Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»**

**(ФГБОУ ВО «КубГТУ»)**

**Институт компьютерных систем и информационной безопасности**

**Кафедра информационных систем и программирования**

**ЛАБОРАТОРНАЯ РАБОТА №1**

**Дисциплина: тестирование и отладка программного обеспечения**

Работу выполнил: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Я.С Давков

Направление подготовки: 09.03.04 Программная инженерия

Преподаватель: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.Г. Волик

**1 Цель работы**

Цель работы – изучить инструментальные средства и возможности отладки программ в интегрированной среде Microsoft Visual Studio или JetBrains Rider.

**2 Краткая теория**

Интегрированные интерактивные среды разработки программ (IDE) включают в себя ряд средств, облегчающих процесс нахождения ошибок в программе, которые не позволяют ей корректно работать. Как Microsoft Visual Studio, так и JetBrains Rider имеют в своем составе схожий набор средств отладки.

**3 Задание**

1) Выполнить задание в соответствии с пунктом 5.1.

2) Выполнить задание в соответствии с пунктом 5.2.

3) Оформить отчёт

**4 Выполнение работы**

Выполним задание 5.1. Наберем программу из примера и выполним её по шагам, фиксируя в строке с каким номером находится строка выполнения при каждом нажатии на F10.

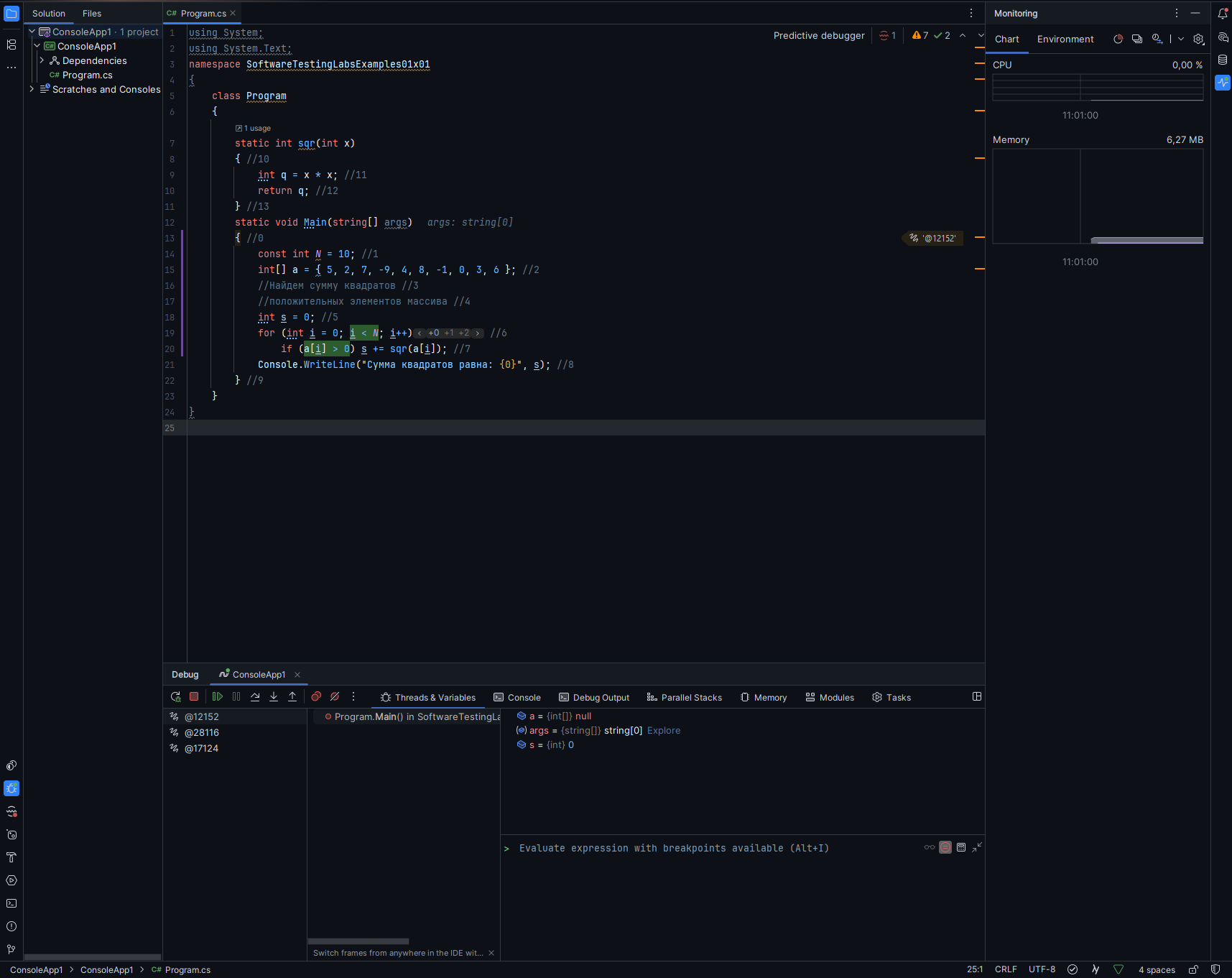


Рисунок 1 – Запущенная программа в режиме Debug

Порядок выполнения программы по строкам: 0, 1, 2, 5, 6, 7, 6, 7, 6, 7, 7, 6, 7, 6, 7, 6, 7, 6, 7, 6, 7, 6, 7, 6, 7, 6, 8, 9.

