

Лекция 3. Обзор языка C, продолжение

Простейший ввод и вывод в C и C++

`stdio.h` — заголовочный файл (объявления) функций стандартной библиотеки ввода/вывода (язык C)

`iostream` — заголовочный файл (объявления) функций потоковой библиотеки ввода/вывода (язык C++)

По сути очень близки, но у каждой есть свои удобства и неудобства.

Концепция:

- `stdio` вызов функций ввода/вывода, параметры задают:
- где выполняется ввод/вывод
 - формат (преобразования значений в печатные символы при выводе и символов в значения при вводе)
 - значения для вывода и указатели на переменные для ввода
- `iostream` — объект “поток”, который отвечает за ввод/вывод
- предоставляет разумную форму ввода/вывода по умолчанию
 - можно настраивать форму ввода/вывода
 - запись операции как последовательность компонент, передаваемых в поток (извлекаемых из потока).

Консольный ввод/вывод (экран и клавиатура). Ввод/вывод с файлами.

Стандартные каналы (потоки)

C: `stdin stdout stderr`

C++: `cin cout cerr`

`using namespace std;` — чтобы не писать `std::cin` и т.п.

Функции <code>stdio</code>	операторы <code>iostream</code>
<code>scanf</code>	<code>cin >></code>
<code>printf</code>	<code>cout <<</code>

и другие аналогичные

Спецификации преобразования <code>stdio</code>	методы и манипуляторы <code>iostream</code>
------------------------------------------------	---------------------------------------------

<code>%d</code>	(decimal)		
<code>%u</code>	(unsigned)		
<code>%x</code>	(hexadecimal)	hex	
<code>%c</code>	(char)		
<code>%s</code>	(string)		
<code>%f %lf</code>	(float, double)		
<code>%e %le</code>	(exponential)	scientific	
<code>%g %lg</code>			
...			
<code>%N...</code>	ширина	width(N)	1.23 1.234566

```

%.M          точность          precision(M)
                                fixed

\n
\t \r      endl
    
```

Пример iocompare.cpp с пояснениями в ходе лекции. Разбирается несколько простейших образцов ввода/вывода строк, целых и вещественных чисел, а также ввод и вывод с файлами в простейшем варианте.

Обработка последовательности

Первая серия учебных задач связана с обработкой последовательностей, и мы уже готовы к тому, чтобы программировать решение подобных задач.

Абстрактная постановка.

Есть набор чисел неопределенной длины. Надо вычислить некоторую характеристику этого набора чисел за один просмотр последовательности, т.е. нельзя числа сохранять, нельзя возвращаться назад, нельзя читать повторно. После того как очередное число из последовательности обработано, оно становится более недоступно.

Математический алгоритм решения подобных задач опирается на идею доказательств по индукции (есть даже некоторая теория индуктивных алгоритмов). Другими словами, решение задач часто сводится к подбору некоторых значений и рекуррентных соотношений для них, по которым эти значения можно пересчитывать шаг за шагом. Естественно, нужно также определиться с начальными условиями, с которых начинается расчет.

В качестве простейшего примера можно рассмотреть задачу вычисления среднего арифметического последовательности чисел.

Подойдем с точки зрения индукции.

Пусть мы знаем значение S среднего арифметического для некоторого уже рассмотренного участка последовательности. И только что получили очередное число x из последовательности. Можем ли мы вычислить среднее арифметическое для нового, удлиненного участка последовательности? Ответ, естественно, нет. Но если мы вместе со средним арифметическим знаем количество прочитанных ранее чисел n , то пересчет среднего арифметического (и нового количества элементов) выполняется очевидным образом:

$$S = (S*n + x)/(n+1);$$

$$n = n + 1;$$

Можно сделать еще проще, если вычислять просто сумму чисел A и их количество n :

$$A = A + x;$$

$$n = n + 1;$$

$$S = A/n; \quad (\text{только когда будет надо})$$

Теперь у нас все есть, чтобы запрограммировать решение этой задачи — формулы для вычислений и примеры последовательного чтения данных из файла.

Пример: `sequence-mean.cpp`

Другие задачи решаются аналогично. Выбирается некоторый набор характеристик, которые можно пересчитывать друг за другом при продвижении по последовательности, и через которые можно выразить требуемый ответ. (более подробно — на семинарах).