

Ravesli [Ravesli](#)


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

Урок №17. Операторы

 [Юрий](#) |

- [Уроки C++](#)

|

 Обновл. 2 Сен 2020 |

 50082

[1](#)  [7](#)

Как мы уже знаем из предыдущих уроков, [выражение](#) — это математический объект, который имеет определенное значение. Однако, термин «математический объект» несколько расплывчатый. Точнее будет так: **выражение** — это комбинация литералов, переменных, функций и операторов, которая генерирует (создает) определенное значение.

Оглавление:

1. [Литералы](#)
2. [Операторы](#)
3. [Заключение](#)

Литералы

Литерал — это фиксированное значение, которое записывается непосредственно в исходном коде (например, 7 или 3.14159). Вот пример программы, которая использует литералы:

```
1 #include <iostream>
2
3 int main()
4 {
5     int a = 3; // a - это переменная, 3 - это литерал
6     std::cout << 5 + 2; // 5 + 2 - это выражение, 5 и 2 - это литералы
7     std::cout << "Hello, world!"; // "Hello, world" - это тоже литерал
8 }
```

Литералы, переменные и функции еще известны как операнды. **Операнды** — это данные, с которыми работает выражение. Литералы имеют фиксированные значения, переменным можно присваивать

значения, функции же генерируют определенные значения (в зависимости от [типа возврата](#), исключением являются функции типа void).

Операторы

Последним пазлом в выражениях являются операторы. С их помощью мы можем объединить операнды для получения нового значения. Например, в выражении $5 + 2$, $+$ является оператором. С помощью $+$ мы объединили операнды 5 и 2 для получения нового значения (7).

Вы, вероятно, уже хорошо знакомы со стандартными арифметическими операторами из школьной математики: сложение ($+$), вычитание ($-$), умножение ($*$) и деление ($/$). Знак равенства $=$ является оператором присваивания. Некоторые операторы состоят более чем из одного символа, например, оператор равенства $==$, который позволяет сравнивать между собой два определенных значения.

Примечание: Очень часто новички путают оператор присваивания ($=$) с оператором равенства ($==$). С помощью оператора присваивания ($=$) мы присваиваем переменной определенное значение. С помощью оператора равенства ($==$) мы проверяем, равны ли между собой два определенных операнда.

Операторы бывают трёх типов:

- ➔ **Унарные.** Работают с одним операндом. Например, оператор $-$ (минус). В выражении -7 , оператор $-$ применяется только к одному операнду (7), чтобы создать новое значение (-7).
- ➔ **Бинарные.** Работают с двумя операндами (левым и правым). Например, оператор $+$. В выражении $5 + 2$, оператор $+$ работает с левым операндом (5) и правым (2), чтобы создать новое значение (7).
- ➔ **Тернарные.** Работают с тремя операндами (в языке C++ есть только один [тернарный оператор](#)).

Обратите внимание, некоторые операторы могут иметь несколько значений. Например, оператор $-$ (минус) может использоваться в двух контекстах: как унарный для изменения знака числа (например, конвертировать 7 в -7 и наоборот), и как бинарный для выполнения арифметической операции вычитания (например, $4 - 3$).

Заключение

Это только верхушка айсберга. Более детально об операторах мы обязательно поговорим на следующих уроках.

Оценить статью:

★★★★★ (657 оценок, среднее: 4,95 из 5)



← [Урок №16. Ключевые слова и идентификаторы](#)

Урок №18. Базовое форматирование кода**Комментариев: 7**

1. *Пастор Егор:*
[28 сентября 2020 в 16:15](#)

Брат Юрий, Я и моя паства благодарим тебя за столь понятные ученья, теперь мы можем выйти на новый уровень могущества. (Я пишу от лица кучки студентов, которые пол курса про**али в клубах. Реально очень хорошие и понятные уроки.

[Ответить](#)



1. *Юрий:*
[28 сентября 2020 в 16:19](#)

Спасибо, братья. Очень приятные слова!))

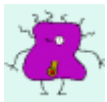
[Ответить](#)



2. *Лазарь:*
[23 марта 2020 в 08:13](#)

У меня стоит Code::Blocks v17. В нём SetConsoleCP(1251) SetConsoleOutputCP(1251) не работают никак, зато setlocale(LC_ALL, "Rus") справляется отлично.

[Ответить](#)



3. *Иван:*
[25 февраля 2018 в 19:05](#)

Спасибо за разъяснения. Да ещё так быстра.

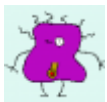
[Ответить](#)



1. *Юрий:*
[25 февраля 2018 в 19:25](#)

Пожалуйста.

[Ответить](#)



4. *Иван:*
[25 февраля 2018 в 16:25](#)

Здравствуй? Подскажи пожалуйста, Почему работает только, если за места имя вводишь цифру?

```
1 #include "stdafx.h"
2 #include <iostream>
3 #include "windows.h"
4
5 int main()
6 {
7     setlocale(LC_ALL, "Rus");
8     using namespace std;
9     cout << "Здравствуй, меня называют СИИ." << endl;
10    Sleep(3000);
11    cout << "А, тебя как?" << endl;
12    long x;
13    cin >> x;
14    Sleep(2000);
15    cout << "Ха-ха-ха, " << x << " ну и имя!" << endl;
16    Sleep(3000);
17    cout << "Досвидание " << x << " !" << endl;
18    Sleep(3000);
19    return 0;
20 }
```

заранее спасибо.

Ответить



1. Юрий:

[25 февраля 2018 в 18:31](#)

Во-первых, кириллица подключается через заголовок:

```
1 #include <Windows.h>
```

и строчки:

```
1 SetConsoleCP(1251);
2 SetConsoleOutputCP(1251);
```

Во-вторых, у тебя для имени установлен тип long. Тип long — это целочисленный тип данных, а имя — это строка. Тип должен быть string:

```
1 string name;
```

Для string подключается отдельный заголовок string:

```
1 #include <string>
```

Готовый код:

```
1 #include "stdafx.h"
2 #include <iostream>
3 #include <Windows.h>
4 #include <string>
5
```

```
6 int main()
7 {
8     SetConsoleCP(1251);
9     SetConsoleOutputCP(1251);
10    using namespace std;
11    cout << "Здравствуй, меня называют СИИ." << endl;
12    Sleep(3000);
13    cout << "А, тебя как?" << endl;
14    string name;
15    cin >> name;
16    Sleep(2000);
17    cout << "Ха-ха-ха, " << name << " ну и имя!" << endl;
18    Sleep(3000);
19    cout << "Досвидание " << name << " !" << endl;
20    Sleep(3000);
21    return 0;
22 }
```

[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