**Game – Exam 08.07.2018 – Solution**

**Program.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Game

{

class Program

{

static void Main(string[] args)

{

int capacity = int.Parse(Console.ReadLine());

Dictionary<String, CapacityList> players = new Dictionary<string, CapacityList>();

string command = "";

do

{

command = Console.ReadLine();

string[] commands = command.Split(' ').ToArray();

switch (commands[0])

{

case "Dice":

string player = commands[1];

int value1 = int.Parse(commands[2]);

int value2 = int.Parse(commands[3]);

if (players.ContainsKey(player))

{

players[player].Add(new Pair(value1, value2));

}

else

{

players.Add(player, new CapacityList(capacity));

players[player].Add(new Pair(value1, value2));

}

break;

case "CurrentPairSum":

string playerName = commands[1];

if (players.ContainsKey(playerName))

{

Console.WriteLine(players[playerName].Sum());

}

break;

case "CurrentStanding":

players = players.OrderBy(element => element.Value.Sum().Difference())

.ToDictionary(element => element.Key, element => element.Value);

foreach (var item in players)

{

Console.WriteLine(item.Key + " - " + item.Value.Sum());

}

break;

case "CurrentState":

string playerNameState = commands[1];

if (players.ContainsKey(playerNameState))

{

players[playerNameState].PrintCurrentState();

}

break;

case "Winner":

players = players.OrderBy(element => element.Value.Sum().Difference())

.ThenBy(element => element.Key)

.ToDictionary(element => element.Key, element => element.Value);

Console.WriteLine("{0} wins the game!", players.First().Key);

break;

}

} while (command != "End");

}

}

}

**Pair.cs**

using System;

namespace Game

{

class Pair

{

public int First { get; set; }

public int Last { get; set; }

public Pair(int first, int secondLast)

{

First = first;

Last = secondLast;

}

public override string ToString()

{

return $"({First}, {Last})";

}

public int Difference()

{

return Math.Abs(this.First - this.Last);

}

}

}

**CapacityList.cs**

using System;

namespace Game

{

class CapacityList

{

private const int InitialCapacity = 2;

private Pair[] items;

private int startIndex = 0;

private int nextIndex = 0;

public CapacityList(int capacity = InitialCapacity)

{

this.items = new Pair[capacity];

}

public int Count { get; private set; }

public Pair SumIntervalPairs()

{ //TODO: сумирайте двойките от startIndex до nextIndex

Pair result = new Pair(0, 0);

for (int i = startIndex; i < nextIndex; i++)

{

result.First = result.First + items[i].First;

result.Last = result.Last + items[i].Last;

}

return result;

}

public Pair Sum()

{ //TODO: сумирайте двойките от 0 до this.Count – всички двойки, които имат право да участват в класирането

Pair result = new Pair(0, 0);

for (int i = 0; i < this.Count; i++)

{

result.First = result.First + this.items[i].First;

result.Last = result.Last + this.items[i].Last;

}

return result;

}

public void Add(Pair item)

{ //TODO: Добавяне на двойката

if (nextIndex == items.Length)

{

items[startIndex] = SumIntervalPairs();

startIndex++;

nextIndex = startIndex;

Count++;

}

items[nextIndex] = item;

nextIndex++;

}

public void PrintCurrentState()

{ //TODO: отпечатайте всички двойки от 0 до nextIndex

for(int i = 0; i < nextIndex; i++)

{

Console.WriteLine(items[i].ToString());

}

}

}

}