Наберем программу из 2 примера и выполним такие же действия.

Порядок выполнения программы по строкам: 0, 1, 2, 5, 6, 7, 7.1, 6, 7, 7.1, 6, 7, 7.1, 6, 7, 6, 7, 7.1, 6, 7, 7.1, 6, 7, 6, 7, 6, 7, 7.1, 6, 7, 7.1, 6, 8, 9.

Выполним трассировку исходной программы. Порядок выполнения по строкам – 0, 2, 5, 6, 7, 10, 11, 12, 13, 7, 6, 7, 10, 11, 12, 13, 7, 6, 7, 10, 11, 12, 13, 7, 6, 7, 6, 7, 10, 11, 12, 13, 7, 6, 7, 10, 11, 12, 13, 7, 6, 7, 6, 7, 6, 7, 10 ,11, 12, 13, 7, 6, 7, 10, 11, 12, 13, 7, 6, 8, 9.

Выполним задание 5.2 - 1) Набрать и откомпилировать следующую программу. 2) После каждой строки программы проставить номер. Например, //1, //2 и т.д. 3) Выполнить трассировку программы (без захода в функции стандартных библиотек), наблюдая за переменными в окне Locals. 4) Остановить отладку программы. 5) Установить точку остановки на операторе if (a[i] > z). 6) Выполнить программу до курсора на строке s = sum(a, N); 7) Продолжить выполнить программы до точки остановки. Далее продолжать пошаговое выполнение до строки Console.WriteLine(k). 8) Добавить в окно Watch 1 переменные x, y, s для наблюдения изменения их значений. Продолжать пошаговое выполнение. 9) Остановить отладку программы. 10) В операторе цикла while задать условную точку останова по числу проходов. Запустить программу для отладки. 11) Продолжать пошаговое выполнение до конца программы, наблюдая изменение значений x, y, s в окне Watch 1. 12) Записать полученные результаты. 13) Остановить отладку программы. 14) В операторе цикла while задать еще одну условную точку останова по логическому условию. Запустить программу для отладки 15) Продолжать пошаговое выполнение до конца программы, наблюдая изменение значений x, y, s в окне Watch 1. 16) Записать полученные результаты.

Листинг программы:

using System;  
using System.Text;  
  
namespace SoftwareTestingLabsExamples01x03  
{  
 class Program  
 {  
 //Метод, считающий сумму элементов массива  
 static int sum(int[] x, int N)  
 { // 1  
 int s = 0; // 2  
 for (int i = 0; i < N; i++) // 3  
 s += x[i]; // 4  
 return s; // 5  
 }  
  
 //Метод для ввода целых чисел с клавиатуры  
 static int ReadInt(string prompt)  
 { // 6  
 Console.Write(prompt); // 7  
 int x = int.Parse(Console.ReadLine()); // 8  
 return x; // 9  
 }  
  
 static void Main(string[] args)  
 { // 10  
 const int *N* = 10; // 11  
 int[] a = new int[N] { 1, 3, -5, 0, 4, 6, -1, 9, 3, 2 }; // 12  
 //Найдем максимальный элемент массива  
 int m = a[0]; // 13  
 for (int i = 1; i < *N*; i++) // 14  
 if (m < a[i]) // 15  
 m = a[i]; // 16  
 Console.WriteLine(m); // 17  
 //Найдем сумму элементов массива  
 int s; // 18  
 s = sum(a, *N*); // 19  
 Console.WriteLine(s); // 20  
 int z = s / m; // 21  
 int k = 0; // 22  
 for (int i = 0; i < *N*; i++) // 23  
 if (a[i] > z) // 24  
 k += a[i]; // 25  
 else // 26  
 k -= a[i]; // 27  
 Console.WriteLine(k); // 28  
 int x, y; // 29  
 x = ReadInt(""); //30  
 y = ReadInt(""); // 31  
 s = 0; // 32  
 while ((x != 0) && (x != 0)) // 33  
 { // 34  
 x--; //35  
 y--; // 36  
 s += x + y; // 37  
 } // 38  
  
 Console.WriteLine(s); // 39  
 } // 40  
 }  
}

В окне можно наблюдать переменные программы в текущем состоянии.

