string binary = string.Empty; - Празен стринг

string str = Console.ReadLine();

string[] date = str.Split(‘:’,’ ’);

int hour = int.Parse(date[0]); - Така достъпвам всеки един от елементите на стринга

char[] letters = ”Pesho”.ToCharArray(); -Така си вземам всяка буква от името

string str = “Pesho”;

foreach(var letter in str){Console.WriteLine(letter)} – Така ще ми изпише всяка буква на н-ред

Console.OutputEncoding = System.Text.Encoding.UTF8 – Така печатам на кирилица

Int[] numbers = new int[n]; - Така си създавам нов масив с n (нулеви) елемента

for (int i=0; i<numbers.Length; i++)

numbers[i] = i+1; - Така си вкарвам n стойности в масива

numbers[3] = 20;

numbers[5] = numbers[2] + numbers[7]; - Така променям стойноста на елемент от масива

for (int i = 0; i < numbers.Length; i++)

Console.WriteLine("numbers[{0}] = {1}", i, numbers[i]); - Така си отпечатвам елементите на масива с техните стойности

Console.WriteLine("Sum = " + numbers.Sum());

Console.WriteLine("Min = " + numbers.Min());

Console.WriteLine("Max = " + numbers.Max());

Console.WriteLine("First = " + numbers.First());

Console.WriteLine("Last = " + numbers.Last());

За да ползвам тези готови функции трябва да си активирам using System.Linq;

Array.Sort(number); - За да ползвам функции трябва да си активирам using System.Linq;

For (int I =0; i< number.Length; i++){ Console.WriteLine(“number[{0}] = {1}”, I, number[i]);

Тази функция ми подрежда числата в масива по големина или букви(име) по азбучен ред

string[] names = { "Peter", "Maria", "Katya", "Todor" }; - Така се въвежда масив от имена

names.Reverse(); - Така се обръща списъка с имена

names[0] = names[0] + " (junior)";

foreach (var name in names)

{

Console.WriteLine(name); - Така прочитам(отпечатвам) всички имена от масива

}

names[4] = "Nakov"; // This will cause an exception!

int width = 4, height = 6;

string[,] matrix =

new string[height, width];

for (int row = 0; row < height; row++)

{

for (int col = 0; col < width; col++)

{

matrix[row, col] = "" + - Така достъпвам елементи от матрицата съответно на ред и колона

(char)('a' + row) +

(char)('a' + col);

}

}

Така си създавам матрица с 4 колони и 6 реда

for (int row = 0; row < height; row++){ for (int col = 0; col < width; col++)

{ Console.Write(matrix[row, col] + " ") } – Така отпечатвам матрицата

List<int> numbers = new List<int>(); - Така си създавам нов празен лист от числа

List<string> names = new List<string>(); - Така си създавам нов празен лист от стрингове

numbers.Add(5); - Така си добавям число към листа numbers

Console.WriteLine(numbers[0]); // 5

List<List<int>> - Така създавам двумерен лист от числа (int)

За да разбереш как работят листовете и какви операции мога да извършвам с тях погледни файла с проджектите от лекцията за напреднали.

Мога да добавям и сортирам числа.

Мога да добавям, премахвам, вкарвам и променям стрингове.

List<string> names =

new List<string>() { "Peter", "Maria", "Katya", "Todor" };

names.Add("Nakov"); // Peter, Maria, Katya, Todor, Nakov

names.RemoveAt(0); // Maria, Katya, Todor, Nakov

names.Insert(3, "Sylvia"); // Maria, Katya, Todor, Sylvia, Nakov

names[1] = "Michael"; // Maria, Michael, Todor, Sylvia, Nakov

foreach (var name in names)

{Console.WriteLine(name);}

Операции със стрингове:

string str = "SoftUni";

Console.WriteLine(str);

for (int i = 0; i < str.Length; i++)

{

Console.WriteLine("str[{0}] = {1}", i, str[i]);

}

Console.WriteLine(str.IndexOf("Uni")); // 4

Console.WriteLine(str.IndexOf("uni")); // -1 (not found)

Console.WriteLine(str.Substring(4, 3)); // Uni – Реже стринга на по-малки стрингове. 4 и 3 показват от коя позиция да започне и колко елемента да отреже.

