

Структуры и их разновидности

№ урока: 7 Курс: C# Essential

Средства обучения: Компьютер с установленной Visual Studio

Обзор, цель и назначение урока

Рассмотрение структур и их членов.
Рассмотрение наследования структур.
Рассмотрение вложенных структур.

Изучив материал данного занятия, учащийся сможет:

- Понимать работу структур.
- Понимать принципы наследования структур

Содержание урока

1. Рассмотрение структур.
2. Конструкторы.
3. Наследование в структурах.
4. Вложенные структуры и классы.

Резюме

- Структура (классическая) – это конструкция языка, позволяющая содержать в себе набор полей различных типов.
- Структура в C# – это конструкция языка, состоящая из ключевого слова `struct`, идентификатора и тела. Структура может содержать в своем теле поля, свойства и методы. Также структуры могут включать в свое тело другие структуры и классы, но такой подход не является широко распространённой техникой.
- Структуры должны использоваться только для хранения маленьких, единых, желательно неизменных значений, которые не будут часто упаковываться.
- В C#, структурам запрещается иметь конструктор по умолчанию.
- Используйте структуры для упрощения списков параметров методов.
- Отличия классов от структур:
 1. Размещение: в области стека (классы – управляемая куча (heap))
 2. Копирование: создаётся отдельная копия объекта, которая после копирования живёт «своей жизнью» (классы – создаётся ссылка на тот же класс (т. н. instance))
 3. Наследование: не разрешается дополнение своими свойствами; от него нельзя наследовать (класс – позволяет, кроме случаев, когда класс создавался с ключевым словом `sealed`, не разрешающим наследование)
 4. Передача параметров: как локальные копии переменных (в классах – как ссылки)
 5. Конструктор: да, кроме конструктора по умолчанию, который не требует параметров (в классах – да, без ограничений)
 6. Освобождение переменной: при выходе за пределы видимости (в классах – во время процесса сборки мусора (garbage collector))

Закрепление материала

- Что такое структура?
- Можно ли создавать структуры без оператора `new`?
- Какие есть ограничения при использовании структур?
- Назовите основные отличия классов от структур.

- Почему структуры могут наследоваться только от интерфейсов.
- Можно ли в структуре создавать конструкторы?

Дополнительное задание

Задание

Используя Visual Studio, создайте проект по шаблону Console Application.

Создайте структуру с именем - **Notebook**.

Поля структуры: модель, производитель, цена.

В структуре должен быть реализован конструктор для инициализации полей и метод для вывода содержимого полей на экран.

Самостоятельная деятельность учащегося

Задание 1

Выучите основные конструкции и понятия, рассмотренные на уроке.

Задание 2

Используя Visual Studio, создайте проект по шаблону Console Application.

Требуется: Описать структуру с именем **Train**, содержащую следующие поля: название пункта назначения, номер поезда, время отправления.

Написать программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив, состоящий из восьми элементов типа **Train** (записи должны быть упорядочены по номерам поездов);
- вывод на экран информации о поезде, номер которого введен с клавиатуры (если таких поездов нет, вывести соответствующее сообщение).

Задание 3

Создайте класс **MyClass** и структуру **MyStruct**, которые содержат в себе поля **public string change**.

В классе **Program** создайте два метода:

- **static void ClassTaker(MyClass myClass)**, который присваивает полю **change** экземпляра **myClass** значение «изменено».
- **static void StruktTaker(MyStruct myStruct)**, который присваивает полю **change** экземпляра **myStruct** значение «изменено».

Продемонстрируйте разницу в использовании классов и структур, создав в методе **Main()** экземпляры структуры и класса. Измените значения полей экземпляров на «не изменено», а затем вызовите методы **ClassTaker** и **StruktTaker**. Выведите на экран значения полей экземпляров. Проанализируйте полученные результаты.

Задание 4

Зайдите на сайт MSDN.

Используя поисковые механизмы MSDN, найдите самостоятельно описание темы по каждому примеру, который был рассмотрен на уроке, так, как это представлено ниже, в разделе «Рекомендуемые ресурсы», описания данного урока. Сохраните ссылки и дайте им короткое описание.

Рекомендуемые ресурсы

MSDN: Структуры **struct** (Руководство по программированию на C#)

<http://msdn.microsoft.com/ru-ru/library/saxz13w4.aspx>

MSDN: Классы и структуры (Руководство по программированию на C#)

<http://msdn.microsoft.com/ru-ru/library/ms173109.aspx>