

Курс «Основы программирования»

Федорук Елена Владимировна ст. преподаватель каф РК-6 МГТУ
им.Н.Э.Баумана

Лекция №3

Синтаксис и семантика языка программирования

Термином **язык** в широком смысле слова обозначают любое средство общения, любую систему символов или знаков для обмена информацией.

Алгоритмический язык — набор символов и система правил образования и истолкования конструкций из этих символов для задания алгоритмов.

Алгоритмический язык для записи программ и данных называют **языком программирования**.

Любой язык, в том числе язык программирования, подчиняется ряду правил. Их принято разделять на правила, определяющие синтаксис языка, и правила, определяющие семантику.

Синтаксис языка — совокупность правил, определяющих допустимые конструкции (слова, предложения) языка, его форму.

Семантика языка — совокупность правил, определяющих смысл синтаксически корректных конструкций языка, его содержание.

Языки программирования относятся к формальным языкам, для которых, в отличие от естественных языков однозначно определены синтаксис и семантика. Описание синтаксиса языка включает определение алфавита и правил построения различных конструкций языка из символов алфавита и более простых конструкций. Для этого обычно используют форму Бэкуса-Наура (БНФ) или синтаксические диаграммы.

Синтаксис языка программирования закладывают в его компилятор. Таким образом, синтаксически корректная программа, написанная на языке программирования, после преобразования ее в последовательность машинных команд обеспечит выполнение ЭВМ требуемых операций.

Характерные черты языка программирования Си

Язык программирования Си был разработан Деннисом Ритчи в начале 1970-х годов для ускорения разработки операционной системы UNIX (**ОС UNIX**). В результате на языке Си написана большая часть ОС UNIX, компилятор Си и практически все прикладные программы для ОС UNIX. Хотя язык Си был задуман как язык системного программирования, это не узко специализированный язык со множеством ограничений. Большой набор операций и типов данных, современное оформление и высокая степень машиннезависимости сделали его привлекательным языком программирования общего назначения.

Первое издание книги Kernigan, Ritchie. The C Programming Language; Prentice Hall, 1978 описывает так называемый **стандарт K&R** языка Си. Американский Национальный Институт Стандартов (ANSI) разработал

формальный **стандарт ANSI языка Си**. Этот стандарт значительно улучшил переносимость программ на языке Си.

Язык Си — компилируемый язык, существенно более быстрый, чем интерпретируемые языки. Он сочетает в себе свойства языков высоко и низкого уровней. Наряду с управляющими операторами и структурами данных, характерными для языков высокого уровня, он содержит много операций, типичных для ассемблеров или машинных языков (в частности — побитовые операции). Кроме того, язык Си обеспечивает косвенную адресацию с использованием указателей для быстрого и эффективного доступа к данным. Сходство языка Си с ассемблером делает относительно простым создание компиляторов Си для разных типов процессоров. Благодаря переносимости и сравнительно малым потерям в скорости выполнения, язык Си в настоящее время используется во многих областях, в которых традиционным было применение ассемблеров.

Программа на Си в основном состоит из набора функций. Понятие функции в языке Си сочетает свойства подпрограмм и процедур — в Си нет различий между ними. С помощью функций достигается модульность программы. Это упрощает отладку и позволяет использовать фрагменты кода в различных программах, избегая дублирования этих фрагментов.

Си — весьма гибкий язык, который может быть использован для программирования решения самых различных задач. Круг этих задач простирается от задач управления аппаратурой компьютера, требующих работы с физической памятью и устройствами ввода-вывода, до деловых и финансовых задач, манипулирующих сложными структурами данных.

Алфавит языка программирования Си

Алфавит языка программирования — множество символов, допустимых при построении конструкций программы.

Алфавит языка Си включает:

- строчные, прописные буквы латинского алфавита (a...z, A...Z) и символ подчеркивания _ (строчные и прописные буквы различаются)
- цифры (0...9)
- специальные знаки: . , ; + - * / = < > % & ! () { } ^ | ? : [] ^ ~ ' " # \
- **служебные слова**: char, short, int, long, float, double, enum, struct, union, signed, unsigned, void, auto, extern, register, static, break, case, continue, default, do, else, for, goto, if, return, switch, while, sizeof, typedef.

Служебные слова, которые также называют ключевыми словами — это словосочетания, считающиеся единым целым и зарезервированные для определенных целей. Их нельзя использовать в программе в другом качестве.

Из символов алфавита в соответствии с правилами синтаксиса строят различные конструкции или лексемы.

Лексема — это единица текста программы, расположенная между символами-разделителями, которая имеет самостоятельный смысл для компилятора и не содержит в себе других лексем.

Лексемами языка Си являются:

- идентификаторы;
- ключевые слова;
- константы;
- знаки операций.

Простейшей из них является конструкция идентификатор.

Идентификатор — это последовательность букв латинского алфавита (включая символ подчеркивания) и цифр, которая обязательно начинается с буквы. Идентификаторы используются как имена переменных, функций и типов данных. Не рекомендуется начинать идентификатор с символа подчеркивания. Идентификатор не должен совпадать ни с одним ключевым словом языка Си.

Пример 1

`NAME1`

`name1`

`Total_5`

`My_name_family`

Идентификатор может быть произвольной длины.

Описательные (наглядные) имена переменных могут значительно улучшить понятность программы и облегчить процесс отладки и модификации программы. Короткие имена переменных не ускорят выполнение программы.

