Автоматически реализуемое свойство

Синтаксис объявления

тип имя { get; set; }

Пример использования

public int UserCount { get; set; }

Общая форма одномерного индексатора

```
тип_элемента this[int индекс]
{
    // Аксессор для получения данных
     get
    {
        // Возврат значения, которое определяет индекс.
    }

    // Аксессор для установки данных
     set
     {
         // Установка значения, которое определяет индекс.
    }
}
```

Пример применения индексатора

```
public class AClass1
    {
        int[] imyArray = new int[20];

        public int this[int ind1]
        {
            get
            { return imyArray[ind1]; }
            set
            { imyArray[ind1] = value; }
        }
    }
```

```
class Program
```

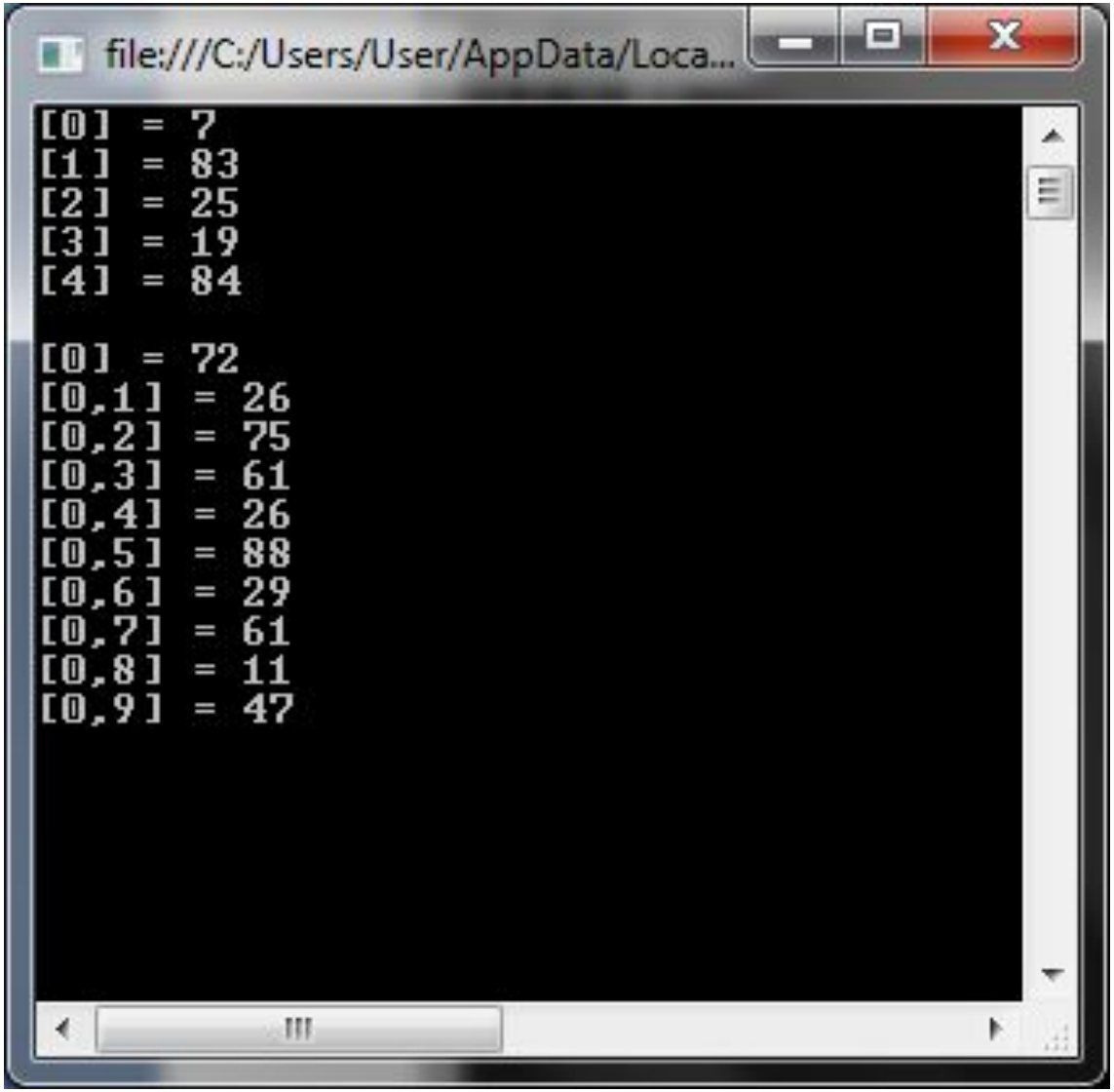
Применение двумерных индексаторов

```
public class AClass1
      {
         int[] imyArray = new int[20];
         int[,] imyArray1 = new int[20,10];

         public int this[int ind1]
         {
            get
            { return imyArray[ind1]; }
            set
            { imyArray[ind1] = value; }
         }

         public int this[int ind1, int ind2]
         {
            get
            { return imyArray1[ind1, ind2]; }
            set
            { imyArray1[ind1, ind2] = value; }
         } }
```

