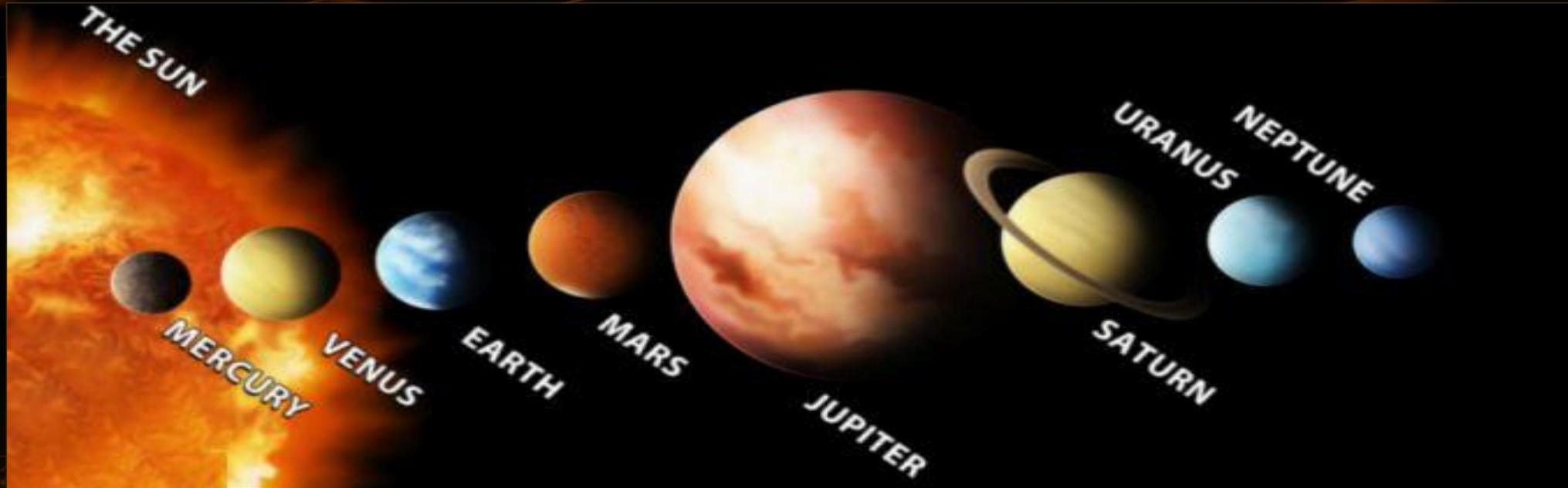


1. Перечисления
2. Структуры
3. Множества



[Description("Adds...")]





# Перечисления Объявление и использование перечислений

# Перечисления в C#

- Перечисление – это тип, который содержит значения из фиксированного именованных констант, объявляемый ключевым словом `enum`

```
public enum DayOfWeek
{
    Mon, Tue, Wed, Thu, Fri, Sat, Sun
}

class EnumExample
{
    static void Main()
    {
        DayOfWeek day = DayOfWeek.Wed;
        Console.WriteLine(day); // Wed
    }
}
```

# Пример перечисления

```
public enum CoffeeSize
{
    Small = 100, Normal = 150, Double = 300
}

public class Coffee
{
    public CoffeeSize size;

    public Coffee(CoffeeSize size)
    {
        this.size = size;
    }

    public CoffeeSize Size
    {
        get { return size; }
    }
}

продолжение примера
```

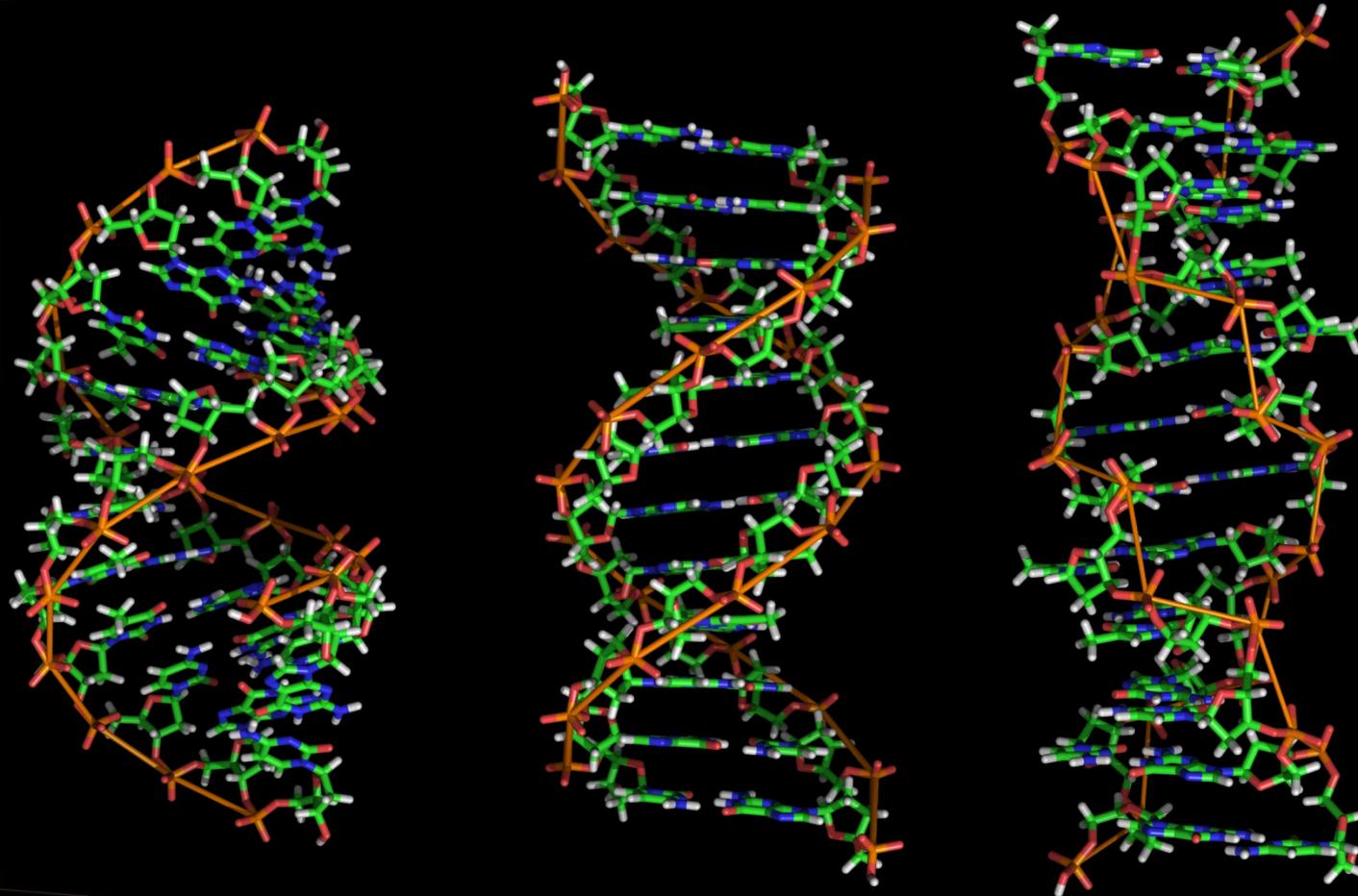
# Пример перечисления (2)

```
public class CoffeeMachine
{
    static void Main()
    {
        Coffee normalCoffee = new Coffee(CoffeeSize.Normal);
        Coffee doubleCoffee = new Coffee(CoffeeSize.Double);

        Console.WriteLine("The {0} coffee is {1} ml.",
            normalCoffee.Size, (int)normalCoffee.Size);
        // The Normal coffee is 150 ml.

        Console.WriteLine("The {0} coffee is {1} ml.",
            doubleCoffee.Size, (int)doubleCoffee.Size);
        // The Double coffee is 300 ml.
    }
}
```

