Текст набираемый в среде Visual Studio сохраняется и компилируется в кодировке Win-1251, а в консоли выводится в кодировке OEM-866 Для решения данной проблемы необходимо установить соответствующую локаль - `setlocale(LC_ALL, "rus")`

```
#include "stdafx.h"
#include "conio.h"
#include "locale.h" // заголовочный файл, подключающий функции
    локализации
int main()
{
int a[10];
setlocale(LC_ALL, "rus"); // функция локализации, задает значения
    некоторых параметров, чувствительных к геополитическому окружению, в
    котором вып-ся программа макрос LC_ALL относится ко всем категориям
    локализации
printf(«Теперь можно использовать кириллицу\n");
getch();
return 0;
}
```

---

