

Раздел «Язык Си» . ReasonToLearnC :

Зачем изучать язык C?

Прежде, чем перейти к рассказу о языках C и C++, хочется предупредить читателя. Языки C и C++ – чрезвычайно мощное, гибкое средство, с помощью которого можно решать алгоритмические и вычислительные задачи любой сложности, осуществлять моделирование физических систем, создавать распределенные компьютерные системы, разрабатывать приложения, использующие различные сетевые протоколы и взаимодействующие с базами данных, с красивым и удобным графическим интерфейсом.

Языки C и C++ универсальны. В этом их плюс и в этом же их минус.

Надо понимать, что язык, специально созданный для моделирования физических систем может оказаться удобней и адекватнее при решении задачи моделирования, нежели язык C/C++. А языки и технологии, придуманные для разработки Web-приложений, удобнее при разработке Web-приложения.

Языки, созданные специально для решения определенной задачи, очевидно, будут лучше в своей области, нежели какие-нибудь универсальные средства.

В начале XXI века программисты в своем инструментарии имели уже около 1000 различных языков программирования, и каждый из этих языков был достоин внимания при решении задач определенного класса. Языки программирования как различные виды животных в одной экосистеме – некоторые из них похожи, другие совсем не похожи. Экологические ниши некоторых видов пересекаются, ниши других – совсем не связаны друг с другом. Очень часто один язык вытесняет другой, также наблюдается постоянная эволюция языков. Языки отличаются классом эффективно решаемых ими задач и своей парадигмой.

Язык C – это императивный, типизированный, высокоуровневый, машинно-ориентированный, не объектно-ориентированный язык программирования. Язык C++ является его объектно-ориентированным расширением.

Язык C **высокоуровневый**:

- Команды и типы данных, которые заложены в C изначально, элементарны, но на основе них можно создавать свои, сколь угодно сложные функции и структуры данных.
- Стандартная библиотека C включает в себя много сложных (высокоуровневых) функций.
- Он позволяет определять свои типы данных, создавать свои библиотеки. Создано множество инструментов для разработки больших проектов на языке C.

То, что язык C **машинно-ориентированный**, означает следующее:

- Основная информационная единица, с которой работает программист на C – это байт. Это значит, что он знает о том, что представляют данные в его программе на уровне байтов и может "добраться" до любого их байта.
- Программист на C представляет, как его программа будет расположена в памяти.
- Программист на C может перейти на самый низкий уровень программирования – делать в своей программе "ассемблерные вставки" и писать на уровне команд процессора.
- Язык C приспособлен для того, чтобы работать непосредственно с любыми устройствами компьютера.
- На языке C можно писать операционные системы (так многие и делают; впервые это произошло с Linux в 1973).

Прилагательное **императивный** означает, что в языке C логику работы программы представляют в виде глаголов (действий) : "пока выполнено условие ... делать действия ...", "выполнить последовательность действий ... N раз", "если выполнено условие ... перейти к действию ...".

В отличие от C язык Perl высокоуровневый, нетипизированный и скриптовый:

Поиск

Поиск

Раздел «Язык Си»

Главная
Зачем учить C?
Определения

Инструменты:

Поиск
Изменения
Index
Статистика

Разделы

Информация
Алгоритмы
Язык Си
Язык Ruby
Язык
Ассемблера
El Judge
Парадигмы
Образование
Сети
Objective C

Logon>>

- В Perl изначально заложено несколько высокоуровневых команд и структур, например, команда сортировки `sort` или преобразования элементов массива `map`.
- Элементарный тип данных в Perl – это строка символов (в то время как в языке C строка – это массив символов).

Например, на языке Perl можно написать:

```
# Программа: Вывод значений переменных окружения в алфавитном порядке
# Язык Perl.
print map {"$_ = $ENV{$_}\n" } (sort keys %ENV);
```

Эта программа, состоящая из одной строки, печатает названия и значения переменных окружения системы (Environment variables), упорядочив их в алфавитном порядке. По сути в одну строку мы поместили несколько глаголов: извлечь ключи таблицы, отсортировать их, отобразить в вид "ключ = значение" и напечатать.

