

# Использование числовых переменных

**№ урока:** 10 **Курс:** Процедурное программирование на языке C#

**Средства обучения:** Visual Studio 2019 Community Edition

## Обзор, цель и назначение урока

На уроке будут рассмотрены основные типы данных, применяемые для создания числовых переменных. Будет рассмотрено понятие алиаса-псевдонима, вопрос преобразования числовых переменных одних типов в другие и использование суффиксов при инициализации переменных.

## Изучив материал данного занятия, учащийся сможет:

- Понимать и применять полные названия типов и их псевдонимы.
- Уметь выполнять преобразования типов.

## Содержание урока

1. Размеры значений числовых типов
2. Полное название типа и его псевдоним
3. Переменные и отрицательные числа
4. Преобразование значений числовых типов
5. Числовые суффиксы

## Резюме

- Для проверки размера типа, можно воспользоваться оператором **sizeof**, который возвращает размер типа в байтах.
- **Псевдоним** – альтернативное имя типа, которое можно использовать вместо системных типов. Например, **decimal** – это псевдоним типа **Decimal**, а тип **int** – псевдоним типа **Int32**. Псевдоним еще называют **алиасом**.
- Полное имя типа и его псевдоним, полностью взаимозаменяемы.
- Тип **short**, переводится как «короткий», и символизирует старое, классическое 16-и битовое машинное слово.
- Тип **int** – это сокращение слова integer, которое переводится как «целочисленный». И тип **int**, имеется ввиду, псевдоним **Int32**, символизирует 32-х битовое машинное слово.
- Тип **long** – переводится как «длинный». Символизирует, 64-х битовое машинное слово.
- Тип **float** – переводится как «плавающий». Произошло название от того, что вещественные числа на языке электроники и информатики называются – числами с плавающей точкой.
- Числа с плавающей точкой, имеют такую характеристику как – точность. Точность числа, это, говоря упрощенно, количество цифр после запятой, то есть количество цифр в дробной части числа.
- Вещественное число типа **float** – это число с одинарной точностью (**float** — это алиас типа **Single**, «одинарный»), а вещественное число типа **double** – это число с двойной точностью (**double** – алиас типа **Double**, «двойной»).
- Числа одинарной точности с плавающей точкой обеспечивают относительную точность 7-8 десятичных цифр.
- Числа двойной точности с плавающей точкой обеспечивают относительную точность около 16 десятичных цифр. В компьютерах, которые имеют 64-разрядные с плавающей точкой арифметические единицы, большинство численных вычислений осуществляется

с помощью чисел двойной точности с плавающей точкой, поскольку использование чисел одинарной точности обеспечивает почти такую же производительность, но меньшую точность.

Пример: `float x = 1.2345678f;`  
`double y = 1.23456789123456;`

- **Decimal**, переводится как десятичный. Тип **decimal** – имеет размер в 2 раза больше, чем **double**. Тип **Decimal** разрабатывался для того, чтобы исключить проблемы, связанные с округлением денежных сумм.
- **Decimal** стоит применять только тогда, когда дело касается финансовых расчетов, то есть, операций с денежными суммами.
- **Кастинг** - преобразование значений одного типа в значение другого типа.  
Кастинг бывает двух видов. Это **неявный** (*безопасный*) кастинг и **явный** (*опасный*) кастинг.  
Опасность кастинга заключается в возможности получения искаженного результата, который потом может испортить все дальнейшие расчеты.
- **Неявное преобразование типа** (безопасное) – преобразование меньшего типа в больший или целого типа в вещественный. Является безопасным, так как не происходит потеря точности.
- **Явное преобразование типа** (опасное) – преобразование большего типа в меньший или вещественного типа в целый. Является опасным, так как происходит потеря точности результата без округления.
- Оператор явного преобразования значения к определенному типу ставится перед той переменной, значение которой требуется преобразовать. Этот оператор и называется явным, потому что, мы явно (руками программиста) указываем его в тексте программы.
- Опасность преобразования, например, из **double** в **byte**, может повлечь за собой:
  - во-первых - потерю точности (то есть потерю дробной части числа);
  - во-вторых - потерю знака;
  - в-третьих - сужение целой части дробного числа (то есть урезание целой части).
- Мы должны быть крайне осторожными, производя явные (опасные) преобразования значений типов, так как в результате *опасного* преобразования может получиться непредсказуемый результат, который может привести к нежелательному итогу.
- Преобразование значений типов называют **приведением типов**.
- Значение, которое присваивается переменным называется – **литералом**.
- **Числовой суффикс**, это буква, которую мы можем поставить в конце числа-литерала некоторых типов. Суффикс — это метка, которой программисты назначают тип числовому значению.
- Суффиксом «U» помечается значение типа **uint**, суффиксом «UL» помечается значение типа **ulong**, суффиксом «L» помечается значение типа **long**, суффиксом «F» - **float**, «D» - **double**, «M» - **decimal**. В роли суффикса могут выступать, как маленькие, так и большие буквы.
- Рекомендуют не использовать маленькую букву «L», потому что она визуально похожа на единицу, и считается, что есть вероятность того, что мы можем ошибиться при анализе числа с таким суффиксом.

### Закрепление материала

- Что такое кастинг?
- Для чем отличается безопасное преобразование типов от опасного?
- Какое значение имеет числовой суффикс?
- Что такое точность числа, как решается вопрос с точностью числовых типов C #?

## Самостоятельная деятельность учащегося

- Ознакомьтесь с дополнительными материалами к уроку.
- Выполните преобразования переменной типа `decimal`, хранящей в себе значение 3,1415926536 во все известные вам числовые типы.
- Создайте переменную типа `float` и поместите в нее число, равное 2,7182818284590452. Выполните преобразование значения переменной во все известные числовые типы. При создании переменных используйте полные имена типов, а не алиасы. Выведите значения каждой переменной на экран. Используйте безопасное приведение.

## Рекомендуемые ресурсы

Справочник по C#. Оператор sizeof

<https://docs.microsoft.com/ru-ru/dotnet/csharp/language-reference/operators/sizeof>

Система общих типов CTS

<https://docs.microsoft.com/ru-ru/dotnet/standard/base-types/common-type-system>

[https://ru.wikipedia.org/wiki/Число\\_одинарной\\_точности](https://ru.wikipedia.org/wiki/Число_одинарной_точности)

[https://ru.wikipedia.org/wiki/Число\\_с\\_плавающей\\_запятой](https://ru.wikipedia.org/wiki/Число_с_плавающей_запятой)

Что нужно знать про арифметику с плавающей запятой

<https://habr.com/ru/post/112953/>