

# Приемы программирования на языке C

## Содержание

1 Ресурсы по языку Си	1
2 Вводные замечания	1
3 Установка MSYS2 и MinGW-W64 для ОС Windows	2
4 Visual Studio Code как среда разработки для языка Си	2
5 Алфавит, идентификаторы, служебные слова	3
5.1 Константы и строки . . . . .	3
5.2 Переменные и именованные константы . . . . .	4
Список литературы	4
Список листингов	4

## 1. Ресурсы по языку Си

<https://learnc.info/c/>

## 2. Вводные замечания

Язык Си – это компилируемый язык программирования высокого уровня, кроссплатформенный, позволяющий создавать программы которые будут работать во всех операционных системах, но для каждой операционной системы компиляцию нужно выполнять отдельно.

Существует несколько стандартов языка Си: C90 (ANSI C/ISO C), C99 и C11. Для того чтобы использовать правила конкретного стандарта, нужно в составе команды компиляции указать следующие флаги: `-std=c90`, `-std=c99` или `-std=c11`. Современный язык Си включает возможности стандарта C11.

Узнать используемый стандарт языка Си внутри программы можно с помощью *макроса* `__STDC_VERSION__`

```
printf("%ld\n", __STDC_VERSION__); // 201112
```

Получить информацию о версии компилятора позволяет макрос `__VERSION__`

```
printf("%s\n", __VERSION__); // Apple LLVM 12.0.0 (clang-1200.0.32.2)
```

Когда мы в командной строке вводим название программы без предварительного указания пути к ней, то

- вначале поиск программы выполняется в текущем рабочем каталоге (обычно это каталог, из которого запускается программа),

- а затем в путях, указанных в системной переменной PATH.

Системные каталоги имеют более высокий приоритет, чем каталоги, указанные в переменной PATH.

### 3. Установка MSYS2 и MinGW-W64 для ОС Windows

Установить компилятор gcc на ОС Windows можно следующим образом. Детали процедуры установки можно найти в книге [2, стр. 24]. Предварительно нам нужно установить библиотеку MSYS2. Переходим на сайт <https://www.msys2.org> и скачиваем файл `msys2-x86_64-20230718.exe`, а затем запускаем его.

Библиотека MSYS2 будет установлена в каталог `C:\msys64`. В этом каталоге расположены скрипты для запуска: `msys2.exe`, `mingw32.exe`, `mingw64.exe`.

Файл `msys2.exe` запускает командную строку, в которой мы можем установить различные библиотеки. Сначала обновим программу, выполнив команду

Окно `msys2.exe`

```
$ pacman -Syu
```

Теперь можно установить библиотеку MinGW-W64

```
$ pacman -S mingw-w64-x86_64-toolchain
```

Для установки всех компонентов нажимаем клавишу <Enter>, а затем на запрос подтверждения установки вводим букву Y и нажимаем клавишу <Enter>.

Библиотека MinGW-W64 будет установлена в каталог `C:\msys64\mingw64`. Добавив путь до `C:\msys64\mingw64\bin` в системную переменную Path, можно будет вызывать компилятор gcc из любой точки

```
$ gcc --version
g++.exe (Rev2, Built by MSYS2 project) 13.2.0
...
```

Все установленные библиотеки скомпилированы под 64-битные операционные системы. Для установки 32-битных версий библиотек нужно в команде заменить фрагмент `x86_64` фрагментом `i686`. Пример

```
$ pacman -S mingw-w64-i686-toolchain
```

### 4. Visual Studio Code как среда разработки для языка Си

Скачать Visual Studio Code можно здесь <https://code.visualstudio.com/>. Для ОС Windows нужно еще установить GCC. На ОС Linux компилятор gcc доступен «из коробки». На ОС MacOS компилятор gcc можно установить с помощью утилиты `brew`.

После установки IDE останется только создать директорию с проектом под язык Си. Когда Visual Studio Code увидит файл с расширением `*.c`, она предложит установить специальное расширение «C/C++ Extension Pack v1.X.X».

## 5. Алфавит, идентификаторы, служебные слова

Идентификаторы, начинающиеся с одного символа подчеркивания «\_» или с двух символов подчеркивания «\_\_», зарезервированы для использования в библиотеках и компиляторах. Поэтому такие идентификаторы не рекомендуется выбирать в качестве имен в прикладной программе на языке Си. Рекомендуется при программировании имена констант записывать целиком заглавными буквами [1, стр. 15].

