**Практическое занятие № 16. Создание классов**

**1 Цель занятия**

Получить практические навыки создания классов в приложении Windows Forms

**2 Перечень оборудования и программного обеспечения**

Персональный компьютер

Microsoft Office (Word)

Microsoft Visual Studio

**3 Краткие теоретические сведения**

**3.1 Классы и объекты**

***Класс*** является основой для создания объектов. ***Объекты*** являются экземплярами класса.

При определении класса объявляются ***данные***, которые он содержит, и ***код (методы)***, работающие с этими данными. Данные и методы, составляющие класс, называются ***членами класса***. Данные содержатся в переменных и константах экземпляра, которые определены классом, а код содержится в методах.

Также в C# определены несколько специфических разновидностей членов класса. Это — *конструкторы*, *деструкторы*, *индексаторы*, *события*, *операторы* и *свойства*.

Синтаксис класса:

**class** <имя\_класса>

**{**

[<спецификатор>] <тип> <имя\_переменной1>;

[<спецификатор>] <тип> <имя\_переменной2>;

…

[<спецификатор>] <тип> <имя\_метода1> ([<параметры>]) {…}

…

**}**

где <спецификатор> может быть public, private, protected, internal.

Члены класса с типом доступа public доступны везде за пределами данного класса, с типом доступа protected – внутри членов данного класса и производных, с типом доступа private - только для других членов данного класса. Тип доступа internal применяется для типов, доступных в пределах одной сборки.

Создание объекта (экземпляра) класса:

<имя\_класса> <имя\_объекта> = new <имя\_класса>();

При создании объекта класса происходит вызов соответствующего конструктора класса.

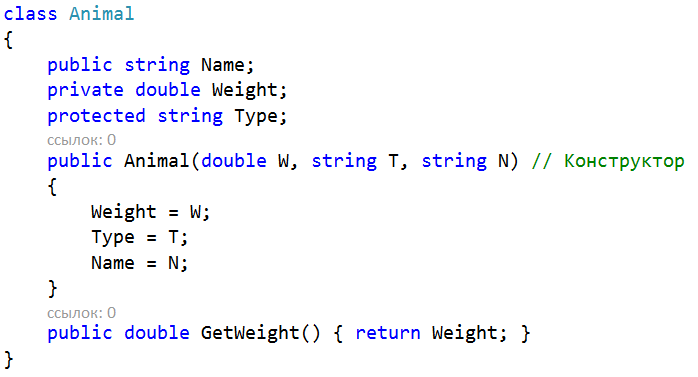
**3.2 Конструктор и деструктор**

***Конструктор класса*** – метод для инициализации объекта при его создании. Он имеет то же имя, что и его класс. В отличие от методов в конструкторах тип возвращаемого значения не указывается явно:

<имя\_класса> (<параметры>) {…}

Конструкторы используются для присваивания начальных значений переменным экземпляра, определенным классом, и для выполнения любых других процедур инициализации, необходимых для создания объекта. В классе могут быть определены несколько конструкторов. Все классы имеют конструкторы независимо от того, определен он или нет. По умолчанию в С# предусмотрено наличие конструктора, который присваивает нулевые значения всем переменным экземпляра (для переменных обычных типов) и значения null (для переменных ссылочного типа). Но если конструктор явно определен в классе, то конструктор по умолчанию использоваться не будет.

**Пример 1:**



***Деструктор*** – метод, вызывающийся автоматически при уничтожении объекта класса (непосредственно перед “сборкой мусора”). Деструктор не имеет параметров и возвращаемого значения:

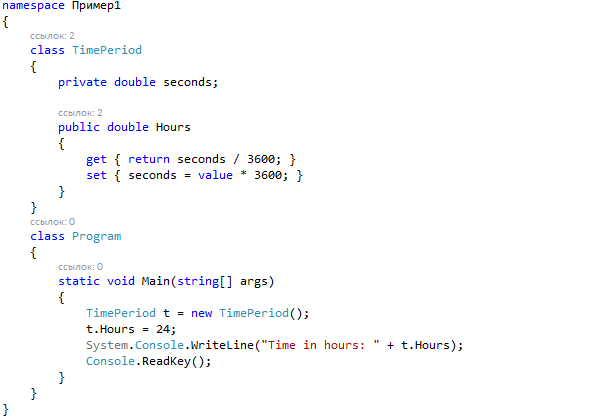
~<имя\_класса> () {…}

**3.3 Свойства**

***Свойство*** — это член класса, предоставляющий гибкий механизм для чтения, записи или вычисления значения частного (private) поля. Свойства можно использовать, как если бы они являлись открытыми членами данных, хотя в действительности они являются специальными методами, называемыми методами доступа. Это обеспечивает простой доступ к данным и позволяет повысить уровень безопасности и гибкости методов.

В примере 2 класс TimePeriod хранит сведения о периоде времени. Внутри класса время хранится в секундах, но свойство с именем Hours позволяет клиенту задать время в часах. Методы доступа для свойства Hours выполняют преобразование между часами и секундами.

**Пример 2:**

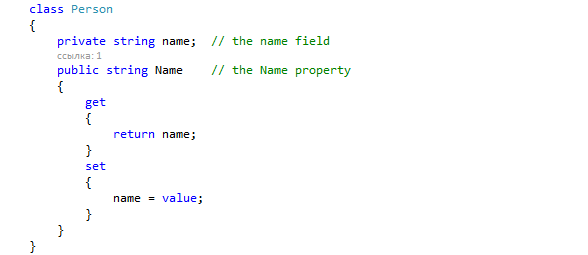


Метод доступа свойства get используется для *возврата значения свойства*, а метод доступа set используется для *назначения нового значения*.

Основная часть метода доступа get похожа на основную часть любого метода. Он должен возвращать значение, имеющее тип заданного свойства. Для этого метод доступа get должен заканчиваться оператором return.

Метод доступа set похож на метод, имеющий тип возвращаемого значения void. В нем используется неявный параметр value, тип которого соответствует типу свойства.

**Пример 3:**



Метод доступа get вызывается для чтения значения свойства при создании на него ссылки:



Если свойству Name не было назначено какое-либо значение, это свойство возвращает значение null.

Когда свойству присваивается значение, выполняется вызов метода доступа set с помощью аргумента, предоставляющего новое значение:

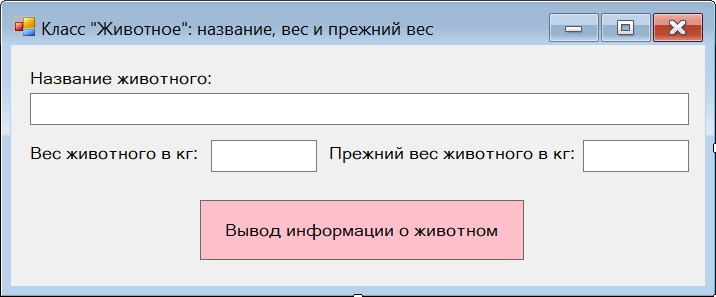


Методы доступа get и set могут иметь различные уровни доступа. Свойства, не реализующие метод доступа set, доступны только для чтения.

