

Перегрузка операторов

№ урока: 16 Курс: C# Essential

Средства обучения: Компьютер с установленной Visual Studio

Обзор, цель и назначение урока

Рассмотрение базового класса `object`.
Рассмотрение техники клонирования.
Рассмотрение перегрузки операторов.

Изучив материал данного занятия, учащийся сможет:

- Понимать и использовать базовый класс `object`.
- Понимать и использовать технику клонирования.
- Понимать и использовать технику перегрузки операторов.

Содержание урока

1. Базовый класс `object`.
2. Клонирование.
3. Паттерн проектирования – Prototype.
4. Интерфейс `ICloneable`.
5. Перегрузка операторов.
6. Перегрузка операторов преобразования значений типов.

Резюме

- В C# все типы, предопределенные и пользовательские, ссылочные типы и типы значений, наследуют непосредственно или косвенно от `Object`. Переменным типа `object` можно назначать значения любых типов.
- Обычно в языках программирования не требуется объявлять класс наследником `Object`, так как наследование происходит неявно.
- Так как все классы в платформе .NET Framework являются производными класса `Object`, все методы, определенные в классе `Object`, доступны для всех объектов в системе. В производных классах некоторые из этих методов, включая перечисленные ниже, могут переопределяться и переопределяются:
 1. `Equals` – данный метод поддерживает сравнение объектов.
 2. `Finalize` – данный метод выполняет операции очистки перед автоматической утилизацией объекта.
 3. `GetHashCode` – данный метод создает число, соответствующее значению объекта, обеспечивающего возможность использования хэш-таблицы. Чтобы переопределить данный метод, необходимо переопределить и метод `Equals`
 4. `ToString` – создает понятную для пользователя строку текста, в которой описывается экземпляр класса.
- Метод `MemberwiseClone` создает "неполную" копию объекта. При этом копируются члены, но не объекты, на которые ссылаются эти члены.
- **Прототип** – паттерн, порождающий объекты. Он определяет, задает виды создаваемых объектов с помощью интерфейса некоторого экземпляра-прототипа, и создает новые объекты путем копирования (клонирования) этого экземпляра. Прототип – это единственный паттерн из серии «порождающих паттернов», который для создания новых объектов использует не явное инстанцирование, а клонирование.
- Интерфейс `ICloneable` поддерживает копирование, при котором создается новый экземпляр класса с тем же значением, что и у существующего экземпляра.

- Реализовав интерфейс **ICloneable**, можно создать все условия для копирования объекта. Интерфейс **ICloneable** содержит один член, **Clone**, предназначенный для поддержки копирования помимо выполняемого с помощью метода **MemberwiseClone**.
- **Оператор** – это элемент программы, который применяется к одному или нескольким *операндам* в выражении или операторе. Операторы, получающие на вход один операнд, например оператор инкремента (**++**) или **new**, называются *унарными* операторами. Операторы, получающие на вход два операнда, например, арифметические операторы (**+**, **-**, *****, **/**) называются *бинарными*.
- В **C#** пользовательские типы могут перегружать операторы путем определения функций статических членов с помощью ключевого слова **operator**. Не все операторы могут быть перегружены, а для некоторых операторов есть ограничения.
- Операторы сравнения можно перегружать, но только парами: если перегружен оператор **==**, то **!=** также должен быть перегружен. Обратный принцип также действителен и действует для операторов **<** и **>**, а также для **<=** и **>=**.
- Для перегрузки оператора в пользовательском классе нужно создать метод в классе с правильной сигнатурой. Метод нужно назвать "operator X", где X – имя или символ перегружаемого оператора. Унарные операторы имеют один параметр, а бинарные – два. В каждом случае один параметр должен быть такого же типа, как класс или структура, объявивший оператор.

Закрепление материала

- Где и как используется класс **object**?
- Какие члены и включает в себя класс **object**?
- Что такое клонирование и какие его виды бывают?
- В чем разница между поверхностным и глубоким клонированием?
- Где и как используется интерфейс **ICloneable**?
- Какие члены и включает в себя интерфейс **ICloneable**?
- Назовите основных участников паттерна Prototype и связи отношений между ними.
- Что такое перегрузка операторов?
- Как выполняется перегрузка операторов?

Дополнительное задание

Задание

Используя Visual Studio, создайте проект по шаблону Console Application.

Создайте структуру описывающую точку в трехмерной системе координат. Организуйте возможность сложения двух точек, через использование перегрузки оператора **+**.

Самостоятельная деятельность учащегося

Задание 1

Выучите основные конструкции и понятия, рассмотренные на уроке.

Задание 2

Создайте класс **Block** с 4-мя полями сторон, переопределите в нем методы:

Equals – способный сравнивать блоки между собой,

ToString – возвращающий информацию о полях блока.

Задание 3

Создайте класс **House** с двумя полями и свойствами.

Создайте два метода **Clone()** и **DeepClone()**, которые выполняют поверхностное и глубокое копирование соответственно. Реализуйте простой способ проверки.

Задание 4

Создайте класс, который будет содержать информацию о дате (день, месяц, год). С помощью механизма перегрузки операторов, определите операцию разности двух дат (результат в виде количества дней между датами), а также операцию увеличения даты на определенное количество дней.

Задание 5

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: `object` (Справочник по C#)

<http://msdn.microsoft.com/ru-ru/library/9kkx3h3c.aspx>

MSDN: класс `object`

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

MSDN: `ICloneable` – интерфейс

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

MSDN: Перегружаемые операторы (Руководство по программированию на C#)

<http://msdn.microsoft.com/ru-ru/library/8edha89s.aspx>