

Ravesli [Ravesli](#)


- [Уроки по C++](#)
- [OpenGL](#)
- [SFML](#)
- [Qt5](#)
- [RegExr](#)
- [Ассемблер](#)
- [Купить .PDF](#)


Урок №8. Структура программ

 [Юрий](#) |

- [Уроки C++](#)

|

 Обновл. 29 Авг 2020 |

 107019

[1](#)  [15](#)

Компьютерная программа — это последовательность инструкций, которые сообщают компьютеру, что ему нужно сделать.

Оглавление:

1. [Стейтменты](#)
2. [Выражения](#)
3. [Функции](#)
4. [Библиотеки](#)
5. [Пример простой программы](#)
6. [Синтаксис и синтаксические ошибки](#)
7. [Тест](#)
8. [Ответы](#)

Стейтменты

Стейтмент (англ. «*statement*») — это наиболее распространенный тип инструкций в программах. Это и есть та самая инструкция, наименьшая независимая единица в языке C++. Стейтмент в программировании — это то же самое, что и «предложение» в русском языке. Мы пишем предложения, чтобы выразить какую-то идею. В языке C++ мы пишем стейтменты, чтобы выполнить какое-то задание. **Все стейтменты в языке C++ заканчиваются точкой с запятой.**

Есть много разных видов стейтментов в языке C++. Рассмотрим самые распространенные из них:

```
1 int x;  
2 x = 5;  
3 std::cout << x;
```

`int x` — это **стейтмент объявления** (англ. «*statement declaration*»). Он сообщает компилятору, что `x` является переменной. В программировании каждая переменная занимает определенное число адресуемых ячеек в памяти в зависимости от её типа. Минимальная адресуемая ячейка — байт. Переменная типа `int` может занимать до 4-х байт, т.е. до 4-х адресуемых ячеек памяти. Все переменные в программе должны быть объявлены, прежде чем использованы. Мы подробно поговорим о переменных на следующих уроках.

`x = 5` — это **стейтмент присваивания** (англ. «*assignment statement*»). Здесь мы присваиваем значение 5 переменной `x`.

`std::cout << x;` — это **стейтмент вывода** (англ. «*output statement*»). Мы выводим значение переменной `x` на экран.

Выражения

Компилятор также способен обрабатывать выражения. **Выражение** (англ. «*expression*») — это математический объект, который создается (составляется) для проведения вычислений и нахождения соответствующего результата. Например, в математике выражение $2 + 3$ имеет значение 5.

Выражения в языке C++ могут содержать:

- отдельные цифры и числа (например, 2, 45);
- буквенные переменные (например, `x`, `y`);
- операторы, в т.ч. математические (например, `+`, `-`);
- функции.

Выражения могут состоять как из единичных символов — цифр или букв (например, 2 или `x`), так и из различных комбинаций этих символов с операторами (например, $2 + 3$, $2 + x$, $x + y$ или $(2 + x) * (y - 3)$). Для наглядности разберем простой корректный стейтмент присваивания `x = 2 + 3;`. Здесь мы вычисляем результат сложения чисел $2 + 3$, который затем присваиваем переменной `x`.

Функции

В языке C++ стейтменты объединяются в блоки — функции. **Функция** — это последовательность стейтментов. Каждая программа, написанная на языке C++, должна содержать главную функцию `main()`. Именно с первого стейтмента, находящегося в функции `main()`, и начинается выполнение всей программы. Функции, как правило, выполняют конкретное задание. Например, функция `max()` может содержать стейтменты, которые определяют большее из заданных чисел, а функция `calculateGrade()` может вычислять среднюю оценку студента по какой-либо дисциплине.

Совет: Всегда размещайте функцию `main()` в файле `.cpp` с именем, совпадающим с именем проекта. Например, если вы пишете программу `Chess`, то поместите вашу функцию `main()` в файл `chess.cpp`.

Библиотеки

Библиотека — это набор скомпилированного кода (например, функций), который был «упакован» для повторного использования в других программах. С помощью библиотек можно расширить возможности программ. Например, если вы пишете игру, то вам придется подключать библиотеки звука или графики (если вы самостоятельно не хотите их создавать).

Язык C++ не такой уж и большой, как вы могли бы подумать. Тем не менее, он идет в комплекте со **Стандартной библиотекой C++**, которая предоставляет дополнительный функционал. Одной из наиболее часто используемых частей Стандартной библиотеки C++ является **библиотека `iostream`**, которая позволяет выводить данные на экран и обрабатывать пользовательский ввод.

Пример простой программы

Теперь, когда у вас есть общее представление о том, что такое стейтменты, функции и библиотеки, давайте рассмотрим еще раз программу «Hello, world!»:

```
1 #include <iostream>
2
3 int main()
4 {
5     std::cout << "Hello, world!";
6     return 0;
7 }
```

Строка №1: Специальный тип инструкции, который называется **директивой препроцессора**.

Директивы препроцессора сообщают компилятору, что ему нужно выполнить определенное задание. В этом случае мы говорим компилятору, что хотели бы подключить содержимое заголовочного файла `<iostream>` к нашей программе. Подключение заголовочного файла `<iostream>` дает нам возможность использовать функционал библиотеки `iostream`, что, в свою очередь, позволяет выводить нам данные на экран.

Строка №2: Пустое пространство, которое игнорируется компилятором.

Строка №3: Объявление главной функции `main()`.

Строки №4 и №7: Указываем компилятору область функции `main()`. Всё, что находится между открывающей фигурной скобкой в строке №4 и закрывающей фигурной скобкой в строке №7 — считается содержимым функции `main()`.

Строка №5: Наш первый стейтмент (заканчивается точкой с запятой) — стейтмент вывода.

`std::cout` — это специальный объект, используя который мы можем выводить данные на экран. `<<` — это оператор вывода. Всё, что мы отправляем в `std::cout`, — выводится на экран. В этом случае, мы выводим текст `"Hello, world!"`.

Строка №6: Оператор возврата `return`. Когда программа завершает свое выполнение, функция `main()` передает обратно в операционную систему значение, которое указывает на результат выполнения программы: успешно ли прошло выполнение программы или нет.

Если оператор `return` возвращает число `0`, то это значит, что всё хорошо! Ненулевые возвращаемые значения указывают на то, что что-то пошло не так и выполнение программы было прервано. Об операторе `return` мы еще поговорим детально на соответствующем уроке.

Синтаксис и синтаксические ошибки

Как вы, должно быть, знаете, в русском языке все предложения подчиняются правилам грамматики. Например, каждое предложение должно заканчиваться точкой. Правила, которые регулируют построение предложений, называются **синтаксисом**. Если вы не поставили точку и записали два предложения подряд, то это является нарушением синтаксиса русского языка.

Язык C++ также имеет свой синтаксис: правила написания кода/программ. При компиляции вашей программы, компилятор отвечает за то, чтобы ваша программа соответствовала правилам синтаксиса языка C++. Если вы нарушили правила, то компилятор будет ругаться и выдаст вам ошибку.

Например, давайте посмотрим, что произойдет, если мы не укажем в конце стейтмента точку с запятой:

```
1 #include <iostream>
2
3 int main()
4 {
5     std::cout << "Hello world!"
6     return 0;
7 }
```

Результат:

E0065: требуется точка с запятой ";"

C2143: синтаксическая ошибка: отсутствие ";" перед "}"

Допущена синтаксическая ошибка в строке №6: мы забыли указать точку с запятой перед оператором return. В этом случае ошибка на самом деле в конце строки №5. В большинстве случаев компилятор правильно определяет строку с ошибкой, но есть ситуации, когда ошибка не заметна вплоть до начала следующей строки.

Синтаксические ошибки нередко совершаются при написании программ. К счастью, большинство из них можно легко найти и исправить. Но следует помнить, что программа может быть полностью скомпилирована и выполнена только при отсутствии ошибок.

Тест

Теперь давайте проверим то, как вы усвоили материал текущего урока. Ответьте на следующие вопросы:

- В чём разница между стейтментом и выражением?
- В чём разница между функцией и библиотекой?
- Чем заканчиваются стейтменты в языке C++?
- Что такое синтаксическая ошибка?

Ответы

Чтобы просмотреть ответ, кликните на него мышкой.

Ответ №1

Стейтмент — это «полное предложение», которое сообщает компилятору, что ему нужно выполнить определенное задание. Выражение всегда имеет результат (исключение — деление на ноль) и является частью стейтмента.

Ответ №2

Функция — это последовательность стейтментов для выполнения определенного задания.

Библиотека — это последовательность функций, которые могут повторно использоваться в других программах.

Ответ №3

Точкой с запятой (;).

Ответ №4

Синтаксическая ошибка — это ошибка, указывающая на нарушение правил грамматики языка C++.