**Задание.** Создать класс «Животное» с полями: название, вес и прежний вес.

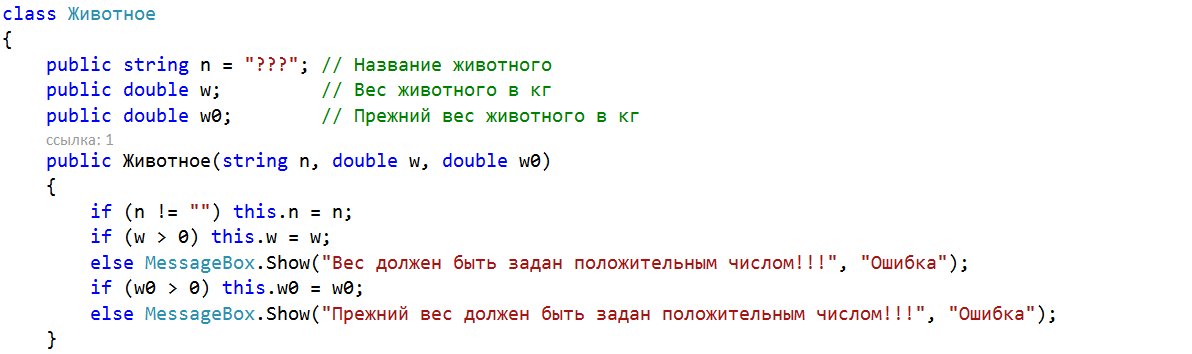
**Решение.**

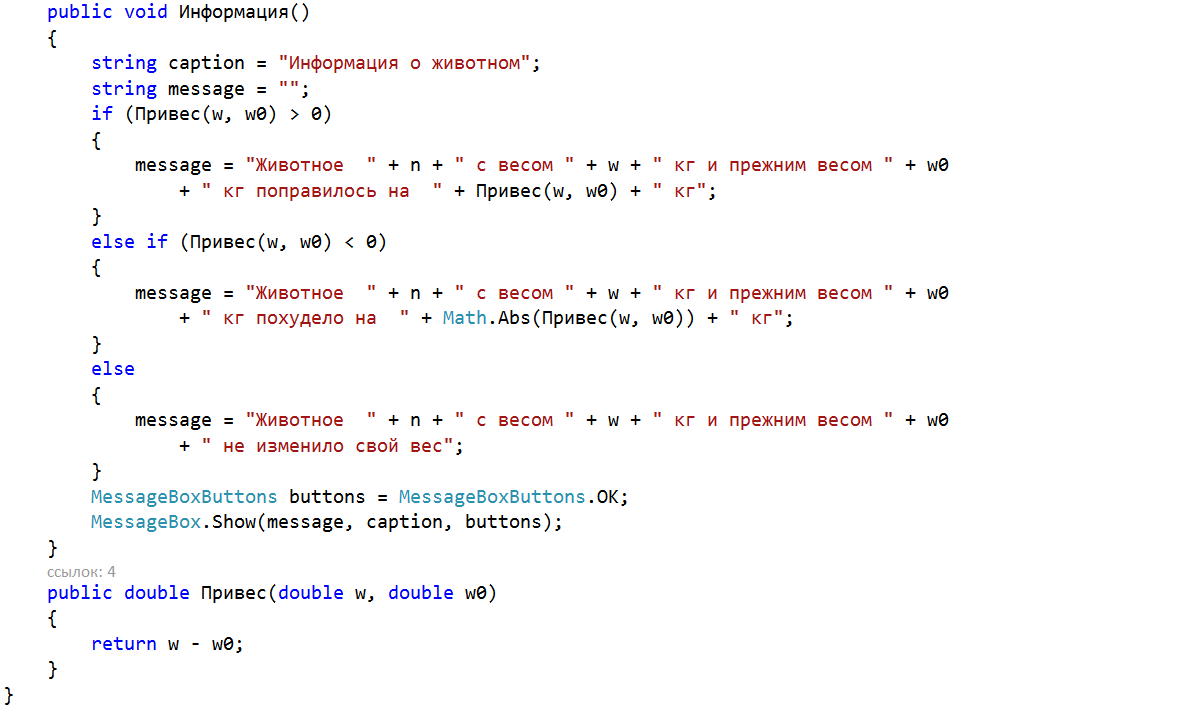
Форма Windows-приложения с элементами, позволяющими отображать и обрабатывать значения полей заданного класса:



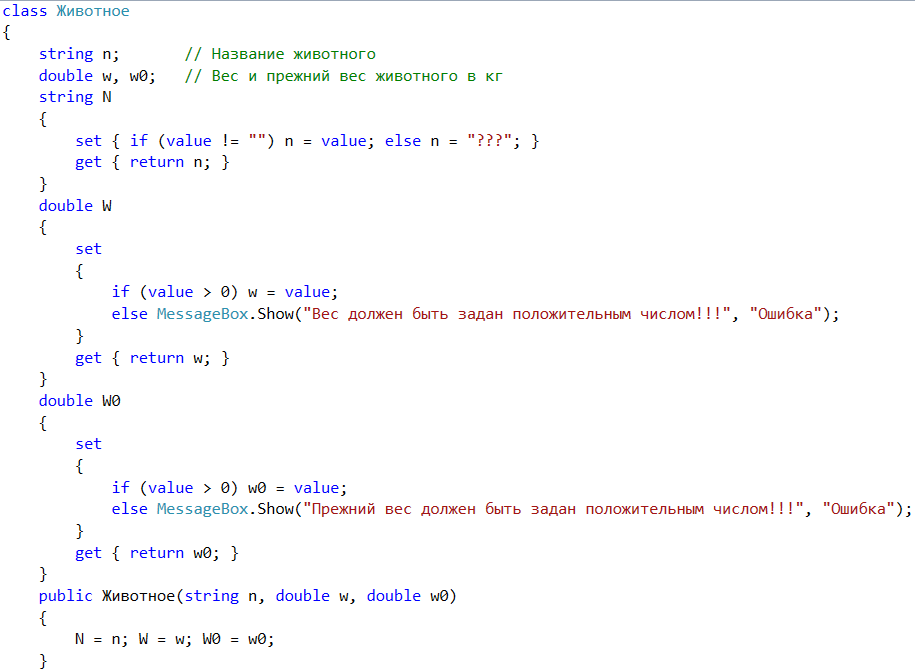
Описание созданного класса (вставить после класса Form)

