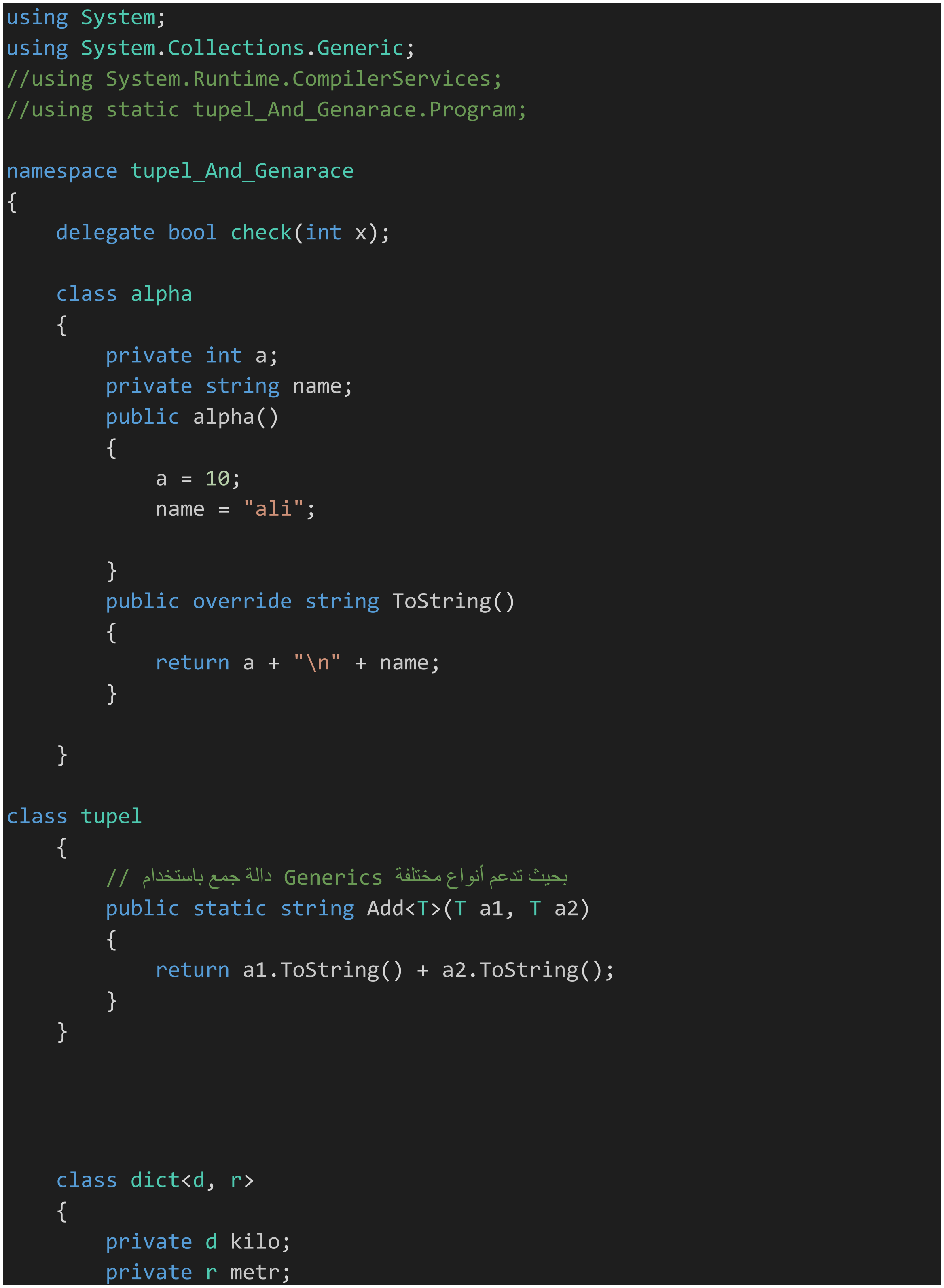
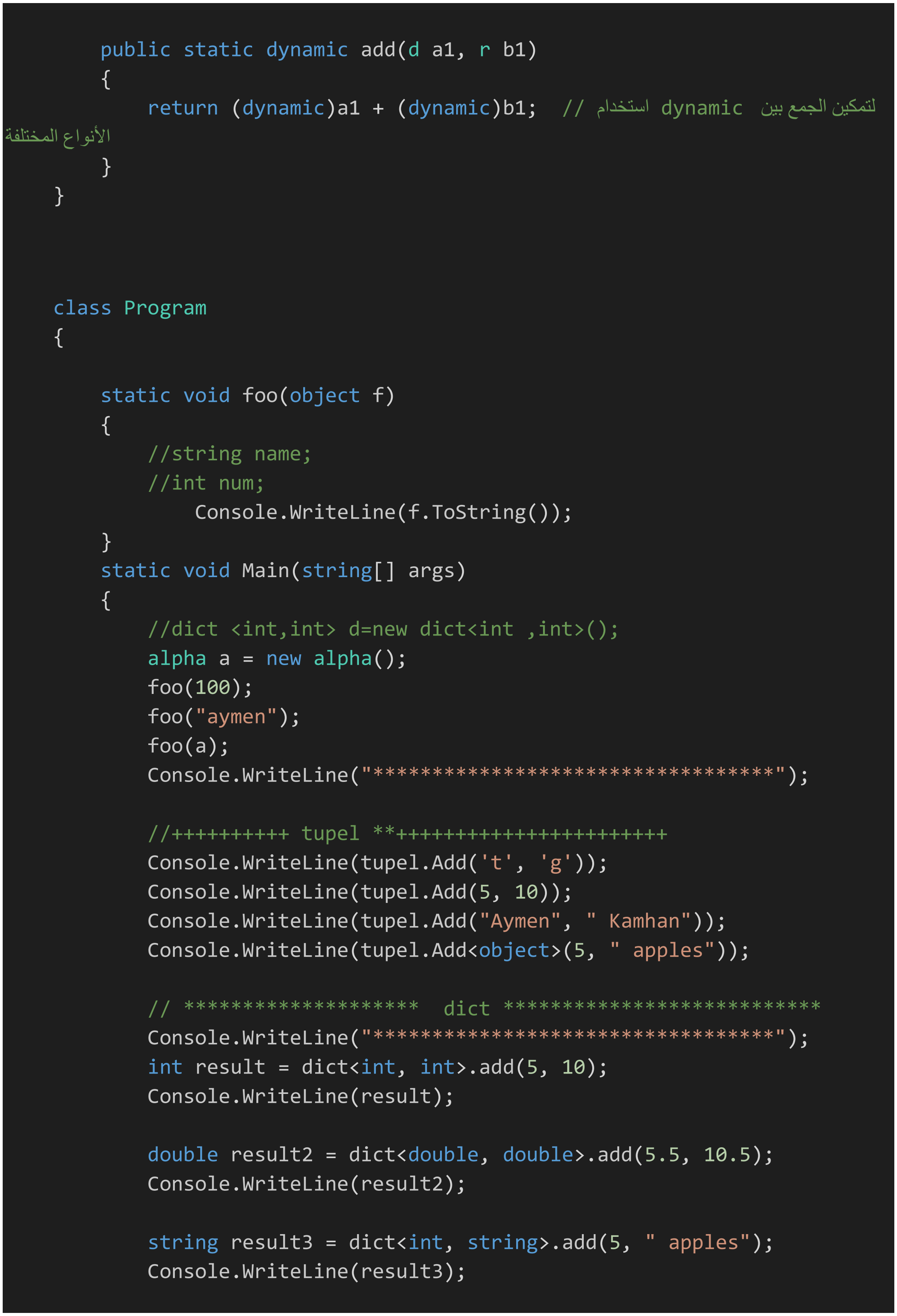
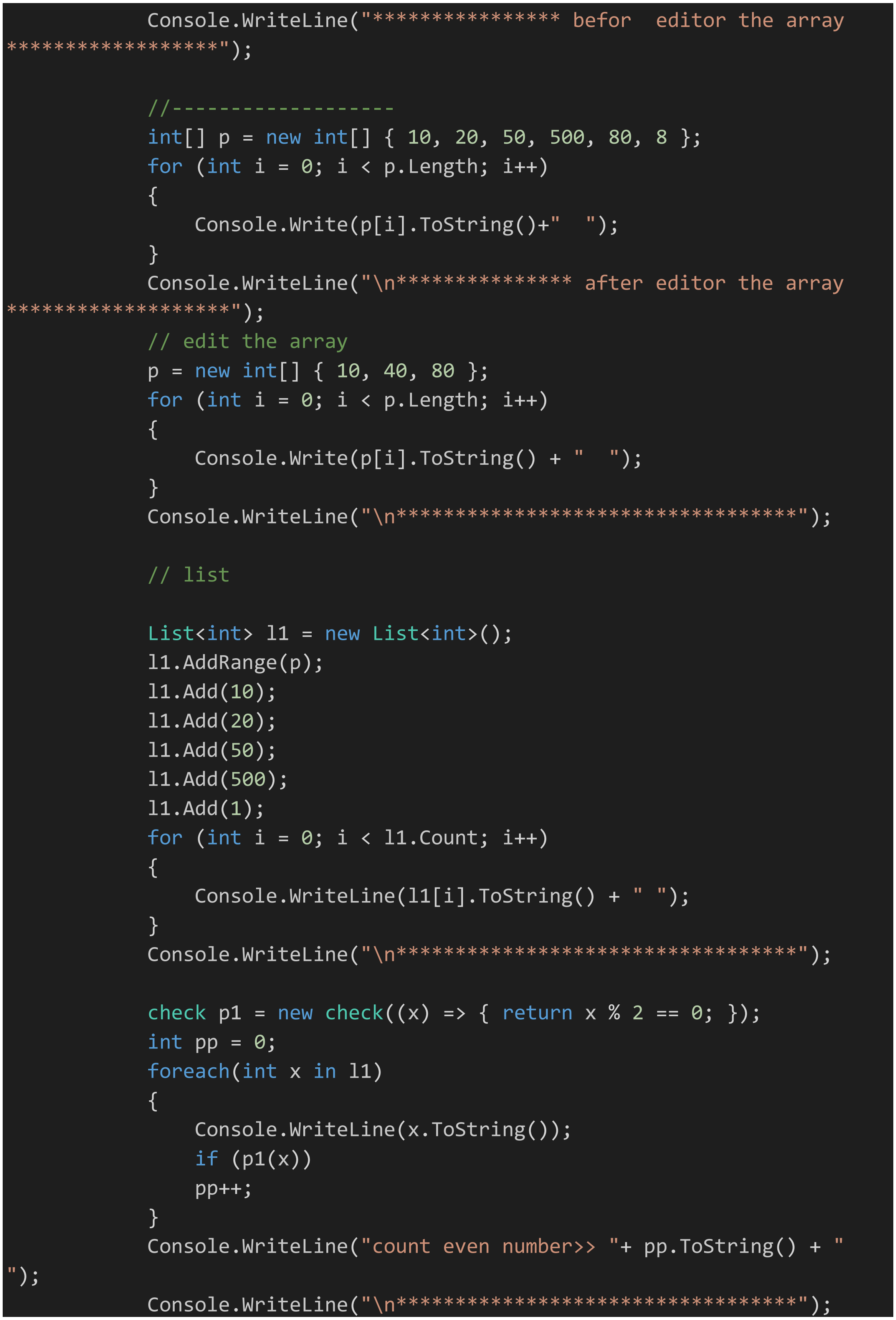
كود مهم في Generics و Tupel و Lise و Delegate
برمجه متقدمة



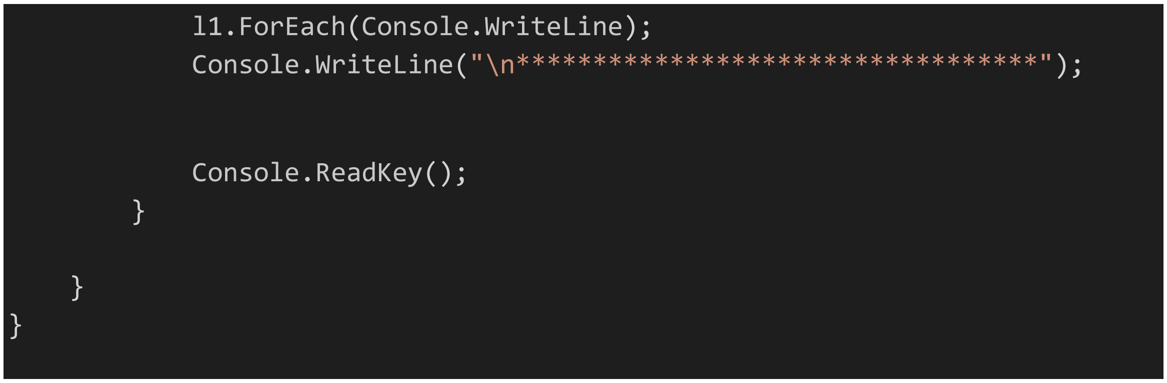
```
using System;
using System.Collections.Generic;
//using System.Runtime.CompilerServices;
//using static tupel_And_Genarace.Program;

namespace tupel_And_Genarace
{
    delegate bool check(int x);

    class alpha
    {
        private int a;
        private string name;
        public alpha()
        {
            a = 10;
            name = "ali";

        }
        public override string ToString()
        {
            return a + "\n" + name;
        }

    }

class tupel
    {
        // دالة جمع باستخدام Generics بحيث تدعم أنواع مختلفة
        public static string Add<T>(T a1, T a2)
        {
            return a1.ToString() + a2.ToString();
        }
    }



    class dict<d, r>
    {
        private d kilo;
        private r metr;
```



```
        public static dynamic add(d a1, r b1)
        {
            return (dynamic)a1 + (dynamic)b1;  // استخدام dynamic  لتمكين الجمع بين
الأنواع المختلفة
        }
    }



    class Program
    {

        static void foo(object f)
        {
            //string name;
            //int num;
                Console.WriteLine(f.ToString());
        }
        static void Main(string[] args)
        {
            //dict <int,int> d=new dict<int ,int>();
            alpha a = new alpha();
            foo(100);
            foo("aymen");
            foo(a);
            Console.WriteLine("***********************************");

            //++++++++++ tupel **+++++++++++++++++++++++
            Console.WriteLine(tupel.Add('t', 'g'));
            Console.WriteLine(tupel.Add(5, 10));
            Console.WriteLine(tupel.Add("Aymen", " Kamhan"));
            Console.WriteLine(tupel.Add<object>(5, " apples"));

            // ********************  dict ***************************
            Console.WriteLine("***********************************");
            int result = dict<int, int>.add(5, 10);
            Console.WriteLine(result);

            double result2 = dict<double, double>.add(5.5, 10.5);
            Console.WriteLine(result2);

            string result3 = dict<int, string>.add(5, " apples");
            Console.WriteLine(result3);
```



```
            Console.WriteLine("**************** befor  editor the array 
******************");

            //--------------------
            int[] p = new int[] { 10, 20, 50, 500, 80, 8 };
            for (int i = 0; i < p.Length; i++)
            {
                Console.Write(p[i].ToString()+"  ");
            }
            Console.WriteLine("\n*************** after editor the array 
*******************");
            // edit the array
            p = new int[] { 10, 40, 80 };
            for (int i = 0; i < p.Length; i++)
            {
                Console.Write(p[i].ToString() + "  ");
            }
            Console.WriteLine("\n***********************************");

            // list

            List<int> l1 = new List<int>();
            l1.AddRange(p);
            l1.Add(10);
            l1.Add(20);
            l1.Add(50);
            l1.Add(500);
            l1.Add(1);
            for (int i = 0; i < l1.Count; i++)
            {
                Console.WriteLine(l1[i].ToString() + " ");
            }
            Console.WriteLine("\n***********************************");

            check p1 = new check((x) => { return x % 2 == 0; });
            int pp = 0;
            foreach(int x in l1)
            {
                Console.WriteLine(x.ToString());
                if (p1(x))
                pp++;
            }
            Console.WriteLine("count even number>> "+ pp.ToString() + " 
");

            Console.WriteLine("\n***********************************");
```



```
            l1.ForEach(Console.WriteLine);
            Console.WriteLine("\n**********************************");


            Console.ReadKey();
        }

    }
}
```