Оценить статью:



(955 оценок, среднее: 4,94 из 5)



[← Урок №7. Решения самых распространенных проблем](#)

[Урок №9. Комментарии](#)

**Комментариев: 15**

1. *Кирилл:*

[27 июля 2020 в 23:59](#)

Юрий, огромное спасибо вам за такие прекрасные уроки с разором понятий, приведением английских терминов.

Вы второй такой замечательный автор в рунете: создатель матпрофи и вы)

[Ответить](#)



2. *Artur:*

[31 мая 2020 в 06:23](#)

Спасибо огромное за ваш труд ! Очень качественный учебный материал.

[Ответить](#)



1. Юрий:
[31 мая 2020 в 13:33](#)

Пожалуйста)

[Ответить](#)



3. Александр:
[5 января 2019 в 23:30](#)

Юрий я вам благодарен за качественно написанный урок. Спасибо

[Ответить](#)



1. Юрий:
[5 января 2019 в 23:52](#)

Пожалуйста 😊

[Ответить](#)



4. somebox:
[9 ноября 2018 в 03:38](#)

> Например, каждое предложение должно оканчиваться точкой.

Нет. Варианта четыре: точка, многоточие, вопросительный и восклицательный знак. 😊

[Ответить](#)



1. Юрий:
[9 ноября 2018 в 17:46](#)

Действительно, здесь я промахнулся 😊

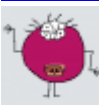
[Ответить](#)



5. Константин:
[1 сентября 2018 в 16:36](#)

Юра, в этом уроке утверждается, мол, C++ не такой уж большой, но он укомплектован Стандартной библиотекой. Вопрос: а где она физически находится? Как туда заглянуть и взять что-нибудь "почитать"? Не в смысле поставить :: и вытащить нужный элемент, а в какой папке она лежит.

[Ответить](#)



6. Алексей:

[14 мая 2018 в 17:55](#)

Программа написана не правильно! она будет вырубаться при запуске, нужно было так:

```
1 #include <iostream>
2 #include <cstdlib>
3 int main ()
4 {
5     cout << "Hello,World!" << endl;
6     system ("pause");
7     return 0;
8 }
```

[Ответить](#)



1. *Юрий:*

[18 мая 2018 в 23:09](#)

У меня в Visual Studio не вырубается. Что делать, чтобы не вырубалось — рассказывается в уроке 7.

[Ответить](#)



2. *Pavel:*

[30 ноября 2018 в 15:47](#)

Программа написана правильно. "Вырубание" программы означает всего лишь то, что после завершения программы ее окно закрывается. В любой среде программирования есть настройки, которые позволяют изменить такое поведение консольной программы. Т.е. можно сделать так, что после завершения программы окно консоли останется открытым.

[Ответить](#)



1. *Эльдар:*

[17 декабря 2018 в 16:02](#)

У него

```
1 using namespace std;
```

не написан.

А программа не должна закрываться, так как у него прописано

```
1 system("pause");
```

[Ответить](#)



1. *Pavel:*

[18 декабря 2018 в 17:25](#)

А причем тут "у него"?

Исходная программа в Visual Studio 2017 действительно закрое окно сразу по завершении. В настройках есть пункт, который изменяет такое поведение. Ну или можно писать

```
1 | system("pause");
```

На вкус и цвет товарища нет.



7. Семён:

[18 февраля 2018 в 19:42](#)

Великолепно ! Все подробно описано !

Но завтра надо почитать ещё раз Для большего усвоения .

[Ответить](#)



1. *rainkiller:*

[27 марта 2018 в 16:05](#)

Как успехи?

[Ответить](#)

Добавить комментарий

Ваш E-mail не будет опубликован. Обязательные поля помечены *

Имя *

Email *

Комментарий

☐ Сохранить моё Имя и E-mail. Видеть комментарии, отправленные на модерацию

☐ Получать уведомления о новых комментариях по электронной почте. Вы можете [подписаться](#) без комментирования.






[TELEGRAM](#)  [КАНАЛ](#)



ПАБЛИК 



ТОП СТАТЬИ

-  [Словарь программиста. Сленг, который должен знать каждый кодер](#)
-  [Урок №1. Введение в программирование](#)
-  [70+ бесплатных ресурсов для изучения программирования](#)
-  [Урок №1: Введение в создание игры «Same Game»](#)
-  [Урок №4. Установка IDE \(Интегрированной Среды Разработки\)](#)

- [Ravesli](#)
- - [О проекте](#) -
- - [Пользовательское Соглашение](#) -
- - [Все статьи](#) -
- Copyright © 2015 - 2020