

№ урока: 14 **Курс:** Процедурное программирование на языке C#

Средства обучения: Visual Studio 2019 Community Edition

Обзор, цель и назначение урока

Этот урок посвящен различным операциям над строковыми переменными, а также различным способам контроля за выводом строк на экран консоли.

Изучив материал данного занятия, учащийся сможет:

- Работать со строковыми переменными с использованием специальных флагов форматирования, форматированным выводом и интерполяцией строк.
- Понимать и уметь применять конкатенацию строк для более удобного использования литералов, и создания новых значений строковых переменных.
- Понимать суть и уметь работать с константами в приложении.

Содержание урока

1. Конкатенация строк
2. Форматированный вывод
3. Флаги форматирования
4. Интерполяция строк
5. Эскейп-последовательности

Резюме

- Конкатенация (Объединение строк) — это добавление одной строки к концу другой. Вы можете сцеплять строки с помощью оператора «плюс».
- Рекомендация Стива Макконнелла по написанию приложений и их рефакторингу: «Присваивайте результат вычисления выражения промежуточной переменной, имя которой резюмирует суть выражения».
- Чтобы повысить удобочитаемость исходного кода, используйте сцепление строк, разделяя длинный строковой литерал на литералы меньшего размера и ставя между ними оператор “+”.
- Количество строк не влияет на производительность во время выполнения.
- **Форматированный вывод**, или как его еще называют составной формат строк – это набор операторов и правил их использования, позволяющий настроить вывод текстовой информации на экран консоли.
- **Маркер подстановки** – это специальный оператор форматированного вывода, позволяющий указать место в строке вывода, куда может быть вставлено значение некой переменной по ее идентификатору.
- **Строка составного формата** – способ представления строки, при котором в основной текст включены элементы для подстановки значений по индексу (маркеры подстановки), которые соответствуют объектам из списка и другие элементы, меняющие формат текста, которые называются элементами форматирования. Операция форматирования создает результирующую строку, состоящую из исходного фиксированного текста, в который включаются строковые представления объектов из списка.

- **Интерполяция** в C#, это просто удобный синтаксис для произведения операций над «строками составного формата» без указания индекса элемента подстановки, а обращаясь напрямую по идентификатору.
- Строка составного формата может быть указана только в параметрах метода. Присвоить ее переменной напрямую не получится.
- У типа `string` есть метод с именем `Format`, которому можно в параметры передать «строку составного формата» для получения готовой строки с подставленным в нужные места значениями.
- Для форматирования числовых результатов и вывода их на экран можно использовать метод `Console.WriteLine()` или `Console.WriteLine()`, который внутри вызывает метод `string.Format()`. Формат задается с помощью флагов форматирования.
- **Флаги форматирования** (или как их еще называют – «Описатели формата») – набор операторов, позволяющий настроить внешний вид выводимых на экран текстовых данных, согласно какому-то из существующих форматов представления данных, например: формат процентов, формат валюты и др.
- Флаг форматирования может иметь следующую форму: `Axx`, где `A` – флаг формата (определяет тип формата), а `xx` – описатель точности (количество отображаемых цифр или десятичных знаков форматированного результата). Например:
`Console.WriteLine("{0:F2}", 99.935);`
- Существуют следующие **флаги форматирования** строк:
 - `C` или `c` – валюта (Currency);
 - `D` или `d` – десятичное число (Decimal);
 - `E` или `e` – научный формат (Scientific, exponential)
 - `F` или `f` – формат с фиксированным значением после запятой (Fixed-point)
 - `G` или `g` – общие (General)
 - `N` или `n` – Number (Number)
 - `X` или `x` – шестнадцатеричный формат (Hexadecimal)
 - `P` или `p` – процентный (Percent)
- **Форма Бэкуса-Наура**, это формальная система, для описания синтаксиса языков программирования.
- **Константа (Constant)** – это область памяти, которая хранит в себе некоторое значение, которое нельзя изменить.
- **Правила использования констант:**
 - Константам необходимо присваивать значение непосредственно в месте создания;
 - Попытка присвоения константе нового значения приводит к ошибке уровня компиляции;
- **Эскейп последовательности** - сочетания символов, начинающиеся с обратной косой черты, за которой следует буква или набор цифр. Для представления знака новой строки, одиночной кавычки или некоторых других символов в символьной константе, необходимо использовать `escape`-последовательности. `Escape`-последовательность рассматривается как один символ и, следовательно, является допустимой символьной константой. Они также используются для обозначения буквенных представлений непечатаемых символов, а также символов, которые обычно имеют специальное значение, например двойных кавычек (`"`).
- `Escape`-последовательности:
 - `\?` Литерал вопросительного знака
 - `\'` Одиночная кавычка
 - `\"` Двойная кавычка
 - `\\` Обратная косая черта
 - `\a` Звонок (предупреждение)

- \b Backspace
- \n Новая строка
- \r Возврат каретки
- \t Горизонтальная табуляция
- \v Вертикальная табуляция

Закрепление материала

- Что такое форматированный вывод?
- В чем отличия форматированного вывода от интерполяции?
- Как применяются escape-последовательности?
- Что такое конкатенация?
- Что такое константа и чем она отличается от переменной?

Самостоятельная деятельность учащегося

- Задание 1.

Ознакомьтесь с дополнительными материалами к уроку.

- Задание 2.

Напишите приложение, в котором программа получает и сохраняет в отдельные переменные от пользователя сначала его имя, потом его фамилию, а затем выводит на экран строку по типу "Hello <Имя_пользователя> <Фамилия пользователя>! It's a god day today!" Данный текст должен быть составлен одной строкой, при этом после первого предложения должен выполняться перенос строки на новую с использованием escape-последовательности.

Рекомендуемые ресурсы

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

Составное форматирование

<https://docs.microsoft.com/ru-ru/dotnet/standard/base-types/composite-formatting>

Строки настраиваемых числовых форматов

<https://docs.microsoft.com/ru-ru/dotnet/standard/base-types/custom-numeric-format-strings>

Строки стандартных числовых форматов

<https://docs.microsoft.com/ru-ru/dotnet/standard/base-types/standard-numeric-format-strings>

[https://ru.wikipedia.org/wiki/Интерполяция_\(текстология\)](https://ru.wikipedia.org/wiki/Интерполяция_(текстология))

\$ — интерполяция строк (справочные материалы по C#)

<https://docs.microsoft.com/ru-ru/dotnet/csharp/language-reference/tokens/interpolated>

https://ru.wikipedia.org/wiki/Форма_Бэкуса_—_Найра

Escape-последовательность

<https://docs.microsoft.com/ru-ru/cpp/c-language/escape-sequences?view=vs-2019>