

Язык программирования С

В.Г.Тетерин – Microsoft Solution Developer (Visual C++) teterin@specialist.ru

www.specialist.ru



МОДУЛЬ 2 СТРУКТУРА ПРОГРАММЫ

Модуль 2. Структура программы

- Использование библиотечных функций
- Ввод с клавиатуры и вывод на экран
- Написание простых программ
- Компиляция, компоновка и выполнение программ

Использование библиотечных функций

- Для вызова функций стандартной библиотеки языка С требуется подключение соответствующих заголовочных файлов при помощи директивы препроцессора #include
- Некоторые стандартные заголовочные файлы:

```
imits.h> - системно-зависимые значения
<float.h> - системно-зависимые значения
<stdio.h> - функции ввода / вывода
<stdlib.h> - функции стандартной библиотеки
<math.h> - математические функции
<ctype.h> - операции с символами
<string.h> - операции со строками
<time.h> - операции с датами и временем
<stdarg.h> - списки аргументов переменной длины
<locale.h> - локализация приложений

<conio.h> - нестандартные функции работы с консолью
```

- Ввод/вывод символа:
- int getchar(void); ввод очередного символа из входного потока;
- int putchar(int ch); вывод символа ch в выходной поток;
 - Замечание: Символ традиционно описывается как int, но используется лишь младший байт
 - Обе функции возвращают введенный/выведенный символ.
 - Если функции возвращают значение EOF это свидетельствует о том, что прочитан признак конца файла при вводе или произошла ошибка ввода/вывода.
 - Определение **EOF** содержится в **stdio.h**.

- Функции семейства ...printf это универсальные функции вывода в языке С.
 - Способ преобразования, форматирования и печати каждого их аргумента полностью управляется форматной строкой.
- Форматная строка это символьная строка, содержащая два типа объектов:
 - обычные символы, которые просто копируются в выходной поток;
 - спецификации формата (начинаются с символа %).
- Каждая спецификация задается в форме
 - %[флаги][ширина][.точность][F|N|h|1]вид_преобразования
 - Все компоненты, заключенные в квадратные скобки, являются необязательными (опциями).
 - Скобки использованы для наглядности, но в спецификации не применяются.

• Символы, определяющие вид преобразования ...printf

8	Симе	вол Тип аргумента	Формат вывода
2	d	целый	целое десятичное со знаком
	i	целый	целое десятичное со знаком
•	u	целый	целое десятичное без знака
•	0	целый	целое восьмеричное без знака
•	X	целый	целое шестнадцатеричное без знака с символами а, b, c, d, e, f
•	X	целый	тот же, что и для X, но с символами A, B, C, D, E, F
	f		avavavva a dagua C
		с плавающей точкой	значение в форме [-]dd.ddddd
	е	с плавающей точкой	значение в форме [-] d.dddddd e [+/-] ddd
•	g	с плавающей точкой	значение в форме е или f в зависимости от его величины и точности; незначащие нули и десятичная точка печатаются только при необходимости
•	Ε	с плавающей точкой	тот же, что и для е, но экспонента обозначается символом Е
•	G	с плавающей точкой	тот же, что и для $ {f g} $, но с символом $ {f E} $ для экспоненты
•			
•	С	символ	отдельный символ
٠.	S	указатель на строку	печатаются символы строки, пока не встретится символ '\0'
•	%	(нет)	печатается символ %
•	n	указатель на int	по указанному адресу запоминается количество напечатанных к этому моменту символов
•	р	указатель	аргумент печатается как указатель

• Спецификатор ширины указывает минимальную ширину поля вывода

	Значение	Воздействие на формат вывода
•	n	печатать по крайней мере n символов, если выводимое значение содержит менее n символов, то поле дополняется пробелами;
•	0n	печатать по крайней мере n символов, дополняя поле слева нулями, если выводимых символов менее n ;
•	*	значение ширины поля является аргументом функцииprintf, предшествующим в списке аргументов выводимому значению

• Спецификатор точности определяет количество печатаемых символов

Значение	Воздействие на формат вывода		
не указано	- устанавливается значение точности по умолчанию :		
	1 - для d, i, o, u, x, X , 6 - для e, E, f		
	все значащие цифры - для ${f g}$, ${f G}$, все символы до первого ' ${f \setminus} {f 0}$ ' - для ${f S}$		
.0	- для e , E , f - подавляется печать десятичной точки		
. n	- для d , i , o , u , x , x - печатается по крайней мере n символов; если фактическое значение содержит менее n символов, то оно слева дополняется до n символов нулями; если в фактическом значении более n символов, то они печатаются полносты без какого-либо отсечения		
	- для e , E , f - печатается n цифр после десятичной точки; перед отбрасыванием лишних разрядов производится округление		
	- для g , G - печатается не более n значащих цифр		
	- для s - печатается не более n символов		

