**Группа** ИС 3-1

**ФИО:** Мозыркий А.Д.

**Курс:** C# Professional

**Урок 1:** Пользовательские коллекции

**Теория**

* Коллекция представляет собой набор объектов схожих типов, сгруппированных вместе.
* Емкость коллекции — это число элементов, которое она может содержать.
* Количество элементов коллекции — это число элементов, которое она реально содержит.
* Нижняя граница коллекции — это индекс ее первого элемента.
* Перечислитель — это объект, который выполняет итерацию в связанной с ним коллекции. Можно считать, что он является перемещаемым указателем на любой элемент коллекции. Перечислитель может быть связан только с одной коллекцией, но коллекция может иметь несколько перечислителей. Оператор foreach использует перечислитель и упрощает обращение с коллекцией.
* Оператор foreach предполагает, что все элементы коллекции имеют один и тот же тип.
* Все перечислители основаны на интерфейсах IEnumerator и IEnumerable или на их обобщенных аналогах IEnumerator<T> и IEnumerable<T>.
* Ключевое слово yield сообщает компилятору, что метод, в котором оно содержится, является блоком итератора. Для реализации поведения, определенного в блоке итератора, компилятор создает класс.
* В блоке итератора ключевое слово yield используется совместно с ключевым словом return для предоставления значения объекту перечислителя, например значения, возвращаемого в каждом цикле оператора foreach.
* Ключевое слово yield всегда используется вместе с ключевым словом break для обозначения конца итерации.
* Интерфейс ICollection определяет размер, перечислители и методы синхронизации для всех нестандартных коллекций. Интерфейс ICollection — это базовый интерфейс для классов в пространстве имен System.Collections, а интерфейс ICollection<T> является базовым интерфейсом для классов в пространстве имен System.Collections.Generic.
* Интерфейс ICollection расширяет интерфейсы IEnumerable, IDictionary и IList и является более специализированным интерфейсом.
* Реализация IList является коллекцией и доступ к ее элементам может быть осуществлен с помощью индекса, например, как в классе ArrayList.
* Iterator – паттерн поведения объектов, предоставляющий последовательный доступ ко всем элементам составного объекта, не раскрывая его внутреннего представления. Данный паттерн допустимо применять для доступа к содержимому агрегированных объектов без раскрытия их внутреннего представления, поддержки нескольких видов активных обходов одного и того же агрегированного объекта, или для предоставления единообразного интерфейса с целью обхода различных агрегированных структур (то есть, для поддержки полиморфной итерации).

