Ravesli Ravesli

- Уроки по С++
- OpenGL
- SFML
- <u>Qt5</u>
- RegExp
- Ассемблер
- <u>Купить .PDF</u>

Урок №16. Ключевые слова и идентификаторы

```
    ♣ Юрий |
    • Уроки С++
    |
    ✓ Обновл. 2 Сен 2020 |
    ◆ 62197
```


Язык С++ имеет зарезервированный набор из 84 слов (включая версию С++17) для собственного использования. Эти слова называются ключевыми словами, каждое из которых имеет свое особое значение.

Вот список всех ключевых слов в языке С++ (включая С++17):

alignas (C++11) alignof (C++11)	decltype (C++11) default	namespace	struct switch
anghor (C++11)	delete	new noexcept (C++11)	template
	do	• ' '	this
and_eq		not	
asm	double	not_eq	thread_local (C++11)
auto	dynamic_cast	nullptr (C++11)	throw
bitand	else	operator	true
bitor	enum	or	try
bool	explicit	or_eq	typedef
break	export	private	typeid
case	extern	protected	typename
catch	false	public	union
char	float	register	unsigned
char16_t (C++11)	for	reinterpret_cast	using
char32_t (C++11)	friend	return	virtual
class	goto	short	void
compl	if	signed	volatile
const	inline	sizeof	wchar_t
constexpr (C++11)	int	static	while
const_cast	long	static_assert (C++11)	xor
continue	mutable	static_cast	xor_eq

<u>C++11</u> также добавил два специальных идентификатора: **override** и **final**. Они имеют особое значение при использовании в определенных контекстах, но не являются зарезервированными ключевыми

словами.

Мы уже сталкивались с некоторыми ключевыми словами, такими как int, void и return. Вместе с набором операторов, ключевые слова определяют весь язык C++ (не включая команд препроцессора). Поскольку они имеют особые значения, то **IDE** всегда выделяют/подсвечивают их другим цветом.

После изучения материалов всех уроков по C++ на Ravesli, вы узнаете суть большинства ключевых слов языка C++, приведенных выше.

Идентификаторы

Идентификатор — это имя переменной, функции, класса или другого объекта в языке C++. Мы можем определять идентификаторы любыми словами/именами. Тем не менее, есть несколько общих правил, которые необходимо соблюдать:

- → Идентификатор не может быть ключевым словом. Ключевые слова зарезервированы.
- → Идентификатор может состоять только из букв (нижнего и верхнего регистра) латинского алфавита, цифр и символов подчёркивания. Это означает, что все другие символы и пробелы запрещены.
- → Идентификатор должен начинаться с буквы (нижнего или верхнего регистра). Он не может начинаться с цифры.
- → Язык C++ различает нижний регистр от верхнего. nvalue отличается от nValue и отличается от nVALUE.

Теперь, когда вы знаете, как *можно* называть объекты, давайте поговорим о том, как их *нужно* называть

Во-первых, в языке С++ имена переменных начинаются с буквы в нижнем регистре. Если имя переменной состоит из одного слова, то это слово должно быть записано в нижнем регистре:

```
1 int value; // корректно
2
3 int Value; // некорректно (должно начинаться с буквы в нижнем регистре)
4 int VALUE; // некорректно (должно начинаться с буквы в нижнем регистре)
5 int ValuE; // некорректно (должно начинаться с буквы в нижнем регистре)
```

Как правило, имена функций также начинаются с буквы в нижнем регистре (хотя есть некоторые разногласия по этому вопросу). Мы будем придерживаться этого стиля, поскольку даже функция main() (главная функция всех программ) начинается с буквы в нижнем регистре, как и все функции из Стандартной библиотеки С++.

Имена идентификаторов, которые начинаются с заглавной буквы, используются для структур, классов или перечислений (об этом позже).

Если имя переменной или функции состоит из нескольких слов, то здесь есть два варианта: разделить подчёркиванием или использовать **CamelCase** — принцип, когда несколько слов пишутся слитно, без пробелов, и каждое новое слово пишется с заглавной буквы. CamelCase (в переводе как «ВерблюжийСтиль») получил свое название из-за заглавных букв, которые напоминают верблюжьи горбы.

```
1 int my_variable_name; // корректно (разделяется символом подчёркивания)
```

```
void my_function_name(); // корректно (разделяется символом подчёркивания)
2
3
4
   int myVariableName; // корректно (используется CamelCase)
5
   void myFunctionName(); // корректно (используется CamelCase)
6
7
   int my variable name; // некорректно (пробелы запрещены)
8
   void my function name(); // некорректно (пробелы запрещены)
9
10
   int MyVariableName; // работает, но не рекомендуется (следует начинать с буквы в них
   void MyFunctionName(); // работает, но не рекомендуется
```

Хотя даже Стандартная библиотека С++ использует символ подчёркивания для переменных и функций, мы же будем использовать CamelCase — для лучшей читабельности кода. Иногда вы будете видеть сочетание двух способов: подчёркивание для переменных и CamelCase для функций.

