# Практическое занятие № 18. Взаимодействие объектов. Использование стандартных и пользовательских атрибутов

**1 Цель занятия**

Получить практические навыки реализации взаимодействия объектов с использованием стандартных и пользовательских атрибутов в приложении Windows Forms

**2 Перечень оборудования и программного обеспечения**

Персональный компьютер

Microsoft Office (Word)

Microsoft Visual Studio

**3 Краткие теоретические сведения**

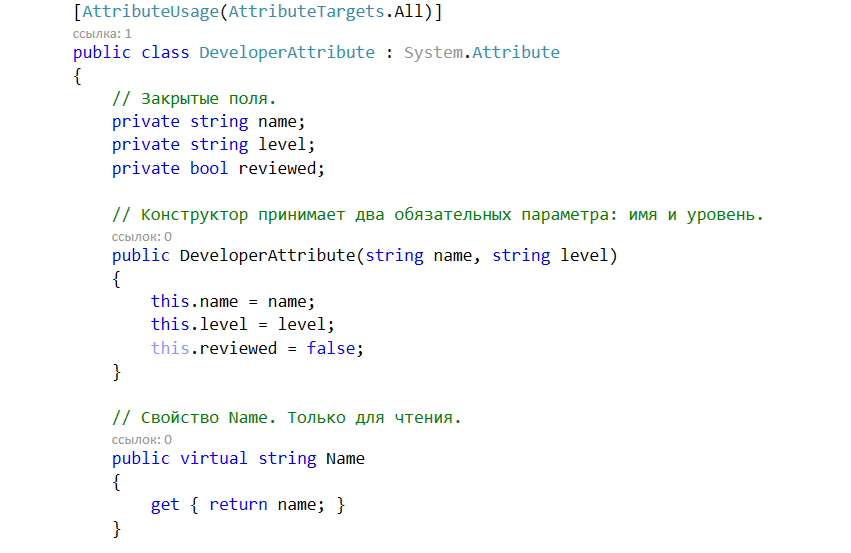
***Атрибут*** — это класс, унаследованный от базового класса Attribute. Суть атрибута в том, что он используется для генерации описаний. Создав атрибут, вы можете наделить его своими свойствами, которыми вы желаете наградить класс, поле, свойство или метод, Атрибут не влияет на значение полей и свойств и на выполнение методов класса, если в теле метода не учитывается информация, хранящаяся в атрибуте. Но есть очень важный момент — значение атрибута невозможно изменять в процессе выполнения кода, так как значения его свойств хранятся в виде констант в скомпилированном модуле.

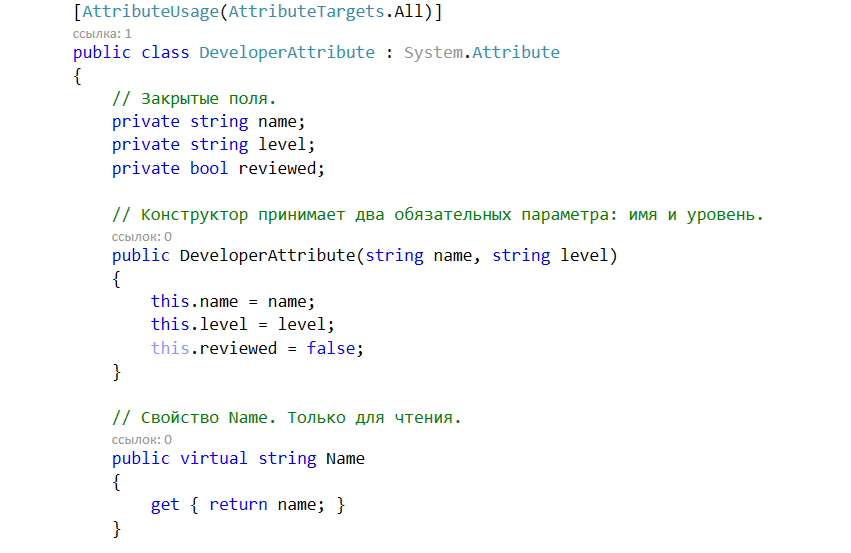
Например, с помощью атрибутов вы можете, проанализировав их зна­чения для типа, построить различные схемы выполнения действий над ним.

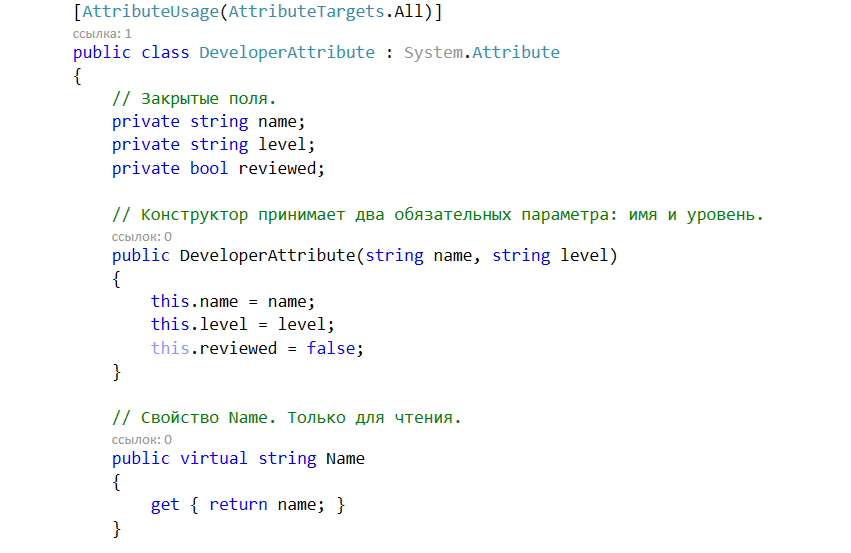
**3.1 Использование пользовательского атрибута**

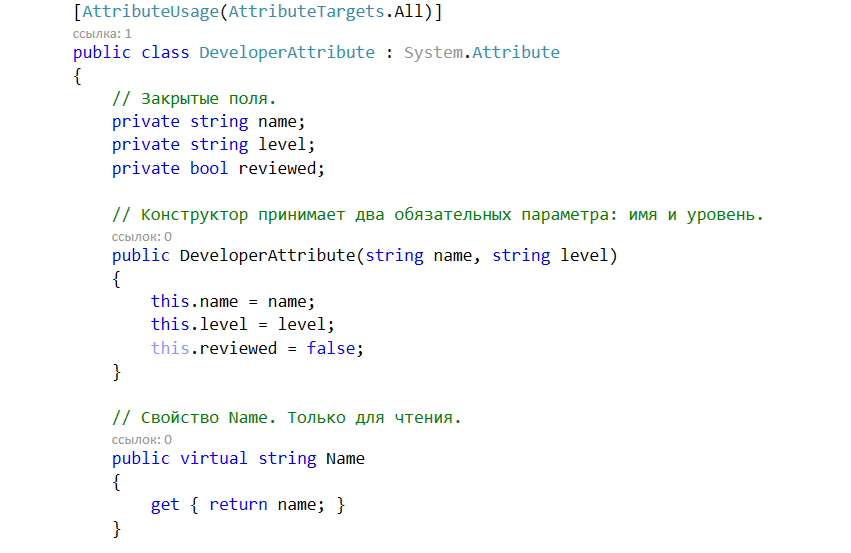
Создадим простой атрибут, документирующий некоторый фрагмент кода. Атрибут из этого примера содержит информацию об имени и уровне программиста, а также о времени последнего пересмотра кода. Он содержит три закрытых переменных, в которых хранятся данные. Каждая переменная связана с открытым свойством для чтения и записи значений. Также имеется конструктор с двумя обязательными параметрами.

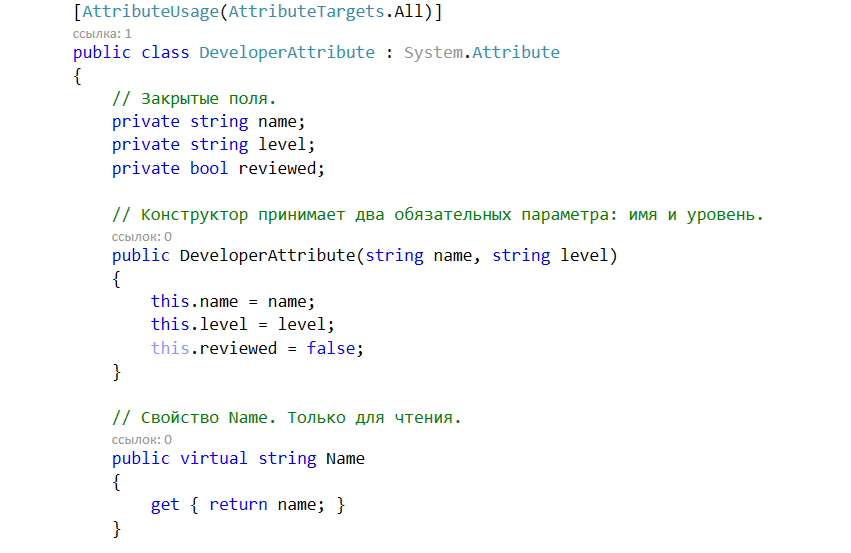
**Пример 1.** Код класса атрибута DeveloperAttribute:

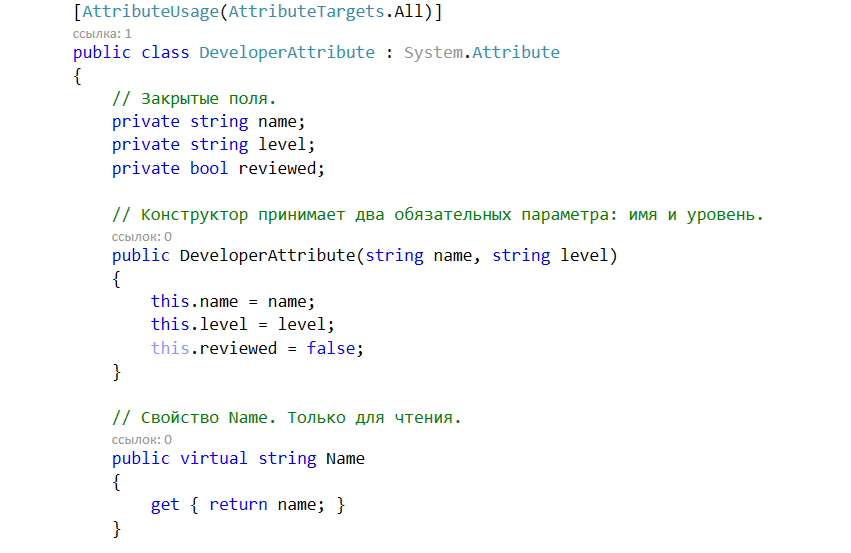
****

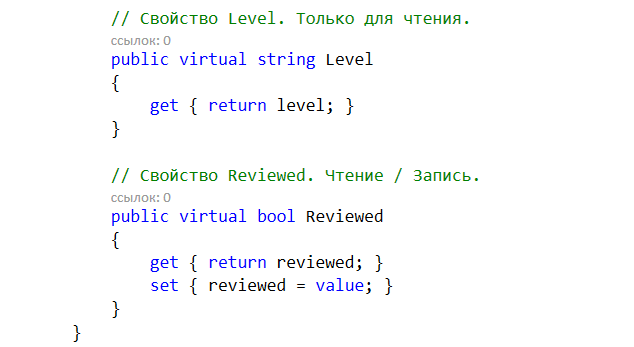
****

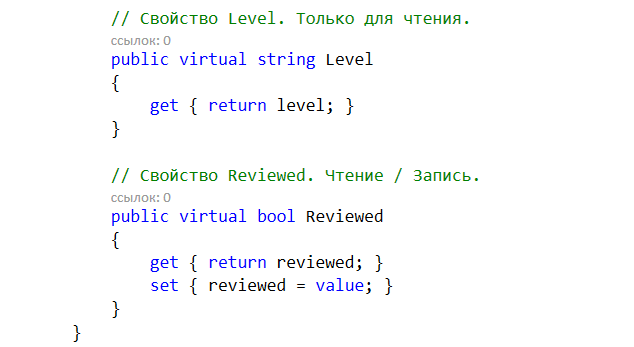
****

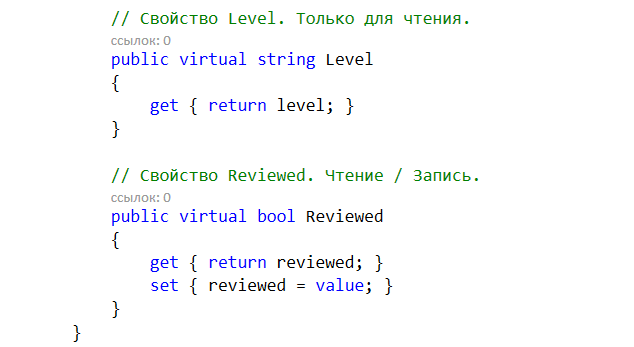
****

****

****

****

****

****

В головном модуле сначала объявляется переменная с типом атрибута, который нужно получить, затем она инициализируется с помощью вызова метода Attribute.GetCustomAttribute(). Теперь можно использовать любые доступные свойства атрибута.

