В одномерном динамическом массиве, состоящем из n элементов, вычислить среднеарифметическое значение элементов массива.

Тестирование белого – тестирование, которое учитывает внутренние механизмы системы или компонента. То есть для тестирования таким образом необходимо знать код, чтобы разработать тест-кейсы и проверить все «пути».

1. Покрытие операторов

Самый первый оператор проверяет ввод размерности массива. Если он меньше 0 или равен 0, то программа будет запрашивать повторный ввод (Console.WriteLine("Введите другое число");) до тех пор, пока не будет введено корректное значение. В ином случае пользователь может закрыть программу.

Если ввод размерности массива был корректен программа запрашивает ввод элементов.

Таблица 1 – Результат тестирования методом покрытия операторов

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № теста | Тест | Ожидаемый результат | Фактический результат | Результат тестирования |
| 1 | N = 0 | Программа запросит повторный ввод | Программа запросила повторный ввод | Пройден |
| 2 | N = -5 | Программа запросит повторный ввод | Программа запросила повторный ввод | Пройден |
| 3 | N = 3 | Программа запросит ввод элементов | Программа запросила ввод элементов | Пройден |

1. Покрытие решений

При покрытии решений должны быть протестированы все переходы из точки входа. Таким образом при введении количества элементов всегда будет один исход для продолжения программы, в ином случае будет повторный запрос ввода, пока значение не будет корректным.

Таблица 2 –Результат тестирования методом покрытия решений

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № теста | Тест | Ожидаемый результат | Фактический результат | Результат тестирования |
| 1 | N = 3;  A=2;  B=3;  C=4 | Программа выведет результат равный 3 | Программа вывела результат равный 3 | Пройден |
| 2 | N = 10;  A=1;  B=3;  C=5;  D=6;  E=7;  F= 10;  G=11;  H=15;  J=9;  K=2; | Программа выведет результат равный 6,9 | Программа вывела результат равный 6,9 | Пройден |

1. Покрытие условий

В покрытии условий проверяется уже не блок в целом, а каждое условие. Так в блоке ввода размерности сначала проверится, что значение больше 0 и только после этого будет проверено, что оно меньше константы. Если первое условие будет выполнено, а второе нет (n = 150), то исход будет такой же, как если бы и первое условие не было выполнено (запрос на вывод будет вызван повторно).

Таблица 3 – Результат тестирования методом покрытия условий

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № теста | Тест | Ожидаемый результат | Фактический результат | Результат тестирования |
| 1 | N = 0 | Программа запросит повторный ввод | Программа запросила повторный ввод | Пройден |
| 2 | N = -1 | Программа запросит повторный ввод | Программа запросила повторный ввод | Пройден |
| 3 | N = 10 | Программа запросит ввод элементов | Программа запросила ввод элементов | Пройден |

1. Покрытие решений, условий

Критерий покрытия решений, условий требует такого достаточного набора тестов, чтобы все возможные результаты каждого условия в решении выполнялись по крайней мере один раз, все результаты каждого решения выполнялись по крайней мере один раз и, кроме того, каждой точке входа передавалось управление по крайней мере один раз.

Таблица 4 – Результат тестирования методом покрытия решений, условий

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № теста | Тест | Ожидаемый результат | Фактический результат | Результат тестирования |
| 1 | N = 0 | Программа запросит повторный ввод | Программа запросила повторный ввод | Пройден |
| 2 | N = -1 | Программа запросит повторный ввод | Программа запросила повторный ввод (Рисунок 7) | Пройден |
| 3 | N = 10 | Программа запросит ввод элементов | Программа запросила ввод элементов | Пройден |
| 4 | N = 3;  A=2;  B=3;  C=4 | Программа выведет результат равный 3 | Программа вывела результат равный 3 | Пройден |
| 5 | N = 10;  A=1;  B=3;  C=5;  D=6;  E=7;  F= 10;  G=11;  H=15;  J=9;  K=2; | Программа выведет результат равный 6,9 | Программа вывела результат равный 6,9 | Пройден |

1. Комбинированное покрытие условий

Такое покрытие подразумевает все возможные комбинации для проверки кода.

1. n=-1 (Программа запрашивает повторный ввод)

2. n=0 (Программа запрашивает повторный ввод)

3. n=3

a[0]=2

a[1]=3

a[2]=4

(Будет получен ответ)

**Приложение**

Код программы:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Polkovnikova

{

internal class Program

{

static void Main(string[] args)

{

int r = 1;

while (r == 1)

{

Console.Write("Введите количество элементов N: ");

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

double sum = 0;

double sr;

double res;

if (n > 0)

{

double[] array = new double[n];

Console.WriteLine("Ввод элементов массива");

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

{ array[i] = Convert.ToDouble(Console.ReadLine()); }

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

{ sum = sum + array[i]; }

Console.WriteLine("Сумма элементов массива =" + sum);

sr = sum / n;

Console.WriteLine("Среднеарифметическое =" + sr);

}

else if (n == 0)

{ Console.WriteLine("Введите другое число"); }

else { Console.WriteLine("Введите положительное число"); }

Console.WriteLine("Продолжить работу?(1/0)");

r = Int32.Parse(Console.ReadLine());

Console.Clear();

Console.ReadKey();

}

}

}

}