Флаги для ...printf

Флаг	Воздействие на формат вывода		
-	- выравнивание результата по левому краю поля и заполнение справа оставшейся части поля пробелами; - если не указан, то выравнивание по правому краю поля и заполнение слева оставшейся части поля пробелами или нулями		
+	- результат преобразований d , i , e , g , E , G всегда печатается с его знаком: плюс (+) или минус (-)		
пробел	- для положительных чисел вместо знака выводится символ пробела		
#	- предписывает вывод результата преобразования в "альтернативной" форме:		
	для c , s , d , i , u - не оказывает действия		
	для о - вывод 0 перед ненулевым результатом		
	для x , X - вывод соответственно 0x или 0X перед результатом		
	для e , E , f - всегда выводить десятичную точку		
	для g , G - как и для e и E и без подавления незначащих нулей		

Замечание. Флаги могут указываться в любых комбинациях и в любом порядке.
 Флаг (+) имеет больший приоритет, чем флаг "Пробел", если они присутствуют одновременно.

- Функции семейства ...scanf универсальные функции ввода в языке Си.
 - Единицей вводимой и обрабатываемой ими информации является **входное поле**.
 - Порядок следования полей, их содержимое и способ обработки полностью определяются форматной строкой.
- Форматная строка это символьная строка, содержащая три типа объектов:
 - "пробельные" символы
 - пробел, символы табуляции и новой строки;
 - "непробельные" символы
 - любые символы ASCII за исключением "пробельных" и символа процента %;
 - спецификации формата
 - начинаются с символа %.
- "Пробельные" символы являются разделителями полей в потоке данных.
 - Встречая "пробельный" символ в форматной строке, функция ...scanf будет читать и отбрасывать все подряд идущие "пробельные" символы во входном потоке вплоть до первого "непробельного".
- Встречая "непробельный" символ в форматной строке, функция ... scanf должна прочитать и отбросить такой же символ во входном потоке,
 - а при обнаружении их несовпадения прекратить свою работу.

- Спецификации формата имеют следующую форму :
 - %[*][ширина][F|N][h|l|L]вид_преобразования
 - Все компоненты, заключенные здесь в квадратные скобки (которые использованы для наглядности, но в спецификации не применяются!), являются необязательными (опциями)
 - * признак подавления присваивания (пропуск поля);
 - **ширина** максимальная ширина поля, но его сканирование может закончиться и раньше, если встретится "пробельный" или не преобразуемый по данному формату символ.
 - **F | N** аргумент-адрес есть соответственно **far**-указатель (**F**) или **near**-указатель (**N**), применяются для "перекрытия" значения, принятого по умолчанию;
 - h | 1 аргумент-адрес есть указатель на short (h) или long (l) соответственно, используются для "перекрытия" умолчания;
 - L аргумент-адрес есть указатель на long double.

• Символы, определяющие вид преобразования для • • • scanf

• Символы Вводим	иое значение Тип аргум	иента-адреса
• d	десятичное целое	int *arg
• 0	восьмеричное целое	тот же
• i	10-е, 8-е или 16-е целое	тот же
• u	10-е целое без знака	unsigned int*arg
• x x	16-е целое	int*arg
• fegEG	с плавающей точкой	float*arg
• S	символьная строка	char arg[]
• c	символ	char *arg или char arg[W+1], если задана ширина поля W
• %	символ процента	вводится символ процента
• n	(нет)	int *arg, значением arg станет число успешн введенных до %n символов
• p	16-е YYYY: XXXX или XXXX	far *arg или near *arg, используется форма, принятая по умолчанию в зависимости о модели памяти.

Для ввода строки можно задавать в квадратных скобках поисковый образ, например,
 %[0-9A-F+-].

- В этом случае функция ...scanf для текущего аргумента будет вводить поле, содержащее только перечисленные символы. Здесь квадратные скобки обязательны.
- Наоборот, спецификация

• предписывает вводить любые символы кроме перечисленных.