В языке C нет команды сортировки и нет встроенных Хеш-таблиц, но, как и в большинстве языков, в C можно создавать свои структуры данных и библиотеки функций. Так в стандартной библиотеке языка Си есть функция `qsort` (объявлена в файле `stdio.h`), которой можно пользоваться для сортировки произвольных объектов – не нужно писать алгоритм сортировки самому. Но не смотря на это, код программы на C, печатающей значения переменных окружения, оказывается существенно объемнее и труднее для понимания человеку, не знакомому с языком:

```
/* Программа: Вывод значений переменных окружения в алфавитном порядке.
   Язык C.
*/
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
extern char **environ;
int
cmp (const void *a, const void *b)
{
    return strcmp(*(char**)a, *(char**)b);
}
int
main ()
{
    int i = 0;
    while (environ[i] != NULL) i++;
    qsort (environ, i, sizeof(char*), cmp);
    i = 0;
    while (environ[i] != NULL) {
        printf ("%s\n", environ[i++]);
    }
    return 0;
}
```

Как видите, человеку, не знающему язык C, разобраться в этой программе довольно сложно – схема проста, но детали синтаксиса могут обескуражить.

Область применения языка C ограничена, для разных задач удобно использовать разные языки, часто отличные от C/C++. Например, для простых задач иногда удобнее применять скриптовые языки, подобные Perl, Ruby, Python или bash, а для крупных Web-приложений – Java, Python или Ruby.

Важным моментом в культуре программирования является методология **reuse** (повторное использование кода) – методология обеспечения модульности, то есть умение использовать разработанные инструменты, программы и библиотеки и умение экспортировать разработанную функциональность в виде удобных, подключаемых к другим проектам, модулей. Термин **reuse** означает, в первую очередь, что вы должны знать, какие разработаны средства, которые помогут вам быстро и эффективно решить поставленную задачу, и использовать их.

Результаты своего труда полезно экспортировать как максимально универсальный инструмент (например, библиотеку функций) полезный для решения множества задач,

инструмент, простой и удобный в использовании, снабженный ясной и подробной документацией.

Такие алгоритмы как сортировка и вычисление математических функций, а также популярные структуры данных давно уже написаны – осталось только научиться ими правильно пользоваться. Поэтому с уровня глаголов – "сложить два числа", "обнулить значение переменной", "поставить белую точку на черном экране" -- мы можем перейти на более высокий уровень: "создать окошко и переместить его в верхний левый угол экрана", "найти корни многочлена", "нарисовать график функции", "получить таблицу из базы данных" и т.д.

Программа, печатающая переменные окружения, также уже написана за нас и встроена в командную оболочку операционных систем Linux и MS Windows – это команда `set`.

```
> set
```

Курс по языку Си посвящен программированию на низком уровне, то есть тому, чем вам, возможно, никогда не придется заниматься – для решения своих задач вы, скорее всего, будете пользоваться высокоуровневыми языками программирования и уже готовыми инструментами.

Но, с другой стороны, программировать на высоком уровне и не уметь программировать на низком уровне, это все равно что оперировать с эллиптическими интегралами и при этом не уметь складывать и умножать числа.

Надо понимать, что польза от изучения языка программирования низкого уровня заключается не столько в обретении инструмента для разработки ПО, а обретении понимания устройства компьютера и ОС. Опыт программирования на С развивает **алгоритмическое мышление**, которое стоит в ряду с абстрактным, образным и др. типами мышления, и, в определенном смысле, является основой при решении произвольных конструкторских задач.

Опыт программирования дарит вам новые идеи, обогащает его методами поиска решения, и развивает способность к рефлексии и абстракции:

практика программирования делает мышление ясным

Это цель, ради которой стоит изучать язык С.

(с) Материалы раздела "Язык Си" публикуются под лицензией [GNU Free Documentation License](#).