1. Оценка характеристик
   1. Задача «Вычисление суммы первых 𝑁 членов ряда»

Вычислить сумму первых 𝑁 членов ряда 1, 3, 5, 7, … Количество суммируемых членов ряда 𝑁 задается с клавиатуры.

*Необходимо разработать программу для вычисления суммы первых N членов ряда и на основе лексического анализа исходного текста программы, оценить её качество с использованием метрик Джилба.*

* + 1. Реализация программы

Текст программы для реализации возможного решения поставленной задачи, разработанной с использованием языка программирования C#, приведен в табл. 1.

Таблица 1

Исходный текст программы

|  |  |
| --- | --- |
| Номера строк | Строки программы |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28 | using System;  namespace lab02  {  class Program  {  static void Main()  {  bool isRepeat;  do  {  int[] row = { 1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29 };  Console.WriteLine("Введите число элементов: ");  int numb = Convert.ToInt32(Console.ReadLine()), summ = 0;  for (int i = 0; i < numb; i++)  {  summ += row[i];  }  Console.WriteLine("Сумма элементов ряда: " + summ);  Console.WriteLine("Повторить?(Y/N)");  char input = Convert.ToChar(Console.ReadLine());  if (input == 'Y' || input == 'y')  isRepeat = true;  else  isRepeat = false;  } while ( isRepeat );  }  }  } |

Программа, разработанная для реализации заданного алгоритма, имеет циклы и один оператор ветвления. Следовательно, из перечисленного набора характеристик можно определить следующие: 𝐿, , , 𝐶𝐿 и 𝑐𝑙.

* + 1. Словарь программы

При подсчете общего количества операторов 𝐿 программы будем руководствоваться правилами, определенными при подсчете операторов в метриках Холстеда. В табл. 2 приведены операторы и операции, используемые в программе.

# Таблица 2

Словарь операторов и операций программы

|  |  |  |  |
| --- | --- | --- | --- |
| № п/п | Операторы, операции | Номера строк | Количество повторений |
| 1 | using … | 1 | 1 |
| 2 | namespace … | 2 | 1 |
| 3 | class … | 4 | 1 |
| 4 | static void … | 6 | 1 |
| 5 | bool … | 8 | 1 |
| 6 | do … while() | 9 – 25 | 1 |
| 7 | int[] … | 11 | 1 |
| 8 | Console.WriteLine() | 12, 18, 19 | 3 |
| 9 | int … | 13, 14 | 1 |
| 10 | Convert.ToInt32() | 13 | 1 |
| 11 | Console.ReadLine() | 13, 20 | 2 |
| 12 | for() | 14 | 1 |
| 13 | char … | 20 | 1 |
| 14 | Convert.ToChar() | 20 | 1 |
| 15 | if() … else … | 21(24) | 1 |
| 16 | [] | 11, 16 | 2 |
| 17 | += | 16 | 1 |
| 18 | < | 14 | 1 |
| 19 | == | 21, 21 | 2 |
| 20 | ++ | 14 | 1 |
| 21 | + | 18 | 1 |
| 22 | {} | 3(28), 5(27), 7(26), 10(25), 11(11), 15(17) | 6 |
| 23 | “” | 12, 18, 19 | 3 |
| 24 | () | 6, 12, 13, 13, 14, 18, 19, 20, 21, 25 | 10 |
| 25 | ; | 1, 8, 11, 12, 13, 14, 14, 16, 18, 19, 20, 22, 24, 25 | 14 |
| 26 | , | 11, 11, 11, 11, 11, 11, 11, 11, 11, 11, 11, 11, 11, 11, 13 | 15 |
| 27 | . | 12, 13, 13, 18, 19, 20, 20 | 7 |
| 28 | = | 11, 13, 13, 14, 20, 22, 24 | 7 |
| 29 | || | 21 | 1 |
| **Всего** | | | 89 |

* + 1. Оценка характеристик программы

Согласно теории Джилба, необходимо определить количество операторов условия, используемых в исходном тексте программы для реализации алгоритма решения поставленной задачи. В языке программирования С# к такой категории инструкций могут относиться условные операторы ветвления 𝑖𝑓 операторы циклов 𝑓𝑜𝑟 ... 𝑤ℎ𝑖𝑙𝑒 и 𝑑𝑜 ... 𝑤ℎ𝑖𝑙𝑒, которые в своем составе в обязательном порядке содержат логическое выражение, являющееся условием выполнения указанных операторов. В зависимости от соблюдения условия программа может существенно изменить ход выполнения, поэтому значимость условных операторов достаточно велика. Именно поэтому автор для них определил отдельную метрику.

В исходном тексте разработанной программы используются условный оператор ветвления if который располагается в строке 21, операторы циклов do ... while и for, в строках 9(25) и 14(17) соответственно.

* количество операторов цикла равно 2;
* количество операторов условия равно 1;
* абсолютная сложность ;
* относительная сложность программы равна:

Основываясь на результатах лексического анализа исходного текста программы можно сделать вывод о невысокой сложности представленного решения, т.к. количество ветвлений в программе достаточно небольшое (2 оператора цикла и 1 оператор условия), что подтверждается низким числом метрики относительной сложности программы, которое равно 0.0337.