В следующем примере атрибут DeveloperAttribute применяется к классу Program в целом. Метод GetAttribute() использует Attribute.GetCustomAttribute() для получения состояния атрибута DeveloperAttribute перед тем, как вывести информацию на консоль.

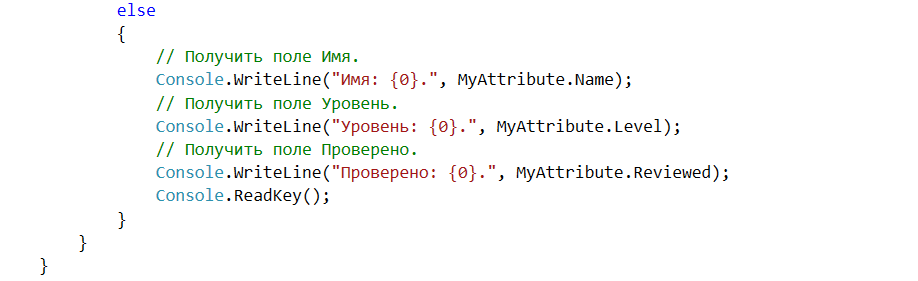
**Пример 2.** Тестирование атрибута DeveloperAttribute в классе Program:

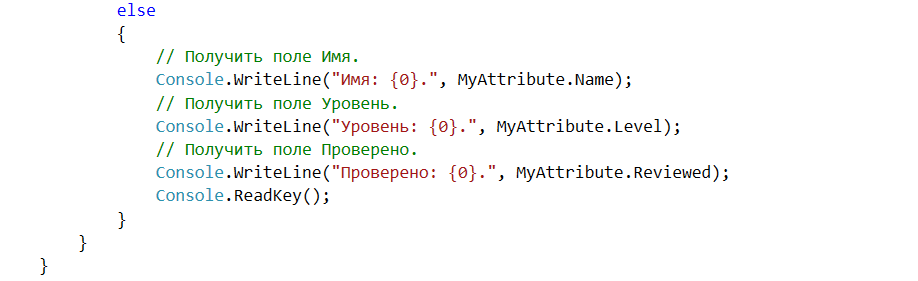










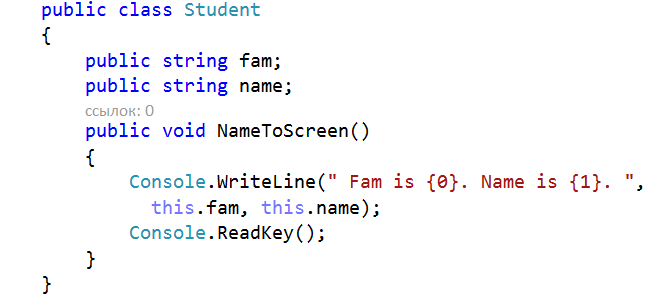


**3.2 Использование предопределенного атрибута** Conditional

Условное выполнение является полезным методом, если вы хотите включить тестирование и отладку кода в проект, но не хотите редактировать проект и удалять код отладки после того, как проект является законченным и функционирует правильно.

Атрибут Conditional будем использовать для выполнения метода класса в зависимости от значения символа под названием DEBUG\_ACCOUNT.

3.2.1. Для применения атрибута Conditional создадим класс Student:



Метод класса NameToScreen() отображает содержимое полей класса: фамилию и имя студента.

3.2.2. Для задания условного использования этого метода в зависимости от символа DEBUG\_ACCOUNT:

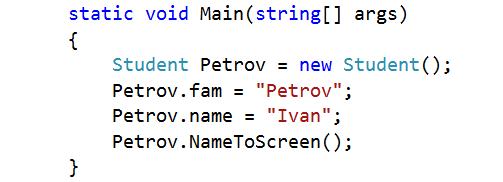
* добавьте перед описанием метода атрибут Conditional:

[Conditional("DEBUG\_ACCOUNT")]

* добавьте using для пространства имен System.Diagnostics.

3.2.3. Для тестирования атрибута Conditional:

* добавьте в Main() код вызова метода NameToScreen():



* запустите тестовую программу.

*Обратите внимание,* что ничего не происходит.

В верхней части файла Program.cs, перед первым использованием директивы, добавьте следующий код:

# define DEBUG\_ACCOUNT

Это определяет символ DEBUG\_ACCOUNT.

Снова запустите тестовую программу. *Обратите внимание*, что метод NameToScreen() начал отображать информацию из класса Student.

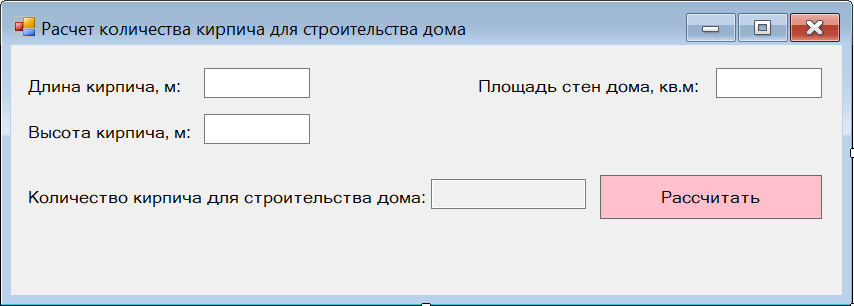
**Задание.** 1. Создать класс (классы) и определить необходимые данные (поля) и методы для расчета количества кирпича разного размера для строительства домов разной площади. Значения данных для расчета должны вноситься пользователем через элементы формы Windows-приложения.

2. Разработать пользовательский атрибут, содержащий данные, которые можно использовать в качестве данных по умолчанию для методов класса, в случае отсутствия их ручного внесения в поля формы.

