Колледж электроники и приборостроения

Практическая работа №5

Выполнил: Коломиец Максим

Группа: 23 ИП

using System;

class Hallo

{

static int Sum(int[] array)

{

int count = 0;

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

{

if (array[i] % 7 == 0)

{

count += array[i];

}

}

return count;

}

static int TakeSum(int[] array, int index1, int index2){

int sum = 0;

for (int i = index1; i <= index2 ; i++)

{

sum += array[i];

}

return sum;

}

static void Main()

{

//---------------------------1 кратны 7

// Random random = new Random();

// int[] array;

// int n = random.Next(10, 100);

// int sumInt;

// array = new int[n];

// for (int i = 0; i < n; i++)

// {

// array[i] = random.Next(2140000000);

// }

// sumInt = Sum(array);

// Console.WriteLine("Сумма чисел массива кратных 7: {0}", sumInt);

//-----------------------------2 12 номер

Console.Write("Введите количество элементов в массиве: ");

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

int[] array = new int[n];

int maxIndex = 0;

int minIndex = 0;

int sumArray = 0;

Console.WriteLine("Введите элементы массива:");

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

{

Console.Write($"Элемент {i + 1}: ");

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

}

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

{

if (array[i] > array[maxIndex]) {

maxIndex = i;

} else if (array[i] < array[minIndex]){

minIndex = i;

}

}

if (maxIndex < minIndex || maxIndex == minIndex) {

Console.WriteLine("Элементы равны или максимальное значение раньше минимального");

} else{

sumArray = TakeSum(array, minIndex, maxIndex);

}

Console.WriteLine("Сумма между минмальным и максимальным ровна: {0}", sumArray);

}

}

Контрольные вопросы

1.Одномерный массив — структура данных для хранения набора элементов одного типа, доступных по индексу.

2. Описание одномерного массива:

- Указывается тип элементов, квадратные скобки `[]`, имя массива и размер.

3. Операция обращения по индексу — доступ к элементу массива по его порядковому номеру (индексу).

4. Предпочтительный оператор цикла для обработки одномерных массивов — `for`, так как он позволяет явно управлять индексом и удобен для последовательного доступа к элементам.

**Ответы на вопросы контрольные вопросы:**

1. Алгоритм с ветвлением — это алгоритм, в котором порядок выполнения команд зависит от выполнения некоторого условия.

2. Условия в операторе ветвления — это логические выражения, которые могут быть истинными (`true`) или ложными (`false`).

3. Знаки отношений для составления условий:

- `==` (равно)

- `!=` (не равно)

- `<` (меньше)

- `>` (больше)

- `<=` (меньше или равно)

- `>=` (больше или равно)

4. Составное условие — это условие, состоящее из нескольких простых условий, объединенных логическими операторами (`&&` — логическое И, `||` — логическое ИЛИ, `!` — логическое НЕ).

5. Правила записи составных условий:

- Используйте круглые скобки для группировки условий.

- Операции логического И (`&&`) имеют приоритет над операциями логического ИЛИ (`||`).

- Операция логического НЕ (`!`) имеет наивысший приоритет.

6. Составной оператор используется в случаях, когда нужно выполнить несколько команд в блоке кода, например, внутри условного оператора (`if`, `else if`, `else`), цикла (`for`, `while`), или `switch`. Составной оператор заключается в фигурные скобки `{}`.