Стоит отметить, что, если вы работаете с чужим кодом, хорошей практикой будет придерживаться стиля, в котором написан этот код, даже если он не соответствует рекомендациям, приведенным выше.

Во-вторых, не начинайте ваши имена с символа подчёркивания, так как такие имена уже зарезервированы для ОС, библиотеки и/или используются компилятором.

В-третьих, (это, пожалуй, самое важное правило из всех) используйте в качестве идентификаторов только те имена, которые реально описывают то, чем является объект. Очень характерно для неопытных программистов сокращать имена переменных, чтобы сэкономить время при наборе кода или потому, что они думают, что всё и так понятно. В большинстве случаев не всё всегда является понятным и очевидным. В идеале переменные нужно называть так, чтобы человек, который первый раз увидел ваш код, понял как можно скорее, что этот код делает. Через 3 месяца, когда вы будете пересматривать свои программы, вы забудете, как они работают, и будете благодарны самому себе за то, что называли переменные по сути, а не как попало. Чем сложнее код, тем проще и понятнее должны быть идентификаторы.

```
Никто не знает, что такое ccount
                 Плохо
int ccount
int
                 Хорошо Теперь понятно, что мы считаем
customerCount
                         В большинстве нетривиальных случаев — плохо, в простых примерах — может быть
                 Плохо*
int i
                         (например, в циклах)
                 50/50
int index
                         Хорошо, если очевидно, индексом чего является переменная
                 Хорошо Всё понятно
int totalScore
                 Плохо
                         Не начинайте имена переменных с символов подчёркивания
int count
                 50/50
                         Хорошо, если очевидно, что мы считаем
int count
                         Какой тип данных?
                 Плохо
int data
int value1,
                 50/50
                         Может быть трудно понять разницу между переменными
value2
int
                 Хорошо Всё понятно
numberOfApples
                 Хорошо Всё понятно
monstersKilled
                         В большинстве нетривиальных случаев — плохо, в простых примерах — может быть
int x, y
                         (например, в математических функциях)
```

*Примечание: Можно использовать тривиальные имена для переменных, которые имеют тривиальное использование (например, для переменных в цикле, в простых математических функциях и т.д.).

В-четвертых, уточняющий комментарий всегда будет только плюсом. Например, мы объявили переменную с именем numberOfChars, которая должна хранить количество символов определенной части строки. Сколько символов в строке Hello, world!: 10, 11, 12 или 13? Это зависит от того, учитываем ли мы пробелы и знаки препинания или нет. Вместо названия переменной numberOfCharsIncludingWhitespaceAndPunctuation лучше оставить хороший комментарий, который прояснит ситуацию:

```
1 // Эта переменная подсчитывает количество символов части строки, включая пробелы и за int numberOfChars;
```

Тест

Какие из переменных неправильно названы и почему?

```
→ int result;
```

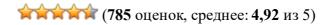
- → int _oranges;
- → int NUMBER;
- → int the name of a variable;
- → int TotalCustomers;
- → int void;
- → int countFruit;
- → int 4orYou;
- → int kilograms of pipe;

Ответ

- → int result. Всё ок.
- → int _oranges. Имена переменных не должны начинаться с символов подчёркивания.
- → int NUMBER. Имена переменных должны начинаться с буквы в нижнем регистре.
- → int the name of a variable. Имена переменных не могут содержать пробелы.
- → int TotalCustomers. Имена переменных должны начинаться с буквы в нижнем регистре.
- → int void. void это ключевое слово и его нельзя использовать в качестве идентификатора для своих переменных или функций.
- → int countFruit. Bcë οκ.

- → int 4orYou. Имена переменных не могут начинаться с цифр.
- → int kilograms_of_pipe. Bcë oĸ.

Оценить статью:





⊖Урок №15. Локальная область видимости



Комментариев: 22



18 февраля 2020 в 15:09

Юрий, скажите, пожалуйста, а есть где-то расшифровка всех ключевых слов и всех операторов? Их значение, и где они могут применяться? Заранее спасибо.

Ответить

2. Александр:

10 апреля 2019 в 04:06

Спасибо Юрий! Понимание начинается когда изучаешь постоянно и систематически. Мне нравится то как вы делаете уроки.

Ответить



Пожалуйста 🙂

Ответить

1. Вячеслав:

21 апреля 2019 в 13:57

Спасибо за уроки. Все здорово. Но, вот в этом уроке наткнулся на странные утверждения, а именно:

"Во-первых, в C++ имена переменных начинаются с буквы в нижнем регистре. Если имя переменной состоит из одного слова, то это слово должно быть записано в нижнем регистре."