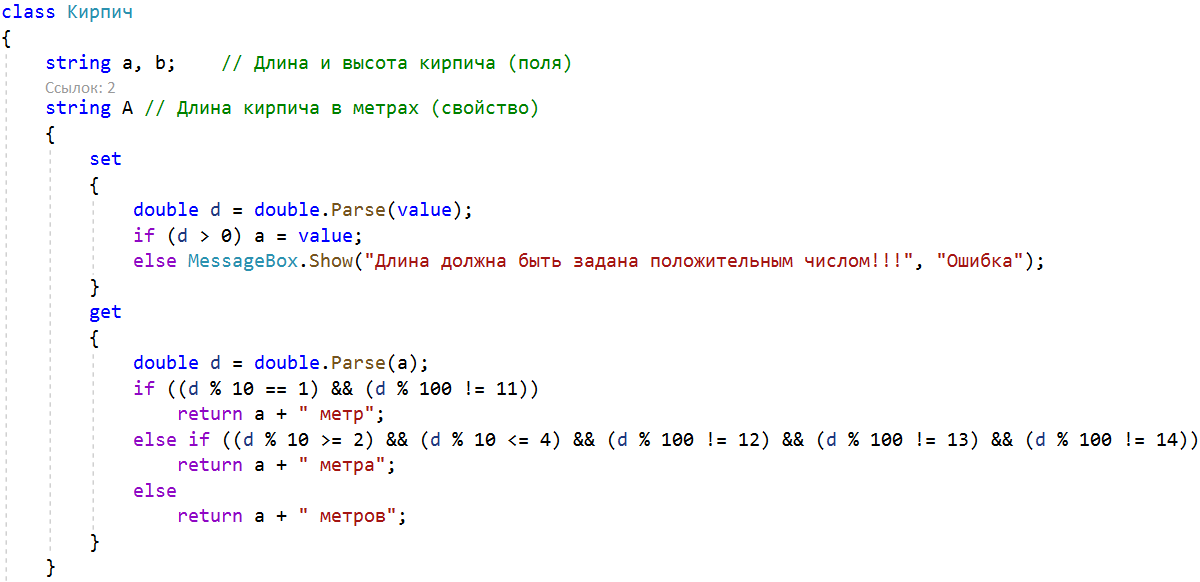
3. Использовать атрибут Conditional для разрешения/запрещения использования разработанного пользовательского атрибута в зависимости от определения символа DEBUG\_ACCOUNT.