- Функции ...scanf прекращают свое выполнение в следующих ситуациях :
 - 1. Очередной символ во входном поле не совпадает с очередным "непробельным" символом в форматной строке.
 - 2. Прочитан признак конца файла EOF.
 - 3. Исчерпана форматная строка.
 - Все функции ...Scanf возвращают количество присвоенных значений (без учета пропускаемых полей) или EOF при чтении конца файла (конца строки для SScanf).
 - Замечание. Если в обращении к функции . . . Scanf количество аргументов-адресов больше, чем число спецификаций преобразования, заданных в форматной строке, то лишние аргументы игнорируются.
 - Но если их меньше, чем спецификаций это ошибка с непредсказуемыми последствиями.
- *Пример*. Ввод даты в форме 23-feb-89 с выделением ее составных частей числа, месяца и года.
 - Предшествующие "пробельные" символы и внутренние символы тире отбрасываются.
- scanf(" %2d-%3s-%2d", &day, &month, &year);

- Некоторые математические функции
 - 1. Вычисление абсолютного значения (модуля)
 - int abs(int x); long labs(long x); double fabs(double x);
 - 2. Определение ближайшего целого, целой и дробной частей

```
    double ceil( double x );
    double floor( double x );
    - наибольшее приведенное к double целое <=x.</li>
```

- double fmod(double x, double y);
 остаток от деления х на у.
- 3. Степенные и логарифмические функции

```
double sqrt( double x);
double pow( double x, double y );
double pow10( int x );
double exp( double x );
double log( double x );
double log( double x );
double log10( double x );
десятичный логарифм х.
```

4. Тригонометрические и обратные тригонометрические функции

```
    double sin( double x );
    double cos( double x );
    double tan( double x );
    double atan( double x );
    double atan2( double y, double x );
    rge atan2( y,x ) == atan(y/x).
```

- Некоторые стандартные функции
 - int rand(void); алгоритм генерации случайных чисел с периодом pow(2,32),
 возвращает число в диапазоне от 0 до RAND_MAX.
 - void srand(unsigned seed);
 устанавливает начальное состояние генератора.
 - void exit(int status);
 завершает процесс выполнения программы,
 - предварительно опустошает все буферы выходных потоков, закрывает все файлы и выполняет все функции завершения;
 - передает **status** в качестве кода завершения процесса (обычно **0** соответствует нормальному завершению, а ненулевое значение какой-нибудь ошибке).
 - int system(const char *command); вызывает командный процессор для исполнения команды command, возвращает статус завершения.

- 6. Функции преобразования и проверки типа символа в коде ASCII
 - Преобразования:
 - int tolower(int c); переводит символ с на нижний регистр.
 - int toupper(int c); переводит символ с на верхний регистр.
 - int toascii(int c); переводит символ с в код ASCII, очищая все биты, кроме младших 7 бит.
 - Все они возвращают преобразованный символ.
 - Предикаты проверки символов имеют формат :
 - int is...(int ch);
 - и возвращают значение 0 (ложь) или != 0 (истина).
 - Ниже приведены лишь их имена и в скобках проверяемые ими условия:

```
isalpha - ('A'-'z', 'a'-'z')
islower - ('a'-'z')
isalnum - ('A'-'z', 'a'-'z', '0'-'9')
isdigit - ('0'-'9')
isascii - (0x00 - 0x7F)
isgraph - (0x21 - 0x7E)
isspace - (0x09 - 0x0D, 0x20)
isupper - ('A'-'z')
isalnum - ('A'-'z', 'a'-'z', '0'-'9')
isxdigit - ('0'-'9', 'A'-'F', 'a'-'f')
isprint - (0x20 - 0x7E)
iscntrl - (0x7F или 0x00 - 0x1F)
ispunct - (iscntrl или isspace)
```

Итоги

- В этом модуле Вы изучили темы:
 - Использование библиотечных функций
 - Ввод с клавиатуры и вывод на экран
 - Написание простых программ
 - Компиляция, компоновка и выполнение программ

Вопросы?

В.Г.Тетерин – Microsoft Solution Developer (Visual C++)

teterin@specialist.ru

www.specialist.ru



ПРИЛОЖЕНИЕ ЗАДАЧИ

Задачи

- 1. Введите экзаменационные оценки студента по 3 предметам и вычислите средний балл
- 2. Вычислить сумму, разность, произведение и частное двух значений x1 и x2
- 3. Вычислить сумму, разность, произведение и частное двух натуральных дробей (a/b) и (c/d). Результаты печатать в виде натуральной дроби.
- 4. Вычислить сумму цифр четырехзначного числа
- Целое четырехзначное число напечатать вразрядку (с пробелами между цифрами)
- 6. Целое четырехзначное число заменить числом, получающимся при записи его цифр в обратном порядке

Задачи

- 7. Ввести время в формате чч:мм:сс, перевести в секунды
- 8. Ввести время в секундах, перевести в формат чч:мм:сс
- 9. Напечатать длину интервала между двумя моментами времени в пределах суток (все формате чч:мм:сс)