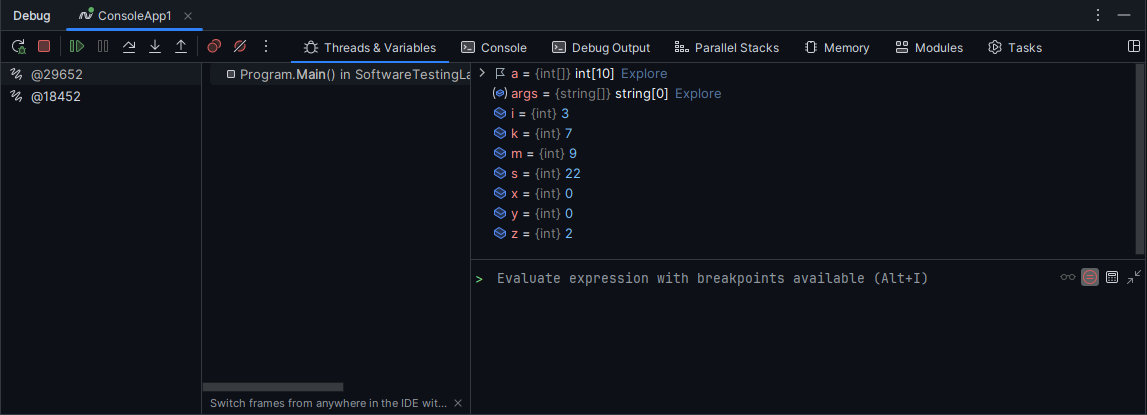


Рисунок 2 – текущие переменные

Установим точку остановки в while цикли посмотрим значения переменных.