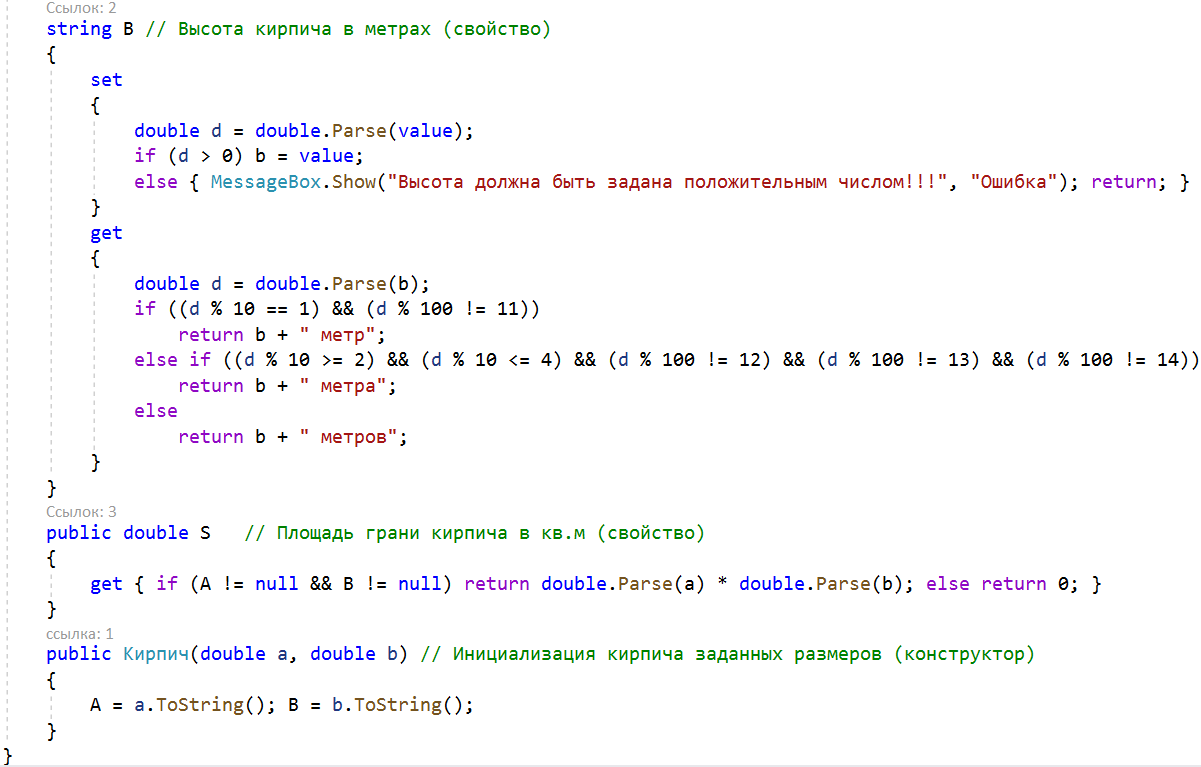
**Решение.**

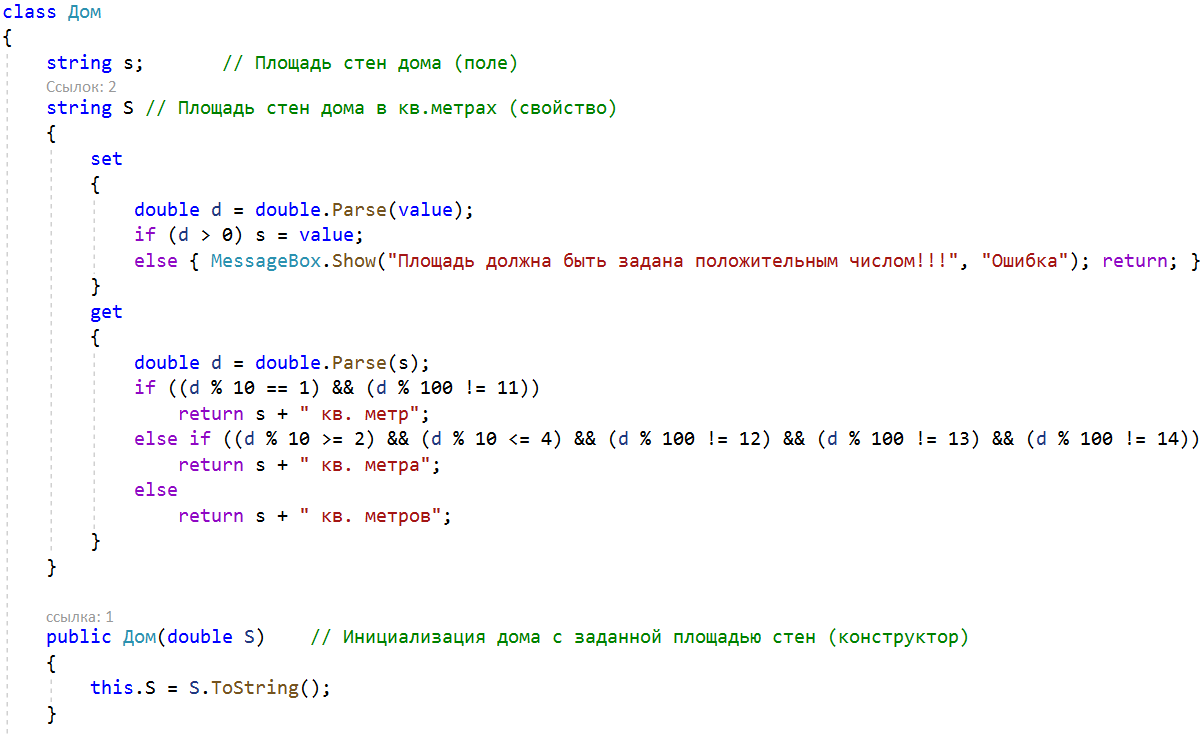
Форма Windows-приложения для проведения заданного расчета:

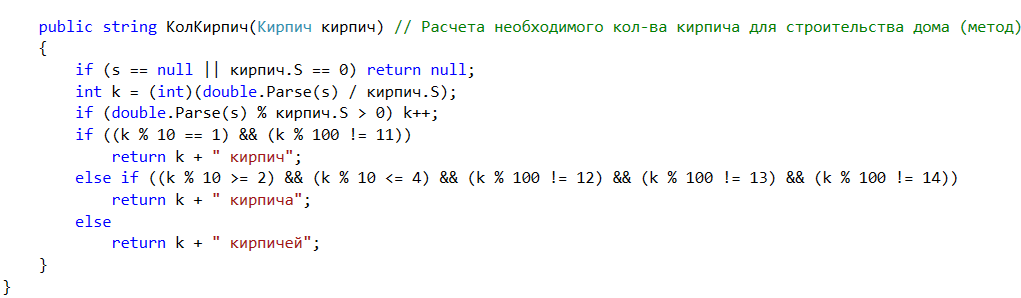


1.1 Код базовых классов с данными (полями и их свойствами), конструкторами и методами обработки данных (после класса Form1):