### 5.1. Константы и строки

По определению, константа представляет значение, которое не может быть изменено. Синтаксис языка определяет 5 типов *констант*:

1. символы,
2. константы перечисляемого типа,
3. вещественные числа,
4. целые числа,
5. нулевой указатель («null»-указатель).

Управляющие последовательности («\n», «\r», etc.) являются частным случаем экскейп-последовательностей (ESC-последовательностей), к которым также относятся лексемы вида «\ddd», либо «\xhh».

*Символьная константа* (символ) имеет *целый тип*, то есть символы можно использовать в качестве целочисленных операндов в выражениях.

Целочисленные именованные константы можно вводить с помощью перечисления `enum`. Пример

```
enum DAY {SUNDAY, MONDAY, ...};
enum BOOLEAN {NO, YES};
```

В первой строке `DAY`, а во второй строке `BOOLEAN` это необязательный произвольный идентификатор – название перечисления.

Если в списке нет ни одного элемента со знаком «=», то значения констант начинаются с 0 и увеличиваются на 1 слева направо. Таким образом, `NO` равно 0, а `YES` – 1. Именованная константа со знаком «=» получает соответствующее значение, а следующая за ней именованные константы без явных значений увеличиваются на 1 каждая.

То есть если

```
enum BOOLEAN {NO=10, YES};
printf("NO=%d, YES=%d", NO, YES); // NO=10, YES=11
```

В Python можно сделать так

```
from enum import Enum, auto

class Boolean(Enum):
    NO = 0
    YES = auto()

Boolean.NO.value # 0
Boolean.YES.value # 1
```

Формально строки не относятся к константам языка Си, а представляют собой отдельный тип его лексем. Строковая константа определяется как последовательность символов, заключенных в двойные кавычки (не в апострофы).

Представление *строковых констант* в памяти ЭВМ подчиняются следующим правилам. Все символы строки размещаются подряд, и каждый символ (в том числе представленный эскейп-последовательностью) занимает ровно 1 байт. В конце записи строковой константы компилятор помещает символ `'\0'`.

Таким образом, количество байтов, выделяемое в памяти ЭВМ для представления значения строки, ровно на 1 больше, чем число символов в записи этой строковой константы.

При работе с символьной информацией нужно помнить, что длина символьной константы `'F'` равна 1 байту, а длина строки `"F"` равна 2 байтам.

## 5.2. Переменные и именованные константы

Одним из основных понятий языка Си является *объект* – именованная область памяти. Частный случай объекта – переменная.

Каждый из целочисленных типов (`char`, `short`, `int`, `long`) может быть определен либо как *знаковый* `signed` либо как *беззнаковый* `unsigned` (по умолчанию `signed`).

Различие между этими двумя типами – в правилах интерпретации *старшего бита внутреннего представления*. Спецификатор `signed` означает, что старший бит внутреннего представления воспринимался как знаковый; `unsigned` означает, что старший бит внутреннего представления входит в код представляемого числового значения, которое считается в этом случае беззнаковым. Выбор знакового или беззнакового представления определяет предельные значения, которые можно представить с помощью описанной переменной. Например на IBM PC переменная типа `unsigned int` позволяет представить числа от 0 до 65 535, а переменная типа `signed int` (или просто `int`) соответствуют значения в диапазоне от -32768 до +32767.

Именованные константы можно вводить с помощью *директивы препроцессора* `#define`, например

```
// препроцессорная константа
#define EULER 2.718282 // точка с запятой не нужна!!!
```

Что эквивалентно

```
const double EULER = 2.718282;
```

До начала компиляции текст программы на языке Си обрабатывается специальным компонентом транслятора – *препроцессором*. Далее текст от препроцессора поступает к компилятору. Итак, основное отличие констант, определяемых *препроцессорными директивами* `#define`, состоит в том, что эти *константы вводятся* в текст программы *до этапа компиляции* – препроцессор обрабатывает исходный код программы и делает в этом тексте замены и подстановки [1, стр. 29].

## Список литературы

1. Подбельский В.В., Фомин С.С. Программирование на языке Си, 2005. – 600 с.
2. Прохоренок Н.А. Язык С. Самое необходимое. – СПб.: БХВ-Петербург, 2020. – 480 с.

## Листинги