

Алгоритмический язык программирования Си Тема 1.1 Введение

Язык программирования как средство связи с ЭВМ

Язык программирования (programming language) — формализованный язык для описания и алгоритмов решения задач на ЭВМ. Языки программирования являются искусственными; в них синтаксис и семантика строго определены, они не допускают свободного толкования выражения, что характерно для естественного языка. Языки программирования разделяются на две основные категории языки высокого уровня и языки низкого уровня.

Язык программирования как средство связи с ЭВМ

Язык высокого уровня (high-level language) — язык программирования, средства которого обеспечивают описание задачи в наглядном, легко воспринимаемом виде, удобном для программиста. Он не зависит от внутренних машинных кодов ЭВМ любого типа, поэтому программы, написанные на языках высокого уровня, требуют перевода в машинные коды программами транслятора либо интерпретатора. К языкам высокого уровня относят Фортран, ПЛ/1, Бейсик, Паскаль, Си, Ада. Язык низкого уровня (low-level language) — язык программирования, предназначенный для определенного типа ЭВМ и отражающий его внутренний машинный код.

Виды языков программирования

Различают следующие виды языков программирования:

Алгоритмический язык (algorithmic language) — совокупность символов, соглашений и правил, используемых для однозначного описания алгоритмов и обычно являющаяся часть языка программирования;

Неалгоритмический язык (nonalgorithmic language) — язык программирования, тексты которого не содержат указаний на порядок выполнения операций и служат лишь исходным материалом для синтеза алгоритма решения задачи;

Автономный язык (freestanding language) — специализированный язык высокого уровня, в замкнутых СУБД (СУБД с автономным языком);

Базовый язык (base language) — машинный язык, общий для семейства ЭВМ, а также язык программирования в СУБД с автономным языком;

Гибридный (комбинированный) язык (hybrid language) — язык программирования, использующий также средства другого языка;

И др.

Язык программирования

C++

C++ (англ. C++) — компилируемый строго типизированный язык программирования общего назначения. Поддерживает разные парадигмы программирования: процедурную, обобщённую, функциональную; наибольшее внимание уделено поддержке объектно-ориентированного программирования.

Разработка языка началась в 1979 году. Целью создания C++ было дополнение C возможностями, удобными для масштабной разработки ПО, с сохранением гибкости, скорости и портативности C. Вместе с тем создатели C++ стремились сохранить совместимость с C: синтаксис первого основан на синтаксисе последнего, и большинство программ на C будут работать и как C++. Изначально новый язык назывался “C с классами”, но затем имя было изменено на C++ — это должно было подчеркнуть как его происхождение от C, так и его превосходство над последним.

Трансляция и создание загрузочного модуля

Программа, подготовленная на языке высокого уровня, проходит несколько этапов:

1. этап. В текстовом редакторе пишется **исходный код** программы на алгоритмическом языке (source code) и сохраняется в файле с расширением *.pas.

2 этап. Трансляция, происходит преобразование **исходного кода** программы (source code) в **объектный код**(object code), т.е. происходит проверка синтаксиса написания операторов, и если ошибок в написании нет, осуществляется перевод на язык машинных кодов. Файл объектного кода имеет расширение *.obj;

3. этап. Компоновка, когда происходит обработка объектного кода **редактором связей**, специальной программой осуществляющей построение **загрузочного модуля**(load module), пригодного к выполнению.

Алфавит языка C/C++

Множество символов языка C включает:

прописные буквы латинского алфавита;

строчные буквы латинского алфавита;

арабские цифры;

разделители: , . ; : ? ! ' " | / \ ~ _ ^ () { } [] < > # % & - = + *

Остальные символы могут быть использованы только в символьных строках, символьных константах и комментариях. Язык C++ различает большие и маленькие буквы, таким образом, *name* и *Name* – разные идентификаторы.

Правила написания программы

При написании комментариев следует придерживаться ряда простых истин:

- программы читаются людьми, компьютеру комментарии не нужны;
- всегда необходимы вводные комментарии, в которых указывается назначение программы, ее автор, дата написания и изменения, краткое описание алгоритма, входных и выходных данных, основных переменных и вызываемых функций
- комментарии должны содержать дополнительную информацию, а не перефразировать программу;
- комментарии должны быть расположены так, чтобы программа не была менее наглядной;
- неправильные комментарии хуже, чем их отсутствие.
- Стиль программирования, расположение операторов в строках, использование пробелов, выбор имен переменных и т. д. должны быть направлены на то, чтобы сделать программу более понятной людям, ее читающим.

Структура программы на С

Любая достаточно большая программа на Си состоит из файлов. Файлы транслируются Си-компилятором независимо друг от друга и затем объединяются программой-построителем задач, в результате чего создается файл с программой, готовой к выполнению. Файлы, содержащие тексты Си-программы, называются исходными.

В языке Си исходные файлы бывают двух типов:

заголовочные, или h-файлы

файлы реализации, или Си-файлы

Имена заголовочных файлов имеют расширение ".h". Имена файлов реализации имеют расширения ".c".

Заголовочные файлы содержат только описания. Прежде всего, это прототипы функций. Также в h-файлах описываются имена и типы внешних переменных, константы, новые типы, структуры и т.п. При трансляции заголовочных файлов, как правило, никакие объекты не создаются...

Ключевые зарезервированные слова

Список ключевых слов C++

asm	else	new	this
auto	enum	operator	throw
bool	explicit	private	true
break	export	protected	try
case	extern	public	typedef
catch	false	register	typeid
char	float	reinterpret_cast	typename
class	for	return	union
const	friend	short	unsigned
const_cast	goto	signed	using
continue	if	sizeof	virtual
default	inline	static	void
delete	int	static_cast	volatile
do	long	struct	wchar_t
double	mutable	switch	while
dynamic_cast	namespace	template	

-
- Данные слова являются зарезервированными, а это значит, что их нельзя использовать для инициализации переменных или классов