

# Константы

**№ урока:** 22 **Курс:** Процедурное программирование на языке C#

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

## Обзор, цель и назначение урока

Константы, как и переменные, являются важным участником процесса работы с данными в коде. На данном уроке будут рассмотрены вопросы создания констант, кастинга констант, а также применения констант на практике.

## Изучив материал данного занятия, учащийся сможет:

- Понимать определение константы.
- Уметь использовать константы в своем коде.
- Понимать и уметь применять рекомендации по именованию и использованию констант.

## Содержание урока

1. Переменные и постоянные величины
2. Константы в школе
3. Кастинг констант
4. Константы на практике и рекомендации по применению

## Резюме

- **Величина** – это характеристика объекта, явления или процесса, которая может быть выражена количественно (то есть, представлена числом). Величины бывают переменными и постоянными.
- **Константа** или постоянная — это величина, значение которой не меняется. В этом она противоположна переменной.
- Именованные константы, аналогичны переменным за исключением того, что вы не можете изменить значение константы после ее инициализации.
- Несколько определений констант:  
**Константа** – это символ, используемый для обозначения некоторой постоянной величины, значение которой нельзя изменять в ходе выполнения вычислений. (школьное определение)  
**Константа** – это конструкция языка программирования, обозначающая некоторое значение, которое нельзя изменять в ходе выполнения программы. (практическое определение)  
**Константа** – это именованная область оперативной памяти, которая хранит в себе некоторое значение, которое нельзя изменить. (техническое определение)
- При преобразовании значения константы из *более ёмкого* типа в *менее ёмкий* тип (например, из типа `int` в тип `byte`), если значение константы попадает в допустимый диапазон значений переменной, то не требуется использовать оператор явного преобразования значения типа. То есть, в данном случае допустим неявный кастинг.

- Над константами нельзя производить унарных операций, таких как инкремент или декремент.
- Макконнелл советует по мере возможности (то есть, где только можно), вместо переменных, использовать константы для определения любых локальных переменных, значения которых должны оставаться неизменными после инициализации.
- Во всех «си-подобных» языках, все имена констант, рекомендуется писать в «АПЕРКЕЙСЕ», то есть заглавными буквами. Все слова, из которых состоит имя, следует отделять друг от друга знаками нижнего подчеркивания.
- Помещайте константы с левой стороны от операторов сравнений при сравнении значений булевых типов.

### Закрепление материала

- Что такое константа?
- Каковы отличия в поведении переменных и констант при преобразовании типов?
- Какая есть рекомендация по именованию констант?
- Почему константы булевого типа рекомендуется размещать с левой стороны от операторов сравнения?

### Самостоятельная деятельность учащегося

- Задание 1

Ознакомьтесь с дополнительными материалами к уроку.

- Задание 2

Напишите программу, в которой в теле метода Main создайте две константы - с именем PI, которой присвойте значение 3,141593, и с именем MY\_CONST, которой присвойте значение true.

Выполните сравнение того значения, которое ввел пользователь, с тем значением, которое хранится в константе PI и сохраните результат сравнения в переменную с именем iSConstBigger. Далее, проверьте на равенство значение переменной iSConstBigger со значением MY\_CONST. Результат последнего сравнения выведите на экран консоли.

### Рекомендуемые ресурсы

<https://ru.wikipedia.org/wiki/Постоянная>

[https://ru.wikipedia.org/wiki/Константа\\_\(программирование\)](https://ru.wikipedia.org/wiki/Константа_(программирование))

Константы (Руководство по программированию на C#)

<https://docs.microsoft.com/ru-ru/dotnet/csharp/programming-guide/classes-and-structs/constants>