Реализация программы на языке C#

|  |  |
| --- | --- |
| **Строка** | **Текст 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  29  30  31  32  33  34  36  37  38  39 | using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace ConsoleApp9  {  class Program  {  static void Main(string[] args)  {  Console.WriteLine("Введите строку ");  string input = "";  input = Console.ReadLine();  string mirror = "";  mirror= MirrorString(input);  Print(mirror);  Console.ReadLine();  }  public static string MirrorString(string input)  {  string mirror = "";  for (int i = input.Length - 1; i >= 0; i--)  {  mirror += input[i].ToString();  }  return mirror;  }  public static void Print(string mirror)  {  Console.WriteLine("Полученная строка: " + mirror);  Console.ReadLine();  }  }  } |

## Оценка характеристик программы

Рассмотрим значения 𝑓 для разработанного решения поставленной задачи, для определения количества функциональных указателей:

𝑓1 — количество внешних вводов (вводов данных пользователем) для рассматриваемой программы

𝑓1=1(13 строка-строка ввода, переменная input-строка 14)

𝑓2 — количество внешних выводов (отчеты, экраны, распечатки, сообщения)

𝑓2=1(34 строка-вывод результата, зеркально-отображенной строки)

𝑓3 – количество внешних запросов (диалоговых вводов - выводов)

𝑓3=0   
 Так как отсутствуют обращения к внутренним и внешним логическим файлам.

𝑓4 – количество локальных внутренних логических файлов

𝑓4=0

Так как в решении такие файлы отсутствуют.

𝑓5 – количество внешних интерфейсных файлов

𝑓5=0  
Так как в решении такие файлы отсутствуют.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Характеристика | Низкая сложность | Средняя сложность | Высокая сложность | Итого |
| Внешние вводы | 3\*1=3 | 0\*4=0 | 0\*5=0 | 3 |
| Внешние выводы | 4\*1=4 | 0\*5=0 | 0\*7=0 | 4 |
| Внешние запросы | 0\*3=0 | 0\*4=0 | 0\*6=0 | 0 |
| Внутренние логические файлы | 0\*7=0 | 0\*10=0 | 0\*15=0 | 0 |
| Внешние интерфейсные файлы | 0\*5=0 | 0\*7=0 | 0\*10=0 | 0 |
| Общее количество | | | | 7 |

Для внутренних и внешних файлов обращение к 1-19 элементам соответствует низкому уровню сложности, к 20-50 соответствует среднему уровню сложности и к более 50 - высокому уровню сложности. В соответствии с уровнем сложности значение характеристики умножается на коэффициент сложности. В таблице коэффициенты сложности расположены справа от знака умножения. При внешних вводах решения программы осуществляется обращение к одному элементу данных, что соответствует низкому уровню сложности. При внешних выводах также осуществляется обращение к одному элементу данных. Внешние запросы, внутренние и внешние файлы отсутствуют.

Определим теперь значения коэффициентов регулировки сложности к:, отвечая на поставленные вопросы:

1. Какое влияние имеет наличие средств передачи данных? Ответ: влияние существенное, тогда 𝑘1 = 5.

2. Какое влияние имеет распределенная обработка данных? Ответ: влияние случайное, поэтому 𝑘2 = 1.

3. Какое влияние имеет распространенность используемой аппаратной платформы? Ответ: влияние случайное, следовательно, 𝑘3 = 1.

4. Какое влияние имеет критичность к требованиям производительности и ограничению времени ответа? Ответ: влияние случайное, поэтому 𝑘4 = 1.

5. Какое влияние имеет частота транзакций? Ответ: влияние случайное, следовательно, 𝑘5 = 1.

6. Какое влияние имеет ввод данных в режиме реального времени? Ответ: влияние основное, отсюда 𝑘6 = 5.

7. Какое влияние имеет эффективность работы конечного пользователя? Ответ: влияние среднее, поэтому 𝑘7 = 3.

8. Какое влияние имеет оперативное обновление локальных файлов в режиме реального времени? Ответ: влияние существенное, тогда 𝑘8 = 1.

9. Какое влияние имеет скорость обработки данных (вычислений)? Ответ: влияние случайное, следовательно, 𝑘9 = 1.

10. Какое влияние имеют количество и категории пользователей? Ответ: влияние случайное, поэтому 𝑘10 = 1.

11. Какое влияние имеет легкость инсталляции? Ответ: влияние

случайное, тогда 𝑘11 = 1.

12. Какое влияние имеет легкость эксплуатации? Ответ: влияние случайное, поэтому 𝑘12 = 1.

13. Какое влияние имеет разнообразие условий применения? Ответ: влияние случайное, отсюда 𝑘13 = 1.

14. Какое влияние имеет простота внесения изменений? Ответ: влияние случайное, следовательно, 𝑘14 = 1.

Подсчитываем сумму коэффициентов:

=5+1+1+1+1+5+3+1+1+1+1+1+1+1=24

Подставляем полученные данные:

Метод ***Print*** осуществляет вывод на экран монитора перевёрнутую строку. Модуль реализует единственную прикладную функцию - тип связности - функциональный, сила связности - 10.

Метод 𝑴𝒂𝒊𝒏 осуществляет решение поставленной задачи. Модуль реализует не единственную прикладную функцию. Действия внутри модуля связаны. Порядок действия внутри модуля важен - тип связности - коммуникационный, сила связности - 7.

Метод ***MirrorString*** осуществляет зеркальное отображение строки.  
Модуль реализует единственную прикладную функцию, следовательно, тип связности - функциональный, сила связности - 10.

Таким образом, в решении преобладают модули с силой связности 7 и 10, что говорит о достаточно высоком качестве программы, легкой ее тестируемости.

**Сцепление:**

Метод 𝒑𝒓𝒊𝒏𝒕 является вызываемым методом, и его входными параметрами являются структурные данные (массивы). Поэтому рассматриваемый модуль имеет сцепление по образцу (сила сцепления 3 ).

Метод ***MirrorString***  является вызываемым, и его входными параметрами являются простые и структурные данные(сила сцепления 3).

Метод 𝑴𝒂𝒊𝒏 является вызывающим и передает вызываемым модулям списки управляющих параметров, явно влияющих на их работу. Следовательно, метод 𝑴𝒂𝒊𝒏 имеет сцепление по управлению (сила сцепления 4).

Сила сцепления программных модулей рассматриваемого решения составляет 3-4, что говорит об уровне качества программы чуть выше среднего.