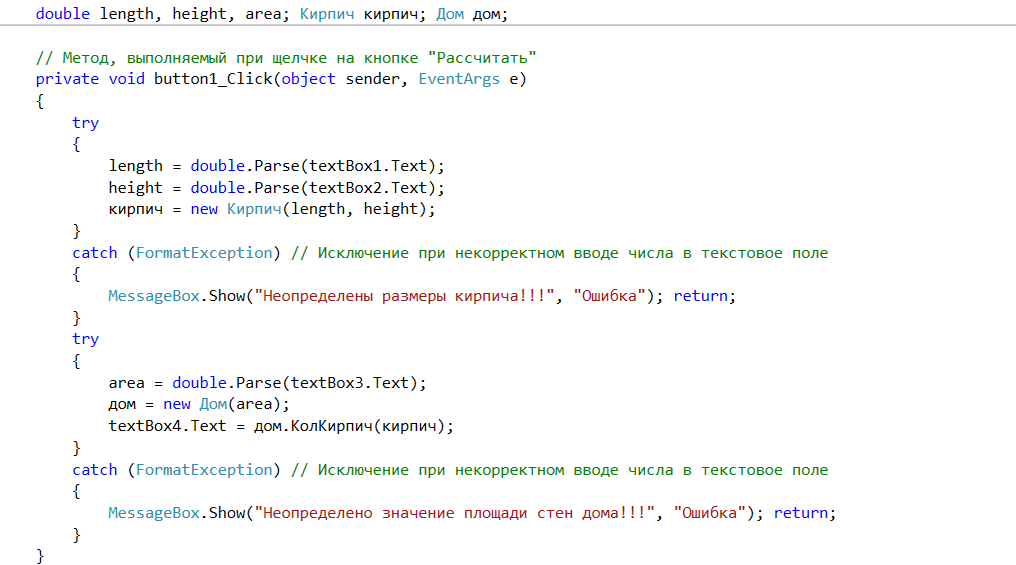


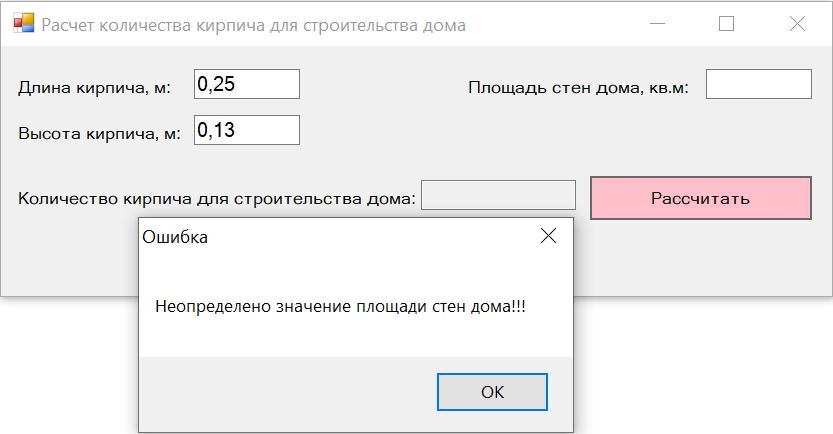


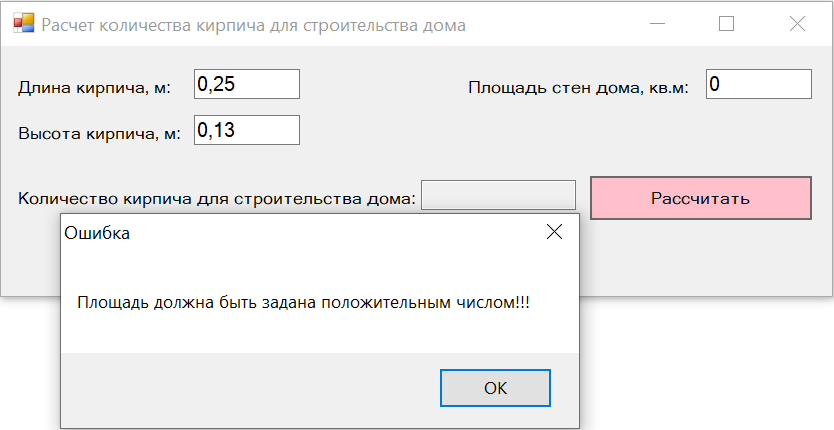


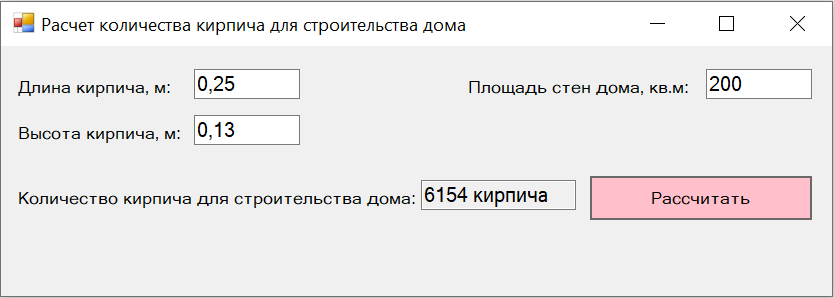


1.2 Код метода button1\_Click() класса Form1, реализующего необходимый расчет при щелчке соответствующей кнопки:

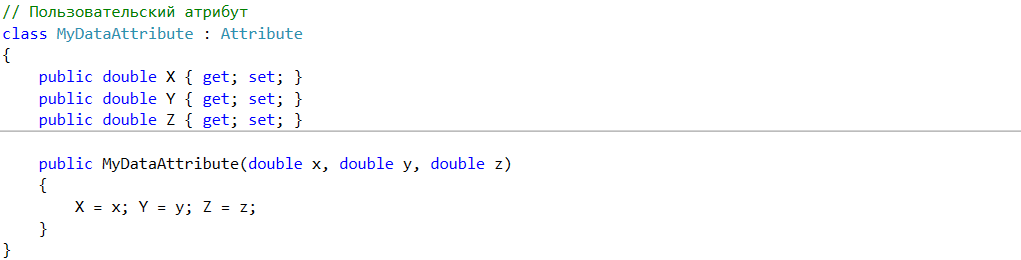








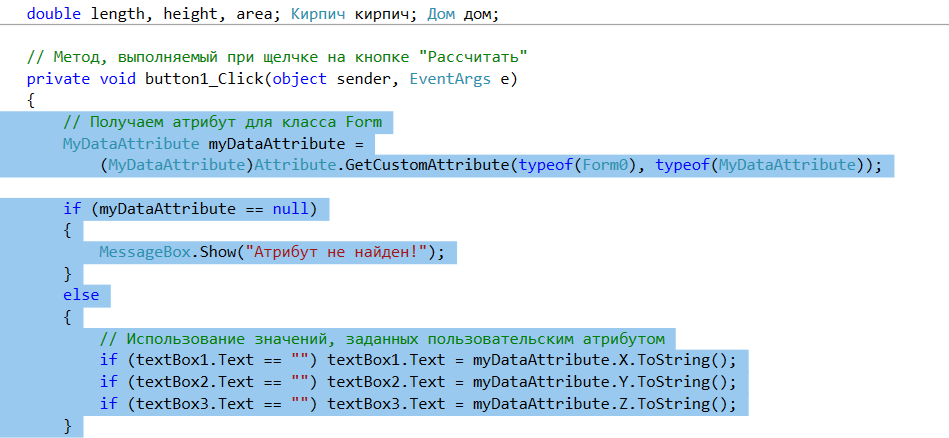
2.1 Код класса-атрибута MyDataAttribute, содержащего данные для установки и заполнения полей формы значениями по умолчанию, и конструктор, инициализирующий эти данные:

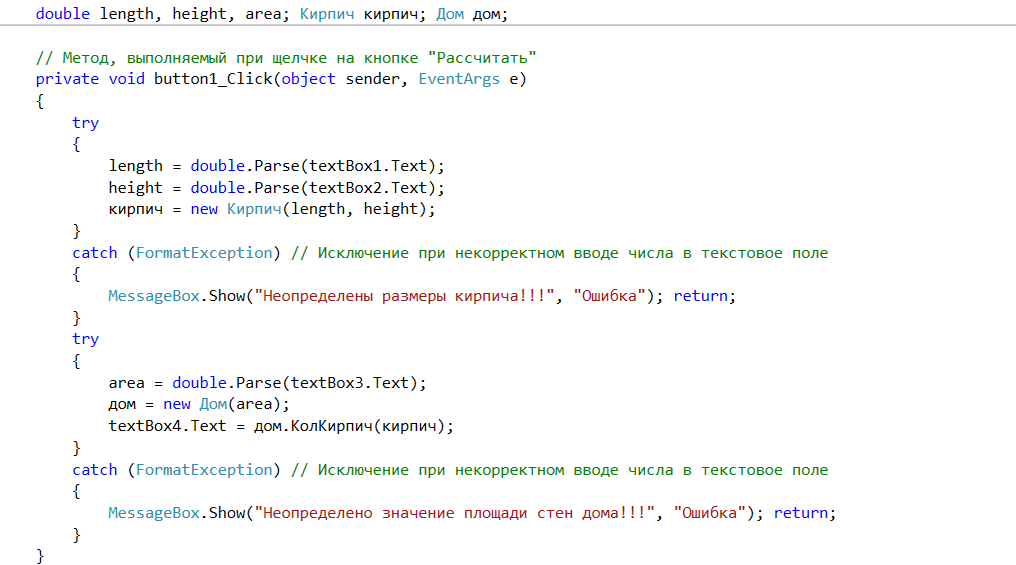


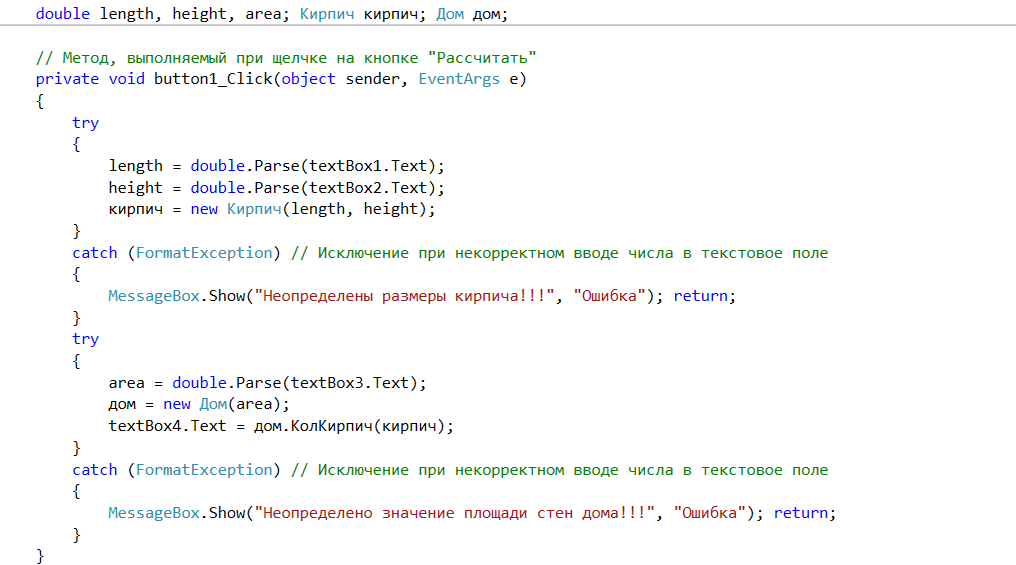
2.2 Инициализация атрибута MyDataAttribute значениями (задана перед описанием класса Form1):

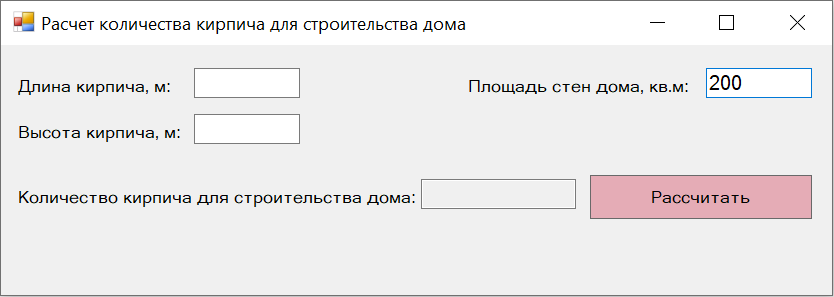


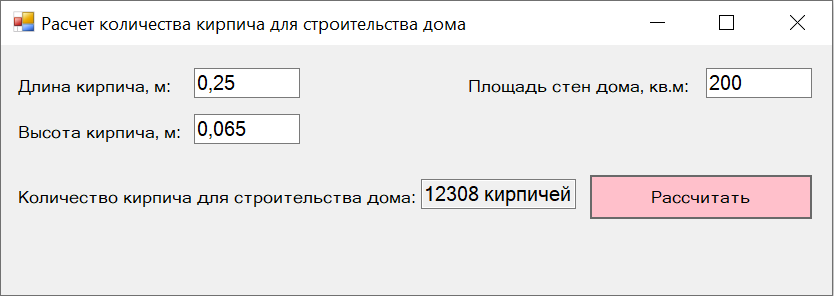
2.3 Модернизированный код метода button1\_Click() класса Form1, осуществляющего, в случае отсутствия ручного внесения данных, заполнение полей формы значениями атрибута:



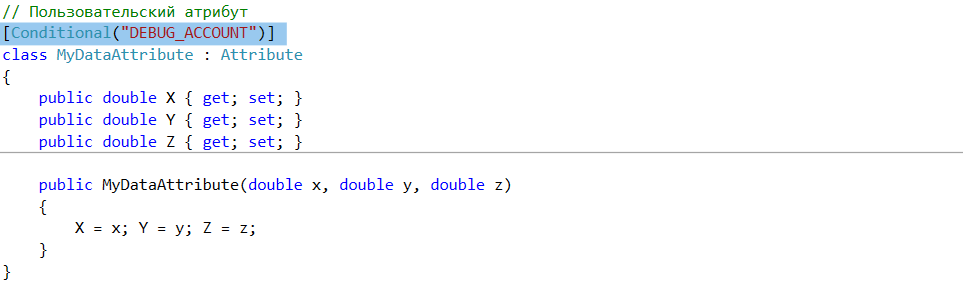


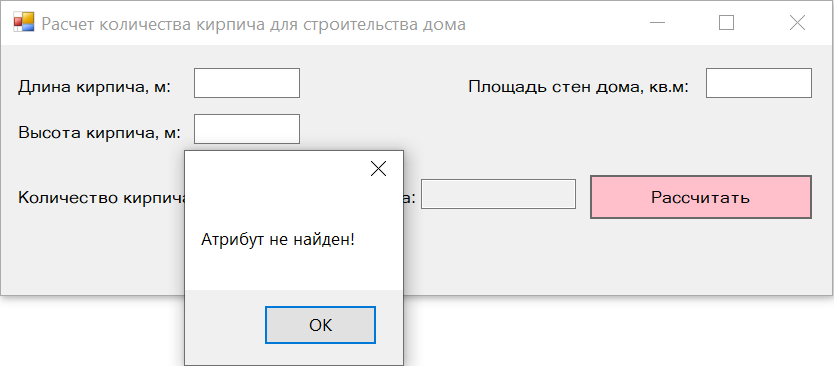


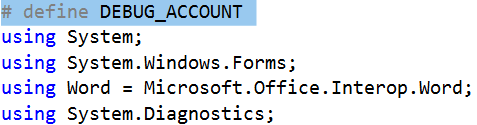


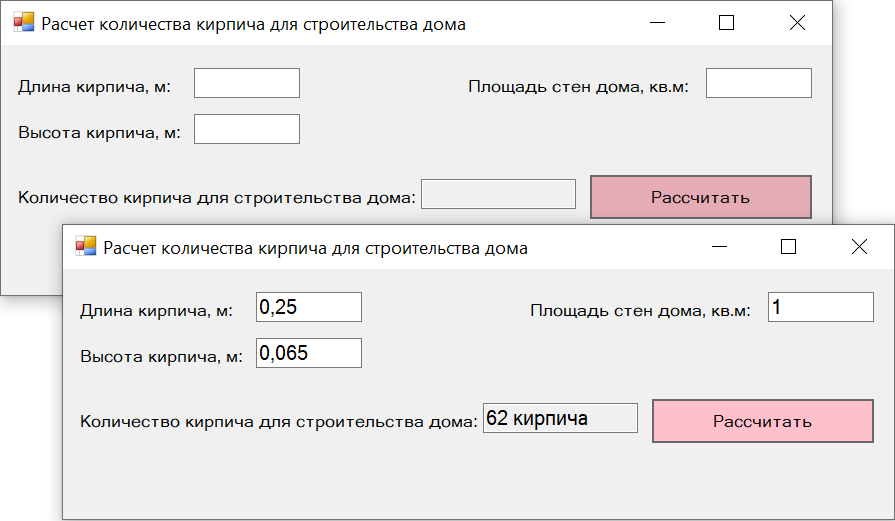


3.1 Инициализация атрибута Conditional значением "DEBUG\_ACCOUNT" (задана перед описанием класса пользовательского атрибута для разрешения/запрещения его использования):









**4 Порядок выполнения работы**

4.1 Изучить теоретические сведения и задание к работе.

4.2 В соответствии с вариантом задания создать класс (классы) и определить в них необходимые для расчета данные (поля и их свойства), конструкторы и методы.

4.3 Разработать и отладить метод обработки данных созданного класса (классов), реализующий заданный расчет при щелчке соответствующей кнопки.

4.4 Разработать пользовательский атрибут, значения данных которого можно использовать в качестве данных по умолчанию для полей созданного класса (классов) в случае отсутствия ручного внесения в них значений через поля формы Windows-приложения.

4.5 Модернизировать код метода, реализующего заданный расчет, для осуществления заполнения полей формы Windows-приложения значениями пользовательского атрибута в случае отсутствия ручного внесения данных.

4.4 Использовать атрибут Conditional для разрешения/запрещения использования разработанного пользовательского атрибута.

**5 Содержание отчета**

5.1 Название работы

5.2 Цель работы

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

**6 Варианты заданий**

6.1 Создать классы Автомобиль и Груз, определить в них необходимые поля и методы для определения количества рейсов автомобилей разной грузоподъемности для перевозки груза заданного веса.

6.2 Создать классы Помещение и Обои, определить в них необходимые поля и методы для определения расхода строительных материалов (обои) разного размера при ремонте помещения. При расчете не учитывать размеры окон и дверей.

6.3 Создать класс Мебель, определить необходимые поля и методы для определения количества необходимых материалов для производства корпусной мебели разного размера и количества. Состав материалов фиксированный (материал основы корпуса, материал кромок корпуса, крепеж).

6.4 Создать класс Блок, определить необходимые поля и методы для определения количества материалов для производства строительных блоков разного размера и количества. Состав материалов фиксированный (вода, цемент, песок, гравий).

6.5 Создать классы Товар и Кредит, определить необходимые поля и методы для определения размера кредита с разными условиями на покупку товаров разного типа.

6.6 Создать классы Плитка и Помещение, определить необходимые поля и методы для определения расхода строительных материалов (напольная плитка) разного размера при ремонте помещения.

6.7 Создать классы Такси и План, определить необходимые поля и методы для определения количества рейсов маршрутного такси разной вместимости для выполнения плана выручки. Продолжительность рейса и стоимость проезда фиксирована, количество пассажиров среднее.

6.8 Создать классы Нефтебаржа и Груз, определить необходимые поля и методы для определения количества рейсов нефтебарж разной грузоподъемности для перевозки груза нефти заданного тоннажа.

6.9 Создать класс Хлеб, определить необходимые поля и методы для определения количества продуктов для выпечки хлеба заданного размера и количества. Состав продуктов фиксированный (мука, вода, масло растительное, дрожжи, соль, сахар).

6.10 Создать классы Краска и Помещение, определить необходимые поля и методы для определения расхода строительных материалов (краска для потолка) разного объема при ремонте помещения.

6.11 Создать классы Доска и Брус, определить необходимые поля и методы для расчета количества досок разного размера, которое выйдет из четырехугольного бруса.

6.12 Создать класс Окно, определить необходимые поля и методы для определения количества материалов для производства металлопластиковых окон разного размера и количества. Состав материалов фиксированный (профиль, стекло).

6.13 Создать класс Лошадь, определить необходимые поля и методы для расчета количества корма для содержания лошадей разного возраста на конюшне (жеребенку от полугода да года требуется в год сена ≈1300 кг, концентратов ≈600 кг, корнеплодов ≈500 кг, зеленого корма ≈3600 кг; молодняку до 2-х лет ‑ сена ≈2200 кг, концентратов ≈1800 кг, корнеплодов ≈500 кг, зеленого корма ≈5000 кг; взрослой лошади до 3-х лет ‑ сена ≈2500 кг, концентратов ≈2100 кг, корнеплодов ≈500 кг, зеленого корма ≈7200 кг; взрослой лошади от 3-х лет ‑ сена ≈2700 кг, концентратов ≈2100 кг, корнеплодов ≈500 кг, зеленого корма ≈7200 кг).

6.14 Создать классы Спецодежда и Ткань, определить необходимые поля и методы для расчета количества материалов для пошива спецодежды разного размера. Состав материалов фиксированный (ткань, нитки, пуговицы).

6.15 Создать классы Семена и Площадка, определить необходимые поля и методы для расчета веса семян, которого хватит для засева в парке газонной травкой площадок, имеющих форму треугольников разных размеров.

6.16 Создать классы Продукты и Поход, определить необходимые поля и методы для расчета веса продуктов, которые должны быть закуплены для туристических походов разной длительности и с различным количеством участников.

6.17 Создать класс Пациент и Лекарство, определить необходимые поля и методы для расчета количества упаковок лекарства, которое необходимо купить в аптеке для прохождения курса лечения пациентами разного возраста.

**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.