Это более, чем странно, потому как у самого Страуструпа четко написано, что именем может быть любое слово, состоящее из букв и цифр, первым символом является буква. Регистр значения не имеет. Допускается символ подчеркивания.

Объясните, пожалуйста.

Ответить



Юрий:

21 апреля 2019 в 13:58

Вы можете называть переменные как хотите и чем хотите, в уроке указаны просто рекомендации.



Вячеслав:

21 апреля 2019 в 14:01

Юрий, спасибо. Я примерно так и предполагал, что это рекомендуемые варианты именования. Однако меня ввело в заблуждение отражение примера с надписями "некорректно".

Еще раз спасибо за оперативный ответ.



• Юрий

21 апреля 2019 в 15:18

Пожалуйста 🙂



5 декабря 2018 в 12:52

Вы пишите что нельзя использовать ключевые слова в названиях функций. Но я видел как у вас в примере использовалось слово = ретурн . Оно слово ключевое . какое этому объяснение ? int return7() — вот этот пример из урока 12.

Ответить



Константин:

22 декабря 2018 в 20:57

Данила, скажи-ка мне, например, чем отличается ник "Данила" от "Данила7"? Сообразил?;-)

Ответить



Юрий:

12 июля 2018 в 21:23

Сайт с которого ведется перевод указан на странице «Уроки С++» в самом конце страницы — LearnCpp.com

Ответить

5. Александр:

27 июня 2018 в 15:08

Добрый день!

Уроки действительно интересны и доходчиво изложены! Спасибо!

Я сам программист 1С (около 5 лет) и мне захотелось научиться писать доработки для 1С в виде внешних dll компонент.

Есть краткий гайд на инфостарте, но он для MS Visual C++6.0 — <u>infostart.ru/public/16916/</u> A вот для Visual C++2017 не разобрался как создать.

Нужно, чтобы на выходе получилась DLL с некоторой функцией (например генератор случайных чисел), к которой можно обратиться из вне. Чтобы получилось ее зарегистрировать через regsvr32.exe.

Если есть возможность ткните носом, где и что изучить надо?

Ответить



Юрий:

28 июня 2018 в 22:57

Привет. Честно, не сталкивался с подобным. Спросите на stackoverflow или Тостере.

Ответить

1. Александр:

2 июля 2018 в 14:29

Ясно.

A как там на stackoverflow затусить?

Что это за портал такой? Просто форум с ответами на вопросы или что-то большее? Не могли бы вы рассказать в паре статей какие ресурсы сильно облегчают жизнь программистам С++ например.

- 1. Что за stackoverflow и есть ли русскоязычный аналог?
- 2. До недавнего времени не знал, что есть онлайн-компиляторы, пока один из пользователей вашего сайта не отписался в комментариях о выполненном задании. onlinegdb.com Тоже очень интересный ресурс: получается можно и код отладить и скинуть ссылку на него?
- 3. Чем еще можно пользоваться? Есть ли какие-нибудь mind-map для запоминания основных операторов?
- 4. Есть ли мобильные приложения, облегчающие изучение/работу в среде С++?
- 5. И вообще хотелось бы увидеть конкретную работу: Например взять какой-то минипроект трудоемкостью на 1-2 часа. (есть такие на фриланесе или не бывает??) Выложить его техническое задание и по шагам реализацию проекта.

Т.е. после вашего пособия каков примерный уровень программиста C++ будет и можно ли браться за какую-то работу или нужно углубляться в какие-то области?

Ответить



2 июля 2018 в 22:51

Спасибо за столь ценный и развернутый комментарий.

- 1. Есть русский ресурс вопросов/ответов в сфере ИТ <u>Tocrep</u>. Но есть и русская версия StackOverflow <u>ru.stackoverflow.com</u>.
- 2. Есть onlinegdb, есть еще сервис <u>Pastebin</u> для хранения и распространения кода (но это не компилятор C+++).
- 3-4-5. Насчет mind-map и мобильных приложений пока что ничего не могу посоветовать. Насчет конкретного проекта эта мысль давно уже кругится в голове. Можно будет сделать. Но на это всё нужно время и ресурсы.

В ближайшем времени планирую выпустить несколько общих статей о программировании с советами для новичков, где будут рассматриваться и сервисы и рекомендации, и нюансы. Всё будет, но не сразу, а постепенно.

2. *Александр*: 4 июля 2018 в 14:15

По поводу ресурсов — у вас даже нет кнопки типа "поддержать проект". Непорядок =))

Все отзывы у вас в основном положительные, я думаю многие не против будут отблагодарить авторов.