а) с проверкой корректности задания значений полям класса в конструкторе:



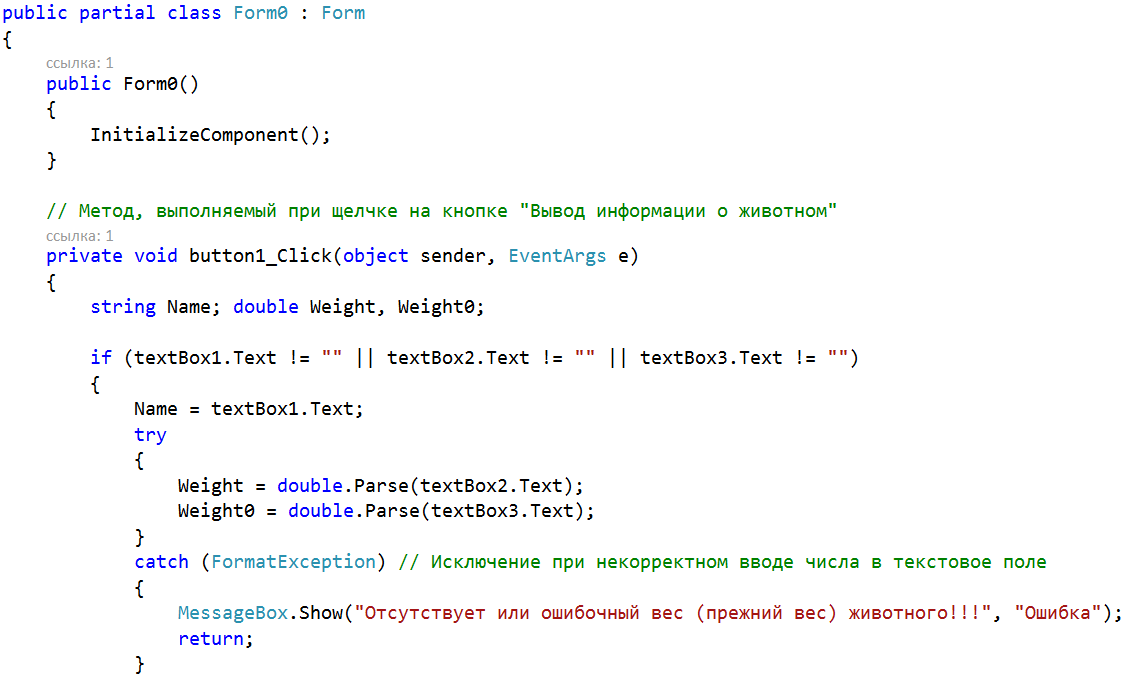


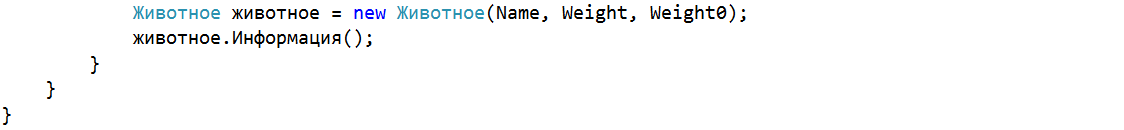
б) с проверкой корректности задания значений полям класса в их свойствах:

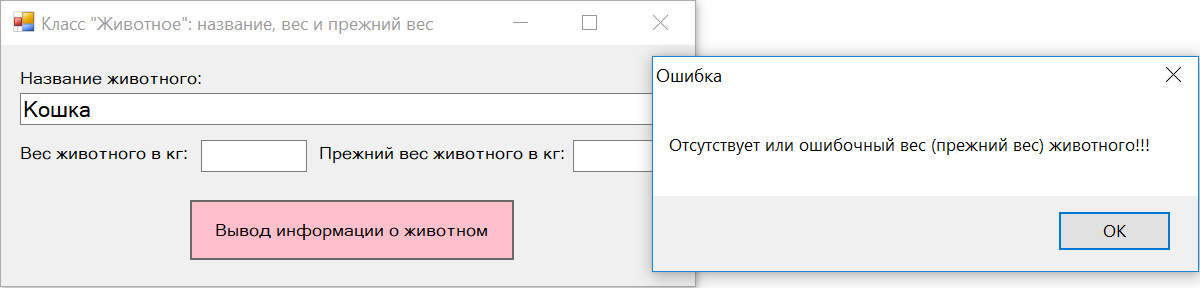


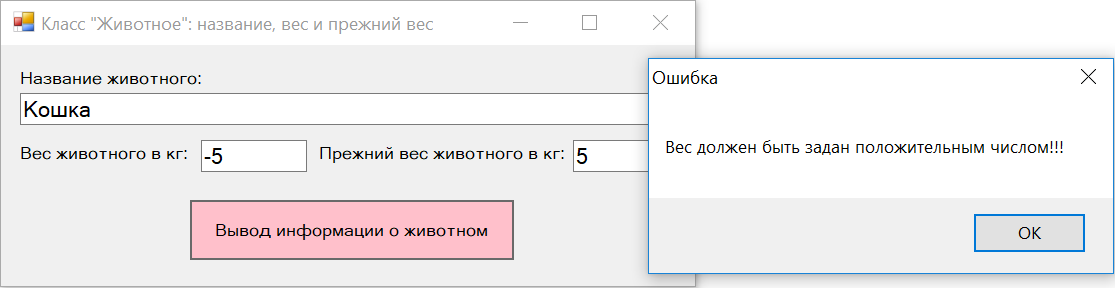
…

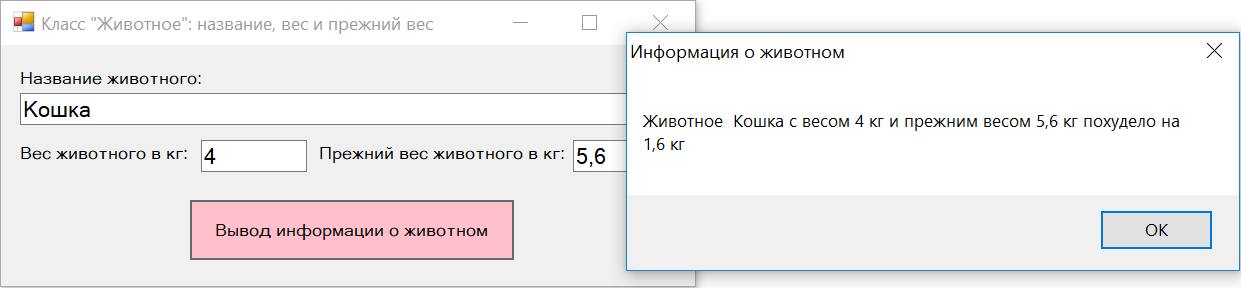
Класс Form с методом button\_Click(), осуществляющим обращение к методам созданного класса:

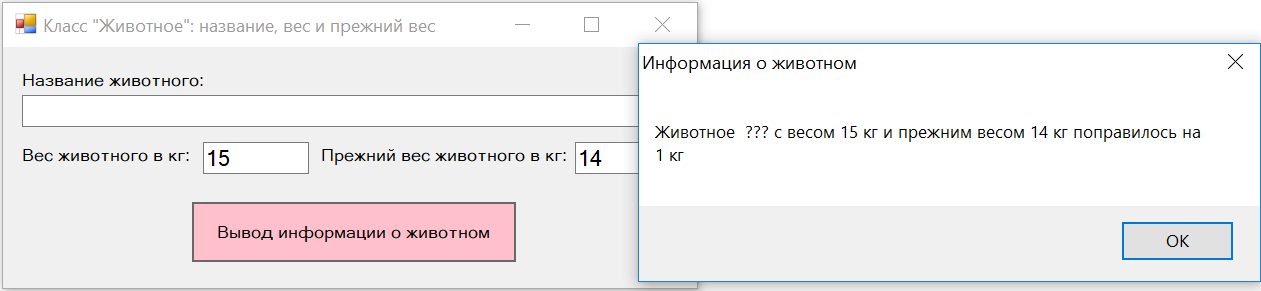












**4 Порядок выполнения работы**

4.1 Изучить теоретические сведения и задание к работе.

4.2 В соответствии с вариантом задания создать форму приложения Windows Forms.

4.3 В соответствии с вариантом задания создать класс с необходимыми данными (полями) и методами.

4.4 В соответствии с вариантом задания разработать и отладить метод обработки данных, осуществляющий обращение к методам созданного класса, используя исключения, выбрасываемые, если форматы или значения входных данных не соответствуют спецификациям параметров обрабатывающих их методов.

**5 Содержание отчета**

5.1 Название работы

5.2 Цель работы

5.3 Программные коды и скрины работы разработанного приложения Windows Forms в соответствии с вариантом задания

**6 Варианты заданий**

6.1 Класс «Товар» с полями: название, вес, цена.

6.2 Класс «Студент» с полями: фамилия, специальность, группа.

6.3 Класс «Геометрическая фигура» с полями: название (прямоугольник, треугольник, эллипс), ширина, высота.

6.4 Класс «Банк» с полями: название (ВТБ 24, Альфа-банк, Возрождение), национальный рейтинг (AAA, AAA, AA), международный рейтинг (A, A-, BBB).

6.5 Класс «Гражданин» с полями: фамилия, пол, дата рождения.

6.6 Класс «Специальность» с полями: наименование (Компьютерные системы и комплексы, Программирование в компьютерных системах, Информационные системы (по отраслям)), код (09.02.01, 09.02.03, 09.02.04), квалификация специалиста (техник по компьютерным системам, техник-программист, техник по информационным системам).

6.7 Класс «Мебель» с полями: название (Стол интегральный, Стол компьютерный, Конференц-стол), артикул (А-33 R, А-10 R, А-13 П), размер Ш×Г×В (1400×900×750, 900×600×750, 1800×900×750).

6.8 Класс «Город» с полями: название (Москва, Санкт-Петербург, Ростов-на-Дону), численность населения (12 000 000, 5 000 000, 1 114 806), площадь (2550 кв.км, 1439 кв.км, 426,5 кв.км).

6.9 Класс «Дисциплина» с полями: наименование (Архитектура компьютерных систем, Системное программирование, Прикладное программирование), форма итоговой аттестации (экзамен, экзамен, экзамен), максимальная нагрузка (96 часов, 120 часов, 150 часов).

6.10 Класс «ВУЗ» с полями: название (Московский государственный технический университет имени Н.Э. Баумана, Санкт-Петербургский государственный университет, Национальный исследовательский ядерный университет МИФИ), год основания (1830, 1724, 1942), рейтинг (4,321, 4,168, 3,924).

6.11 Класс «Государство» с полями: название (Армения, Беларусь, Россия), столица (Ереван, Минск, Москва), плотность населения (107,9 чел./кв.км, 47,0 чел./кв.км, 8,3 чел./кв.км).

6.12 Класс «Животное» с полями: название (переднеазиатский леопард, тур, выдра), отряд (хищные, парнокопытные, хищные), семейство (кошачьи, полорогие, куньи).

6.13 Класс «Автомобиль» с полями: марка (Toyota, Kia, Mazda), модель (Prius, Rio, CX-9), модификация (1.8, 1.4 хэтчбек, 3.7).

6.14 Класс «Газета» с полями: название (Аргументы и факты, Собеседник, Вечерняя Москва), подписной индекс (40860, 99596, Е0079), периодичность (1 раз в неделю, 1 раз в неделю, 5 раз в неделю).

6.15 Класс «Компьютер» с полями: тактовая частота процессора, объем ОЗУ, емкость диска.

6.16 Класс «Служащий» с полями: табельный номер, номер подразделения и оклад.

6.17 Класс «Накладная» с полями: номер, дата отпуска и сумма.

**7 Используемая литература**

7.1 Г. Гагарина, Е. В. Кокорева, Б. Д. Виснадул Технология разработки программного обеспечения. Форум, Инфра-М, 2009

7.2 Эндрю Троелсен Язык программирования С# 2010 и платформа .NET 4.0, М., ВИЛЬЯМС, 2011

7.3 А. Марченко C#. Введение в программирование, М, Вильямс, 2009

7.4 http://msdn.microsoft.com/ru-ru/library/67ef8sbd.aspx.