Console.WriteLine(str.Replace("Soft", "Hard")); // HardUni

Console.WriteLine(str.ToLower()); // softuni

Console.WriteLine(str.ToUpper()); // SOFTUNI

string allLangs = "C#, Java; HTML, CSS; PHP, SQL";

string[] langs = allLangs.Split(new char[] {',', ';', ' '}, - Разделям вътрешните стрингчета по запетая, точка и запетая и т.н. и ги правя елементи от един нов масив

StringSplitOptions.RemoveEmptyEntries); -

foreach (var lang in langs)

Console.WriteLine(lang);

Console.WriteLine("Langs = " + string.Join(", ", langs)); - Тази операция връща отделните стрингчета от масива отново в един общ стринг

Console.WriteLine(" \n\n Software University ".Trim()); - Тази операция изрязва всички празни места в началото и края но не и по средата

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

string str = Convert.ToString(num, 2).PadLeft(64, '0');

char[] array = str.ToCharArray(); Първо инта го превръщам в стринга и после в масив от чар.

Console.WriteLine(Convert.ToUInt64(new string(array),2)) – и после го връщам в инт

int[] numbers = new int[4];

for (int i = 0; i < 4; i++)

numbers[i] = int.Parse(Console.ReadLine()); - Създавам си нов масив от 4ри елемента и във фор цикъл го пълня.

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

int[] numbers = new int[count];

int countBits = 0;

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

numbers[i] = int.Parse(Console.ReadLine());

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

int maxNumber = numbers.Max(); - Така си вземам най-голямото число.Трябва да активирам System.Linq!!! А преди това си иниц. Масив и го пълня.

int maxHelperNumber = int.MinValue;

if (number > maxHelperNumber)

maxHelperNumber = number; - Това е другия начин за намиране на най-голямо число.

int position = x + 1;

if (y==1)

position = x + 4;

if (y==2)

position = x + 7;

firstValue += position - 1; - Това е матрица от 3x3 с firstValue =10.

Math.Abs(leftover) – Тази функция връща винаги положително число.Мога да я използвам при изваждания и резултата ще е винаги положително число.

Math.Min((allLegs / 4), tableTops); - Така намирам минималното от две числа.

DateTime time = DateTime.Parse(Console.ReadLine());

if (time.Hour >= 14 && time.Hour <= 18) – Тук използвам DateTime класа и мога да взема от него часа, мин и датите и да боравя с тях.

private static void CheckBoxes( int firstWidth, int firstHeight, int firstDepth,int secondWidth, int secondHeight, int secondDepth)

if (firstWidth < secondWidth && firstHeight < secondHeight && firstDepth < secondDepth) Console.WriteLine("({0}, {1}, {2}) < ({3}, {4}, {5})",firstWidth, firstHeight, firstDepth, secondWidth, secondHeight, secondDepth); - Така си правя готов метод за намиране на размерите на две кутии и дали едната се събира в другата.

CheckBoxes(w1, h1, d1, w2, h2, d2);

CheckBoxes(w1, h1, d1, w2, d2, h2);

CheckBoxes(w1, h1, d1, h2, w2, d2);

CheckBoxes(w1, h1, d1, h2, d2, w2);

CheckBoxes(w1, h1, d1, d2, w2, h2);

CheckBoxes(w1, h1, d1, d2, h2, w2);

// Try to fit the second box inside the first box (6 possibilities)

CheckBoxes(w2, h2, d2, w1, h1, d1);

CheckBoxes(w2, h2, d2, w1, d1, h1);

CheckBoxes(w2, h2, d2, h1, w1, d1);

CheckBoxes(w2, h2, d2, h1, d1, w1);

CheckBoxes(w2, h2, d2, d1, w1, h1);

CheckBoxes(w2, h2, d2, d1, h1, w1);

След това се връщаме в главният метод и ги проверяваме всички комбинации.

double ab = Math.Sqrt(Math.Pow((bX - aX),2) + Math.Pow((bY - aY),2)); - Така се намира права ab.

DateTime startTime = new DateTime(2014, 12, 05, startHour, startMinutes, 0);

DateTime endTime = startTime.AddHours(durationHours).AddMinutes(durationMinutes);

Console.WriteLine("{0:hh:mm:tt}", endTime); - Така се взема часа и минутите и след това се добавят часове и минути. Също така се форматира output.