**Практика**

**Задание 1**

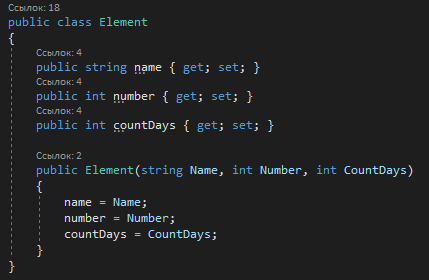


Рис.1.1 Реализация класса Element

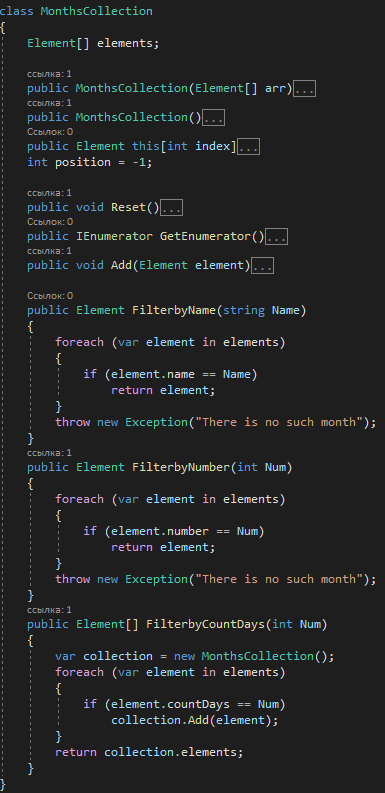


Рис.1.2 Реализация коллекции MonthCollection

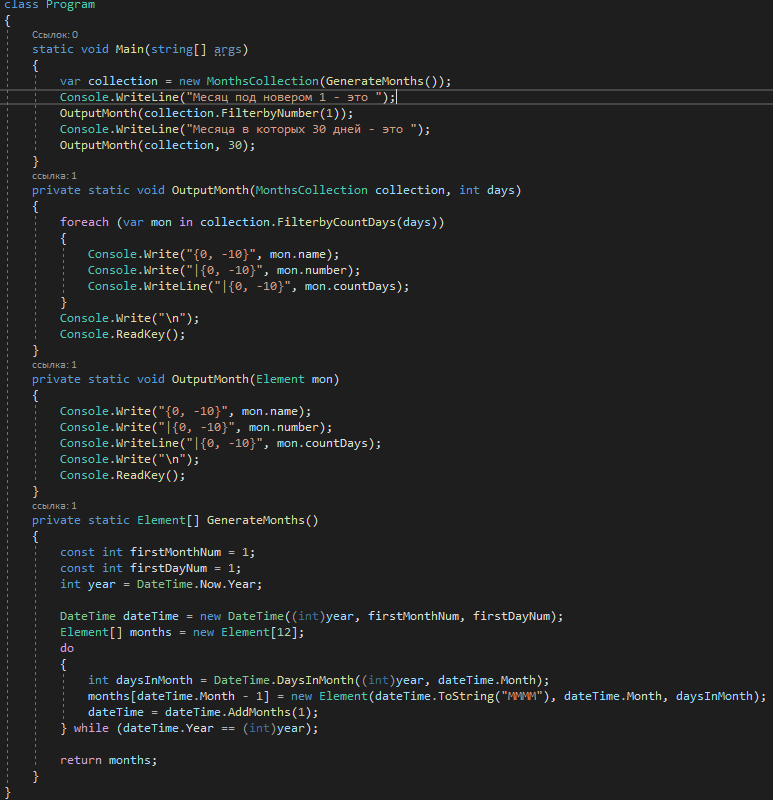


Рис.1.3 Реализация кода программы

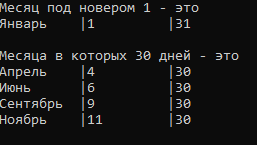


Рис.1.4 Результат работы программы

Результат работы программы соответствует ожидаемому.

**Задание 2**

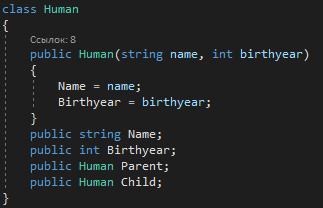
****

Рис.2.1 Реализация класса Human

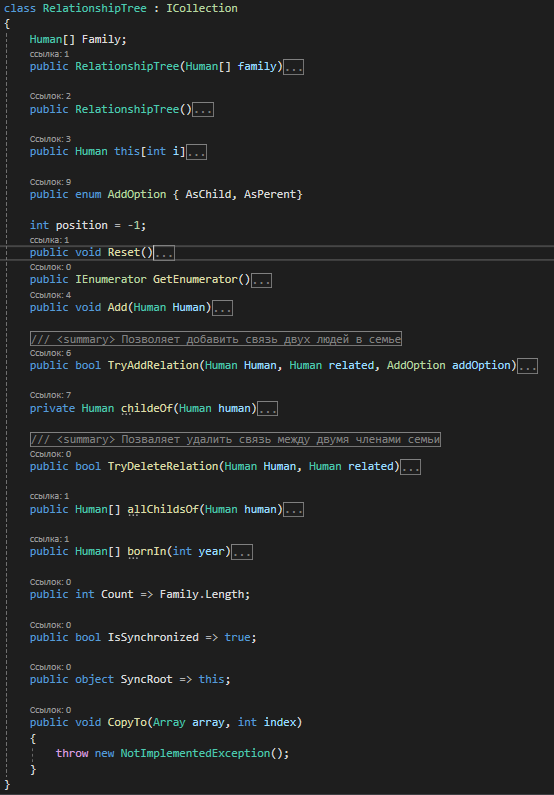


Рис.2.2 Реализация коллекции RelationshipTree



Рис.2.3 Реализация кода программы

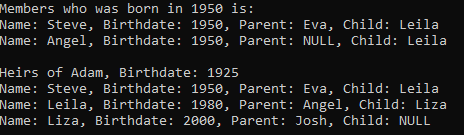


Рис.2.4 Результат работы программы

Результат работы программы соответствует ожидаемому.

**Задание 3**

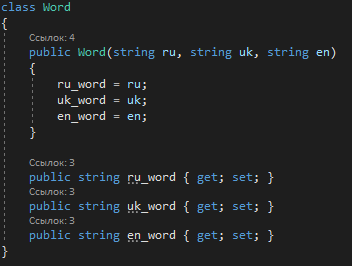


Рис.3.1 Реализация класса Word

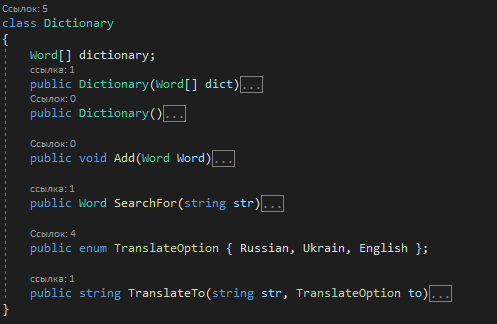


Рис.3.2 Реализация коллекции Dictionary

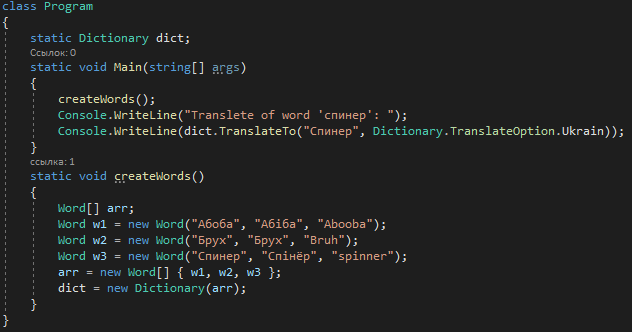


Рис.3.3 Реализация кода программы



Рис.3.4 Результат работы программы