Dokumentation af projektet 'TriangleWiz' for Danske Bank.

v. Tomas Medici

Data er hardcoded i programmet i Program.cs:

lection

```
using System;
using System.Collections.Generic;

namespace TriangleWiz
{
    class Program
    {
        static void Main(string[] args)
        {
            TriangleWizard wiz = new TriangleWizard();

            List<List<int>> triangleList = new List<List<int>>();

            triangleList.Add(new List<int> { 215 });
            triangleList.Add(new List<int> { 192, 124 });
            triangleList.Add(new List<int> { 117, 269, 442 });
            triangleList.Add(new List<int> { 218, 836, 347, 235 });
            triangleList.Add(new List<int> { 320, 805, 522, 417, 345 });
            triangleList.Add(new List<int> { 229, 601, 728, 835, 133, 124 });
            triangleList.Add(new List<int> { 248, 202, 277, 433, 207, 263, 257 });
            triangleList.Add(new List<int> { 359, 464, 504, 528, 516, 716, 871, 182 });
            triangleList.Add(new List<int> { 461, 441, 426, 656, 863, 560, 380, 171, 923 });
            triangleList.Add(new List<int> { 381, 348, 573, 533, 448, 632, 387, 176, 975, 449 });
            triangleList.Add(new List<int> { 223, 711, 445, 645, 245, 543, 931, 532, 937, 541, 444 });
            triangleList.Add(new List<int> { 330, 131, 333, 928, 376, 733, 017, 778, 839, 168, 197, 197 });
            triangleList.Add(new List<int> { 131, 171, 522, 137, 217, 224, 291, 413, 528, 520, 227, 229, 928 });
            triangleList.Add(new List<int> { 223, 626, 034, 683, 839, 052, 627, 310, 713, 999, 629, 817, 410, 121 });
            triangleList.Add(new List<int> { 924, 622, 911, 233, 325, 139, 721, 218, 253, 223, 107, 233, 230, 124, 233 });

            (int, List<int>) result = wiz.CalculatePath(triangleList);
            Console.WriteLine("Output: ");
            Console.WriteLine("Max sum: {0} ", result.Item1);
            Console.WriteLine("Path: " + String.Join(", ", result.Item2));
        }
    }
}

```

Der instantieres et object af klassen TriangleWizard, hvor selve beregninger foregår i. Metoden CalculatePath kalder så en række andre metoder i klassen:

```
using System;
using System.Collections.Generic;
using System.Linq;

namespace TriangleWizard_2
{
    public class TriangleWiz
    {
        public TriangleWiz()
        {
            container1 = new List<List<int>>();                         //Containers for general use, due to the alternating approach
            container2 = new List<List<int>>();
        }

        public (int, List<int>) CalculatePath(List<List<int>> bruttolist)
        {
            int size = bruttolist.Count;
            AllPaths(bruttolist, size);                                 //Calculates indexes of all possible combinations of paths down through the triangle
            AllLists(bruttolist, size);                                 //Translates these lists of indexes to actual lists of numbers from the triangle
            ValidLists(bruttolist, size);                               //Selects the lists of numbers that obey the alternating odd/even rule
            (int count, int sum) = HighestSum(size);                    //Calculates the highest sum, and returns the corresponding list of numbers' index in container

            if (size % 2 != 0)                                          //If length of bruttolist ('height' of triangle) is odd, use data from container1
            {
                return (sum, container1[count]);                        //Returns the sum and the list containing the numbers in the path
            }
            else                                                        //Else use data from container2
            {
                return (sum, container2[count]);
            }
        }
```

Resultatet kørt på udleverede data bliver så: