

Правила использования методов

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

Средства обучения: Visual Studio 2019 Community Edition

Обзор, цель и назначение урока

Правильное использование методов требует знания особенностей их создания, способов перегрузки методов, возможность создания вложенных методов, правила именования методов и другие особенности работы с ними. Эти темы будут рассматриваться на текущем уроке.

Изучив материал данного занятия, учащийся сможет:

- Понимать, что такое перегрузка метода и как ее использовать.
- Уметь использовать локальные методы.
- Понимать правила именования методов в коде.
- Понимать смысл понятия функциональная связность в отношении методов и необходимость реализовывать её в своих программах.

Содержание урока

1. Перегрузка методов
2. Опциональные параметры как замена перегрузки
3. Перегрузка методов с позиционными ref, in и out параметрами
4. Перегрузка методов с опциональными in-параметрами
5. Вложенные методы
6. Множественные возвраты
7. Правила именования методов
8. Методы предикаты
9. Связность в методах

Резюме

- Перегрузка, в программировании, обозначает возможность использования одноимённых подпрограмм, процедур или функций в языках программирования.
- **Перегрузка методов** - возможность создания нескольких методов с одним именем, но разным типом, или количеством параметров.
- Существует возможность неоднозначности при вызове перегруженных методов с опциональными (необязательными) параметрами. В этом случае будет возникать ошибка уровня компиляции.

- Перегруженные методы могут отличаться наличием/отсутствием `ref` и `out` аргументов, но только в том случае, если они имеются только у одного метода. Если единственное различие между методами заключается в том, что один метод принимает аргумент `ref`, а другой – `out`, они не могут быть перегружены.
- У метода с простым опциональным параметром, может быть только одна перегрузка с ссылочным опциональным параметром, и причем, только с `in` параметром.
- Опциональные параметры недопустимо помечать модификаторами `ref` и `out`.
- Вложенные, или **локальные методы** – это обычные методы, только созданные не в блоках-классов, а внутри других методов. Все правила, которые касаются работы с обычными методами, распространяются и на работу со вложенными методами.
- Стив Макконнелл пишет, что подход к написанию метода с множеством `return`-ов, это, в большинстве случаев, не очень хорошо. Он советует, по возможности, минимизировать число возвратов из создаваемых нами методов.
- Локальные методы, с точки зрения процедурного программирования, это практически то же самое, что и безымянные блоки, только с настоящим именем и возможностями, которые присущи обычным методам.
- Правила именования методов.
 - Рекомендуется методы именовать в **Pascal Case**.
 - Имена методов должны быть **глаголами в неопределённой форме** на английском языке и без частицы «To», например: перевести, написать, копировать, вычислить, сложить, вычесть, удалить - Translate, Write, Copy, Calculate, Add, Subtract, Delete.
- **Метод-предикат** - это метод, который возвращает булево значение.
- Связность в программировании, применительно к методам, обозначает, некую меру самодостаточности того или иного метода.
- **Функциональная связность** — это самый сильный и лучший вид связности. Такая связность присутствует, когда метод выполняет одну и только одну операцию. Примерами методов, обладающих высокой (хорошей) связностью, являются методы: вычисления синуса, получения фамилии заказчика, удаления файла, вычисления платы за кредит и определения возраста по дате рождения.

Закрепление материала

- Что такое перегрузка методов?
- Как можно перегружать методы с опциональными параметрами?
- Что такое функциональная связность?

- Какие правила именования применяются к методам?

Самостоятельная деятельность учащегося

- Задание 1

Ознакомьтесь с дополнительными материалами к уроку.

- Задание 2

Создать набор перегрузок метода PrintResult:

- Принимает 3 целочисленные переменные и выводит на экран результат их сложения.
- Принимает 3 целочисленные переменные и выводит на экран результат их разности.
- Принимает 2 целочисленные переменные и выводит первое число в степени второго.
- Принимает 3 целочисленные переменные и вводит на экран сумму значений этих переменных представленных в строковом виде (пример: «3» + «4» + «5»).

Рекомендуемые ресурсы

https://ru.wikipedia.org/wiki/Перегрузка_процедур_и_функций

CIL&C# Перегрузка методов по возвращаемому значению

<https://www.youtube.com/watch?v=VuTyuT7tJAQ>

https://en.wikipedia.org/wiki/Nested_function