Общая структура программы на языке Си

Программа на языке Си — это набор функций и объявлений данных, содержащийся в одном или более файлах. Одна из этих функций — функция `main()`.

Функция — это именованный фрагмент программы. Данные могут передаваться в функцию и функция может возвращать значение.

Программа на языке Си состоит из функций, произвольным образом упорядоченных в файле. Фактически функции могут быть размещены и в разных файлах. Концепция функции в языке Си покрывает все типы подпрограмм в других языках программирования: функции и процедуры.

Функции позволяют писать модульные программы, в которых каждая функция выполняет собственную задачу. Функции обычно невелики (страница или меньше, могут быть и однострочные).

Функции позволяют избежать дублирования кода в одной программе. Кроме того, несколько программ могут совместно использовать код функции.

Программы легче читаются, так как детали “скрыты” внутри функций.

Программы легче пишутся, поскольку функции позволяют разбить большие задачи на задачи меньшего размера, более простые для решения.

Выполнение программы всегда начинается с **функции `main()`**. Обычно функция `main()` располагается в начале файла. Порядок расположения

функций в файле не имеет значения. Часто они располагаются в алфавитном порядке для удобства поиска при модификации программы.

Пример 1

Структура программы на языке Си:

объявления

```
int main(...)/ *определение прототипа функции*/
```

```
{
```

```
    объявления
```

```
    операторы
```

```
}
```

объявления

```
int f(...)/ *определение прототипа функции*/
```

```
{
```

```
    объявления
```

```
    операторы
```

```
}
```

```
void g(...)/ *определение прототипа функции*/
```

```
{
```

```
    объявления
```

```
    операторы
```

```
}
```

Примечание 1

Определения прототипов функций не могут быть вложенными.

Язык Си использует свободный, а не позиционный формат расположения программы, что делает несущественным позиции размещения элементов программы. При чтении текста программы на Си компилятор распознает лексемы, разделенные *символами промежутков* (пробелы, знаки табуляции и символы перевода строки). Как правило, символы промежутков игнорируются.

Имя файла, содержащего программу на языке Си должно оканчиваться расширением **.c**.

Минимальная программа на языке Си

Минимальная программа на языке Си должна включать функцию с именем `main()`. В начале выполнения программы управление из операционной системы всегда передается функции `main()`.

Примеры минимальной программы, допустимые с точки зрения синтаксиса, но ничего не делающие:

```
int main ( )
```

```
{
```

```
}
```

или

```
int main ( ){} 
```

или

```
int main
```

```
(
```

```
)
```

```
{
```

```
}
```

Комментарии

Комментарии языка Си — это множество символов, заключенное в "комментарные" скобки `/* */`.

Использование комментариев в программе очень важно. Комментарии могут значительно облегчить восприятие программы (программисты говорят: "повысить читабельность программы"). Удобство отладки и модификации программы сильно зависит от наличия ясных комментариев. Рекомендуется включать комментарии в начало каждого файла с текстом программы на языке Си и в начало каждой функции, определяя назначение функции, смысл параметров и возвращаемого значения. Все важные и неясные фрагменты программы также должны быть прокомментированы.

Комментарии начинаются с символов `/*` и заканчиваются символами `*/`. Они не могут быть вложенными.

```
/*Это /* типичная */ ошибка */
```

Компилятор интерпретирует фрагмент:

```
/*Это /* типичная */
```

как законченный комментарий, после чего будет выдано сообщение об ошибке в оставшемся фрагменте

```
ошибка */
```

Комментарии могут также начинаться с символов `//` (как в языке программирования Си++). В этом случае в качестве комментария рассматриваются все символы, начиная с символов `//` вплоть до конца строки.

Комментарии компилятором игнорируются.

Выражения

Выражение — это конструкция языка Си, имеющая значение.

Этой конструкцией может быть идентификатор, константа, выражение в скобках или осмысленным образом связанные знаками операций выражения. Скобки используются для группировки выражений с целью задания желаемого порядка вычислений.

Пример 1

Выражения - идентификаторы:

- `val`
- `x`
- `total_sum`

Пример 2

Выражения - константы:

- `5`
- `127.6`
- `'B'`
- `"Main menu"`

Пример 3

Выражения, содержащие операции:

- `x + y`
- `current < max`
- `val - num / 10`

Пример 4

Выражения в скобках:

`(x + y)`

`(current < max)`

Операторы

Оператор — это конструкция языка Си, которая вызывает выполнение действия.

В языке программирования Си существуют следующие операторы:

- простые операторы;
- блоки операторов (блоки, составные операторы);
- управляющие операторы.

Простой оператор или оператор-выражение — это выражение, заканчивающееся символом `;` (точка с запятой).

Специальным случаем простого оператора является пустой оператор, который ничего не делает. **Пустой оператор** состоит из единственного символа `;` (точка с запятой). Он используется в тех случаях, когда синтаксис языка требует наличия оператора там, где не требуется никаких действий.

Блок операторов (блок, составной оператор) — это последовательность операторов, заключенная в фигурные скобки { }.

Блок используется для объединения нескольких операторов в один оператор. После закрывающей фигурной скобки точка с запятой может не указываться. После открывающей фигурной скобки можно объявлять переменные. Такие переменные, называемые автоматическими, известны только внутри этого блока, после выхода из блока они больше неизвестны.

"Тело" функции также представляет собой блок операторов.

Управляющие операторы — операторы языка Си, реализующие ветвления алгоритма, циклы и передачу управления.

Все управляющие операторы начинаются с ключевого слова, такого как: **if**, **while**, **for** и т.д.