# Структуры C#



# Структуры C#

- Что представляют структуры в C#?
  - Это тип данных по значению (такой как примитивные типы)
    - Примеры структур: **int**, **double**, **DateTime**
    - Классы это ссылочные типы
  - Объявляется ключевым словом **struct**
  - **Структуры** имеют поля, свойства, конструкторы, метода и т.д. (так же как и классы)
  - Всегда есть конструктор без параметров
    - Он не может быть удален
  - В основном используется для хранения данных

# Пример структуры C#

```
struct Point
{
    public int X { get; set; }
    public int Y { get; set; }
}

struct Color
{
    public byte RedValue { get; set; }
    public byte GreenValue { get; set; }
    public byte BlueValue { get; set; }
}

enum Edges { Straight, Rounded }
```

*(example continues)*

# Пример структуры C# (2)

```
struct Square
{
    public Point Location { get; set; }
    public int Size { get; set; }
    public Color SurfaceColor { get; set; }
    public Color BorderColor { get; set; }
    public Edges Edges { get; set; }

    public Square(Point location, int size,
        Color surfaceColor, Color borderColor,
        Edges edges) : this()
    {
        this.Location = location;
        this.Size = size;
        this.SurfaceColor = surfaceColor;
        this.BorderColor = borderColor;
        this.Edges = edges;
    }
}
```



# Множества

Коллекция, содержащая только отличающиеся элементы, называется *множеством (set)*.

# Что такое множества?

- В составе .NET4 имеются два множества - `HashSet<T>` и `SortedSet<T>`.
  - Класс `HashSet<T>` содержит неупорядоченный список различающихся элементов
  - в `SortedSet<T>` элементы упорядочены
- Оба они реализуют интерфейс `ISet<T>`
- Интерфейс `ISet<T>` предоставляет методы для создания объединения нескольких множеств, пересечения множеств и определения, является ли одно множество надмножеством или подмножеством другого.

# Пример множества

```
static void Main() {  
    SortedSet<char> ss = new SortedSet<char>();  
    SortedSet<char> ss1 = new SortedSet<char>();  
  
    ss.Add('A'); ss.Add('B'); ss.Add('C'); ss.Add('Z');  
    ShowColl(ss, "Первая коллекция: ");  
  
    ss1.Add('X'); ss1.Add('Y'); ss1.Add('Z');  
    ShowColl(ss1, "Вторая коллекция");  
  
    ss.SymmetricExceptWith(ss1);  
    ShowColl(ss, "Исключили разноименность (одинаковые элементы) двух  
множеств: ");  
    ss.UnionWith(ss1);  
    ShowColl(ss, "Объединение множеств: "); }  
}
```

# Пример множества (2)

```
ss.ExceptWith(ss1);  
ShowColl(ss, "Вычитание множеств");  
Console.ReadLine();  
}  
  
static void ShowColl(SortedSet<char> ss, string s) {  
    Console.WriteLine(s);  
    foreach (char ch in ss)  
        Console.Write(ch + " ");  
    Console.WriteLine("\n");  
}
```

# Результат выполнения

```
file:///C:/projects/test/ConsoleApplication1/ConsoleApplication1/bin/Debug/ConsoleApplication1...
Первая коллекция :
A B C Z
Вторая коллекция
X Y Z
Исключили разноименность (одинаковые элементы) двух множеств :
A B C X Y
Объединение множеств :
A B C X Y Z
Вычитание множеств
A B C
```