Перегрузка индексаторов

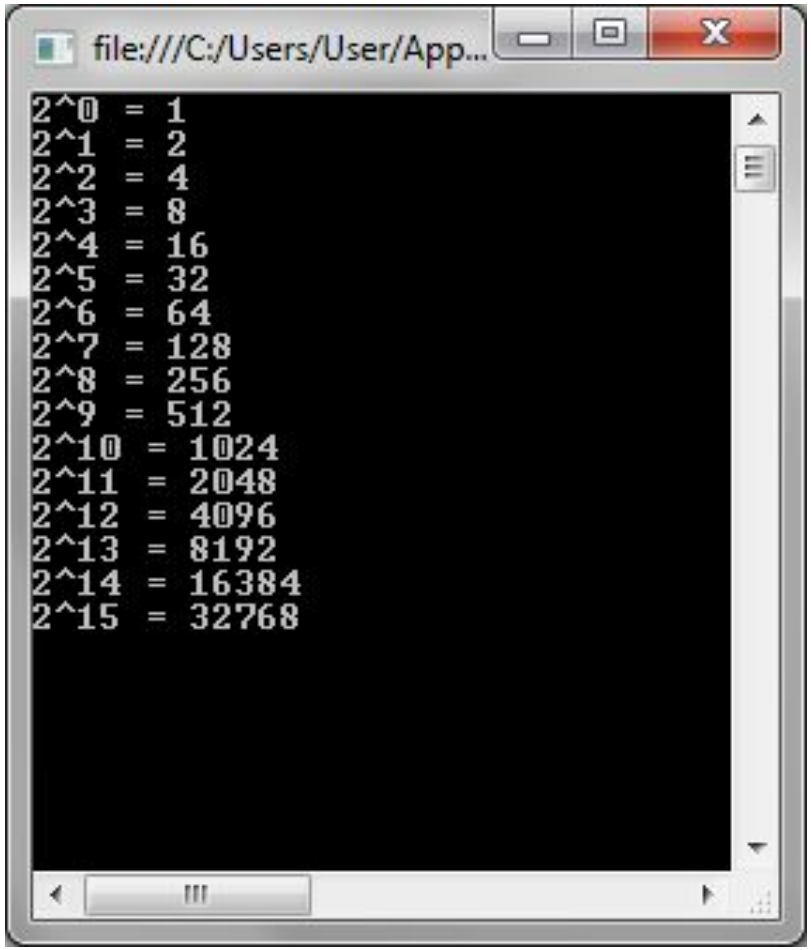
```
public class AClass1    {
        int[] imyArray = new int[5];
        int[] imyArray1 = new int[10];

        public int this[int ind1]
        {   get   { return imyArray[ind1]; }
            set   { imyArray[ind1] = value; }
        }
        public int this[double ind2]
        {get  {
            int ind;
            if ((ind2 - (int)ind2) < 0.5)
                    ind = (int)ind2;
            else ind = (int)ind2 + 1;
            return imyArray1[ind]; }
            set  {
                int ind;
                if ((ind2 - (int)ind2) < 0.5)
                      ind = (int)ind2;
                else ind = (int)ind2 + 1;
                imyArray1[ind] = value; }
        } }
```

Результат выполнения программы



Индексаторы без базового массива

```
public class AClass1
    {
      public double this[int ind1]
    {
       get
       {
          return Math.Pow(2.0, ind1);
       }
    }
  }
  class Program
  {
    static void Main(string[] args)
    {
       AClass1 Ac1 = new AClass1();
       for (int i = 0; i <= 15; i++)
          Console.Write("2^{0} = {1}\n", i, Ac1[i]);
       Console.ReadLine();
    }  }
```



Применение модификаторов доступа в аксессорах

```
class PropAccess
   {
      int prop;

      public PropAccess() { prop = 0; }

      public int MyProp
      {
         get
         {     return prop;      }
         private set
         {     prop = value;      }
      }
      public void IncrProp()
      {
         MyProp++;
      }
   }

   class Program
```

Применение модификаторов доступа в автоматически реализуемых свойствах

```
public int Length { get; private set; }
```