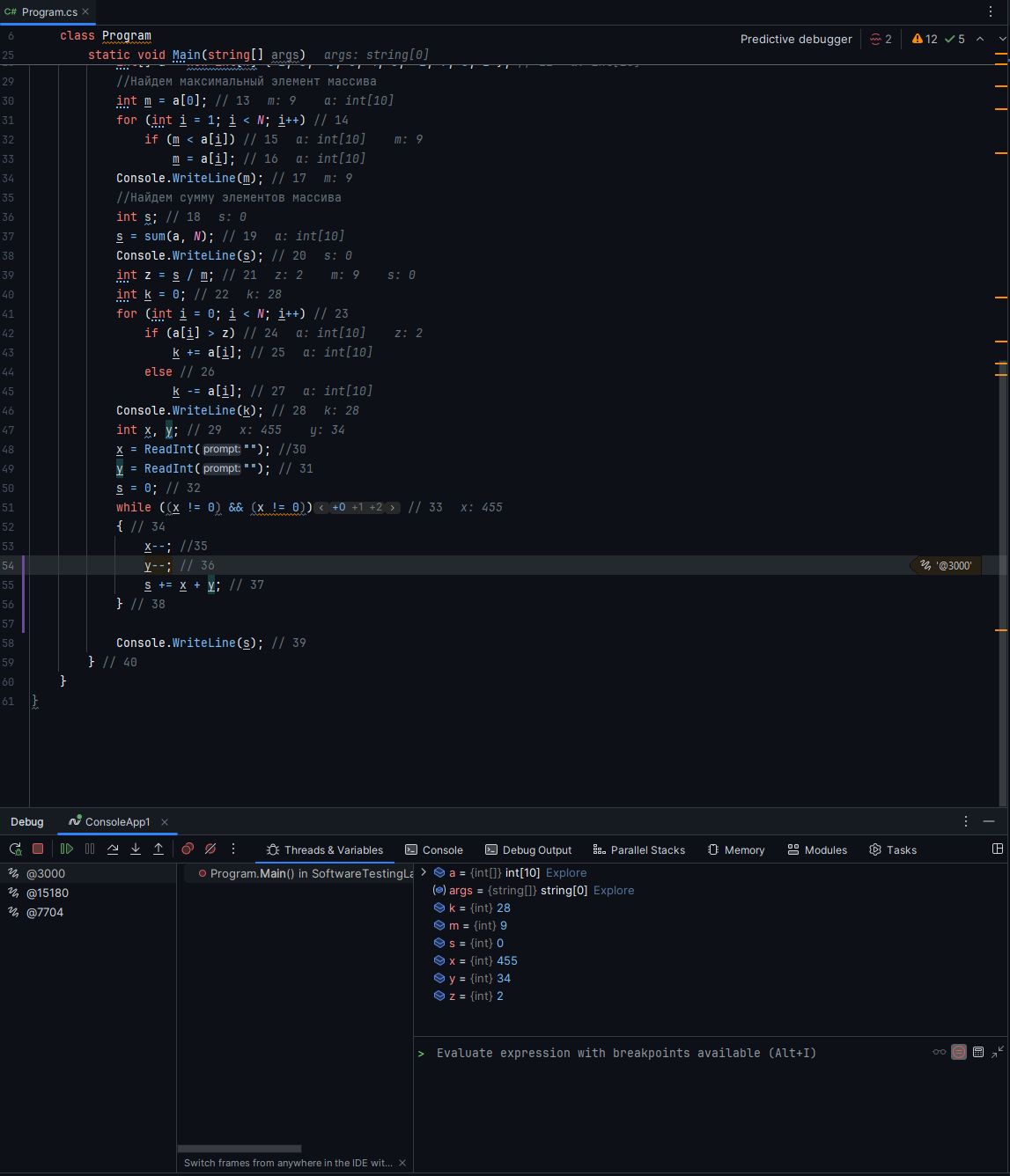


Рисунок 3- процесс выполнения подпунктов 10-12

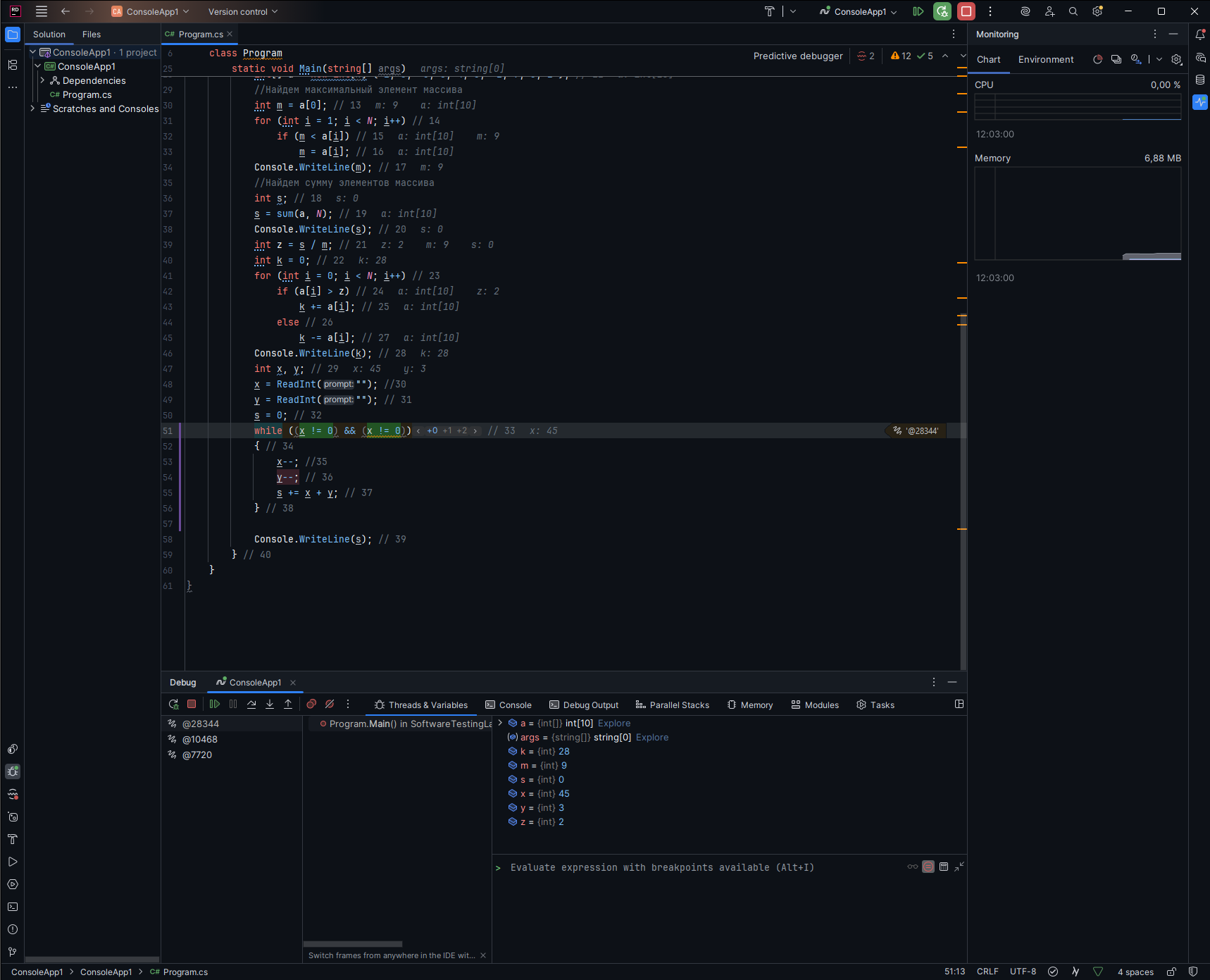


Рисунок 4 – процесс выполнения подпунктов 14-16