По поводу мотивации — я думаю интересно многим будет увидеть реальный проект, который из урока в урок будет обрастать новыми подробностями. Например создание мини-игры (насколько можно это реализовать — не знаю) Перемещение фигуры по экрану. Т.е. что-то, что можно будет пошупать в результате прохождения доброй половины курса.

А еще лучше скомпилировать эту мини-игру под андроид. Я думаю желающих у вас прибавится.



Юрий:

10 июля 2018 в 00:23

Как я уже говорил, для этого нужно время и ресурсы (здесь упор больше на приложенные усилия). Я еще не закончил перевод этого курса по C++ и сразу перебегать на второй цикл статей, не закончив предыдущий — это просто того не стоит. В будущем можно будет что-то придумать. Пока что нужно довести первое дело до конца.

P.S. Если будут желающие поделиться своим опытом разработки каких-либо проектов и напишут цикл статей об этом раньше, чем я закончу перевод уроков — без проблем, опубликую.

6. Erding:

6 сентября 2017 в 18:09

Очень нравятся уроки, поскольку всё понятно описано даже для людей, знакомых с этим достаточно плохо.

Я хотел спросить совета у Вас:

В свои 15-ть лет, я безумно хочу обучиться программированию на языке С++. Читаю абсолютно все Ваши уроки полностью, а также прочитал весь курс введения. Но часто, я просыпаюсь с мыслью, что я опять забыл тот или иной код. Я нахожу свободное время, сажусь, снова перечитываю урок, записываю это где-нибудь, но не могу выйти на более-менее автоматический режим. (Без заметок, подсказок.) Это, иногда, попросту угнетает, отбивая при этом желание продолжать.

Подскажите, пожалуйста, что нужно делать, для лучшего запоминания информации, которую я получаю с Ваших уроков?

Ответить



Юрий:

6 сентября 2017 в 18:54

Помнить всё и не нужно. Сомневаюсь, что есть программисты, которые запоминают всю, однажды полученную, информацию раз и на всю жизнь. Это нормально, всё не выучишь и не запомнишь. Главное — понять сугь, получить базис (фундамент) знаний, а всё остальное — это практика. Ведь для создания программы Вам нужно реализовать теорию, т.е. написать код, исходя из своих знаний. Отсюда и вытекает факт, что для лучшего запоминания чего-либо — нужно это применить на практике. Прочитали урок применили на практике. Хорошим вариантом будет написания своей программы или игрушки (небольшой, простенькой), можно попробовать найти на YouTube видео-уроки в стиле «Как написать/создать игру на C++» и повторять юд. А если Вы забыли какой-то термин, понятие или правило, то это не страшно — зашли в Интернет, перечитали, вспомнили. И, как уже было сказано, всё не выучишь и не запомнишь, а практическое применение знаний — самый лучший вариант запоминания информации.

Ответить



Сергей:

24 августа 2017 в 21:30

По мере продвижения по уроках с++, количество оценок заметно уменьшылось.

Ответить



Юрий:

24 августа 2017 в 21:59

Далеко не все продолжают учиться, мотивация кончилась, так всегда 🙂



Ответить

1. *Андрей*: 2 декабря 2018 в 08:05

А некоторые "на автомате" продолжают упорно не замечать кнопок оценки Ибо привыкли, что потребуют регу (в результате которой почту завалят спамом (бедные мои спамоприемники)), какие-нибудь репосты или вообще вылезет/ попытается вылезти очередная ужасная реклама какого-нибудь мыла/яндекса... о_о В общем пойду проголосую за все статьи, что прочитал В Надеюсь это кому-то важно

Ну и да, спасибо за перевод. Коротко и понятно 🙂

Ответить



В некоторых вообще могут быть блокираторы оценок и кнопок шеринга — и их тоже можно понять. Спасибо, что читаете $\ensuremath{\mathfrak{C}}$

Добавить комментарий

Ваш E-mail не будет опубликован. Обязательные поля поме	чены *
Имя *	
Email *	
Комментарий	
□ Сохранить моё Имя и Е-таіl. Видеть комментарии, отпр	равленные на модерацию
Получать уведомления о новых комментариях по элект комментирования.	ронной почте. Вы можете <u>подписаться</u> без
Отправить комментарий	
TELEGRAM X КАНАЛ	
Электронная почта	
паблик Ж_	

ТОП СТАТЬИ

- 🗏 Словарь программиста. Сленг, который должен знать каждый кодер
- 2 70+ бесплатных ресурсов для изучения программирования
- ↑ Урок №1: Введение в создание игры «Same Game»
- <u>\$\frac{\text{V}}{\text{V}} Урок №4. Установка IDE (Интегрированной Среды Разработки)</u>
- Ravesli
- - О проекте -
- - Пользовательское Соглашение -
- - Все статьи -
- Copyright © 2015 2020