

Правила создания переменных

№ урока: 18 **Курс:** Процедурное программирование на языке C#

Средства обучения: Visual Studio 2019 Community Edition

Обзор, цель и назначение урока

Данный урок поможет понять и выучить основные правила именования переменных. Также на уроке рассматривается вопрос строгой и слабой типизации переменных, понятие неявно типизированных переменных и вопрос их инициализации.

Изучив материал данного занятия, учащийся сможет:

- Знать существующие правила именования при создании идентификаторов.
- Понимать смысл и уметь использовать неявнотипизированные локальные переменные.

Содержание урока

1. Правила именования
2. Строгая и слабая типизация
3. Неявно типизированные переменные
4. Инициализация

Резюме

- Имя переменной в программировании, ещё называется идентификатором.
- Первый символ имени переменной может быть буквой, знаком подчеркивания или символом At - @ - «собака». Все другие символы, в качестве первого символа имени – недопустимы.
- Остальные символы имени переменной, кроме первого, могут быть буквами, цифрами или знаком подчеркивания, а все другие символы использовать в имени переменной – недопустимо.
- Использование символа «собака» допустимо только на первой позиции.
- **Ключевые слова** – это предварительно определенные зарезервированные идентификаторы, имеющие специальные значения для компилятора. Их нельзя использовать в программе в качестве идентификаторов, если только они не содержат префикс @.
- Язык C# чувствителен к регистру, поэтому одинаковые имена в разном регистре – это разные имена. Например, MyName и myName – это разные имена.
- Спецификация языка C# рекомендует придерживаться определенных правил (casing conventions) при создании идентификаторов (выбора имен для ваших переменных, методов и т.д).
- К таким правилам можно отнести:
 - Pascal casing** – каждое слово в идентификаторе начинается с большой буквы;
 - Camel casing** – каждое слово, исключая первое, в идентификаторе начинается с большой буквы;
 - Uppercase** – идентификатор состоит из букв, написанных в верхнем регистре (все буквы большие). Такой стиль в C# допустим, но не рекомендуется его применять, только для имен переменных, которые представляют собой аббревиатуры.
- **Венгерская нотация** – это такой стиль именования, когда мы в начале имени переменной указываем какую-то специфическую информацию. И для этого, в начале имени переменной, мы приписываем какой-то префикс.
- Важная рекомендация: не начинать имена с символов подчеркивания и не пользоваться венгерской нотацией
- **Статическая типизация** (или строгая типизация) – это подход, в программировании, когда переменная, связывается, с её типом, в момент создания этой переменной, и тип, назначенный этой переменной, уже не может, быть, изменён позже (то есть, переменной допустимо присваивать значения только этого типа).

- **Динамическая типизация** (или слабая) — это подход в программировании, когда переменная не связывается с каким-то конкретным типом в момент её создания, и тип переменной может быть изменён позже (то есть, такой переменной допустимо присваивать значения не только какого-то одного типа, а ей можно присваивать значения разных типов)
- **С#** — это строго типизированный язык, имеющий механизмы, использующие динамическую типизацию.
- **Неявно-типизированная локальная переменная var** это ключевое слово, указывающее компилятору, что он должен определить тип для этой переменной самостоятельно, исходя из присваиваемого значения, или выражения инициализации».

Закрепление материала

- Что такое венгерская нотация?
- Чем способ именования Pascal casing отличается от Camel casing?
- Что такое var?
- Что такое статическая типизация?
- Какие особенности типизации присутствуют в С#?
- Какой способ именования нужно использовать для именования переменных?

Самостоятельная деятельность учащегося

- Задание 1

Ознакомьтесь с дополнительными материалами к уроку.

- Задание 2

Используя Visual Studio, создайте проект по шаблону Console Application. Проверьте, можно ли создать переменные со следующими именами: uberflu?, _Identifier, \u006fIdentifier, &myVar, myVariab1e

- Задание 3

Создайте переменные с идентификаторами, соответствующими правилам именования, рекомендуемым компанией Microsoft:

- Переменная, хранящая в себе значение налога на добавленную стоимость (процент).
- Переменная, хранящая в себе значение площади жилой квартиры.
- Переменная, хранящая в себе значение энергии, потребленной квартирой номер 250 за месяц (киловатт*часов).
-

Рекомендуемые ресурсы

Обзор книги Стива Макконнелла «Совершенный код»

<https://www.youtube.com/watch?v=Eo-aSagDFd0>

Неявно типизированные локальные переменные (руководство по программированию на С#)

<https://docs.microsoft.com/ru-ru/dotnet/csharp/programming-guide/classes-and-structs/implicitly-typed-local-variables>

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

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

Правила именования

[https://docs.microsoft.com/ru-ru/previous-versions/dotnet/netframework-4.0/ms229002\(v=vs.100\)](https://docs.microsoft.com/ru-ru/previous-versions/dotnet/netframework-4.0/ms229002(v=vs.100))

Неявно типизированные локальные переменные (руководство по программированию на C#)
<https://docs.microsoft.com/ru-ru/dotnet/csharp/programming-guide/classes-and-structs/implicitly-typed-local-variables>