МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

Федеральное государственное бюджетное

образовательное учреждение

высшего образования

ВЯТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ФАКУЛЬТЕТ КОМПЬЮТЕРНЫХ И ФИЗИКО-МАТЕМАТИЧЕСКИХ НАУК

КАФЕДРА ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ №4

**по дисциплине**

**“** **Тестирование программного обеспечения”**

**Тестирование ПО методами «Белого ящика»**

Выполнил: студент гр. ФИб-4302-51-00 Шатунов Д. И. \_\_\_\_\_\_

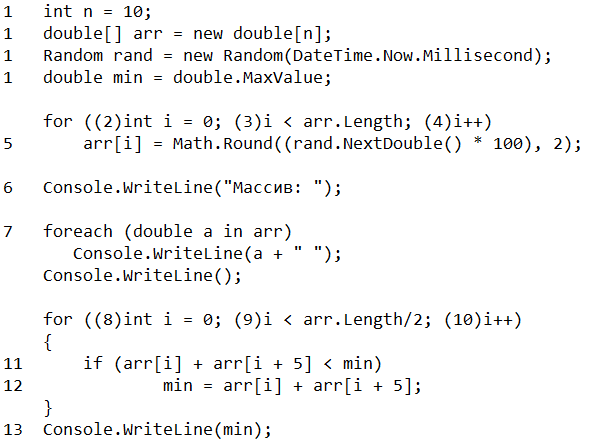
Проверил: старший преподаватель кафедры ПМиИ Фищева И.Н. \_\_\_\_\_

Киров 2022

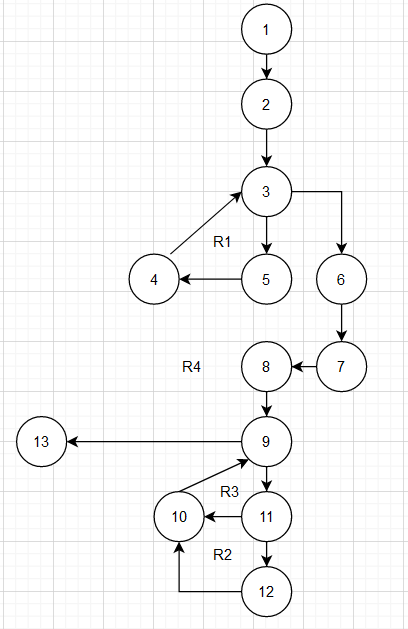
**Вариант 13**

**Задание:** Дан одномерный массив A1, A2, ..., A10 целых чисел. Получить наименьшее среди А1+А6, А2+А7, ..., А5+А10

**Текст программы:**

****

**Потоковый граф**



**Цикломатическая сложность потокового графа**

V(G) = 4 региона.

**Базовое множество независимых линейных путей:**

Путь 1: 1-2-3-6-7-8-9-13. Данный путь невозможен, так как невозможно избежать цикла, в котором массив заполняется.

Путь 2: 1-2-(3-5-4)-6-7-8-(9-11-10)-13. Данный путь невозможен, так как невозможно получить сумму двух элементов (в диапазоне от 100 до -100) большую чем double.MaxValue.

Путь 3: 1-2-(3-5-4)-6-7-8-(9-11-12-10)-13. Нормальный вариант функционирования программы.

Путь 4: 1-2-3-6-7-8-9-13. Данный путь невозможен, так как невозможно пропустить цикл.

**Построение тестовых вариантов**

Тестовый вариант 1 ТВ1:

ИД: Массив: {0.1, 10.2, 3.56, 1.2, 0.1, 0.4, 11.4, 90.1, 4.23, 1.54}

ОЖ.РЕЗ.: Минимальная сумма двух элементов для данного массива = 0.5

**Тест**

[Test]

public void CalculationTestCorrect()

{

double[] arr = {0.1, 10.2, 3.56, 1.2, 0.1, 0.4, 11.4, 90.1, 4.23, 10.24};

double a = Point.Calculation(arr);

Assert.AreEqual(0.5, a);

}

**Результаты тестирования**

****

C1 = for (int i = 0; i < arr.Length / 2; i++)

C2 = arr[i] + arr[i + 5] < min

Оуc = (d1)

d1 = (>, <, =)

**Тестовые варианты**

С1: при d1 = «>» происходит выход из условия

При d1 = «<» переменная i увеличивается на 1

При d1 = «=» происходит выход из условия

C2: при d1 = «>» происходит выход из условия

При d1 = «<» минимальный элементом становится сумма - arr[i] + arr[i + 5]

При d1 = «=» происходит выход из условия

**Тесты С1**

[Test] // при i = 10 (>) программа не зашла в цикл for и не дошла до условия if

public void CalculationWhenIEquals10()

{

int i = 10;

double[] arr = { 0.1, 10.2, 3.56, 1.2, 0.1, 0.4, 11.4, 90.1, 4.23, 10.24 };

double a = Point.Calculation2(arr, 10);

Assert.AreEqual(double.MaxValue, a);

}

[Test] // при i = 0 (<) программа зашла в цикл for и выполнила условие if

public void CalculationThenIEquals0()

{

int i = 0;

double[] arr = { 0.1, 10.2, 3.56, 1.2, 0.1, 0.4, 11.4, 90.1, 4.23, 10.24 };

double a = Point.Calculation2(arr, i);

Assert.AreEqual(0.5, a);

}

[Test] // при i = 5 (=) программа не зашла в цикл for и не дошла до условия if

public void CalculationWhenIEquals5()

{

int i = 5;

double[] arr = { 0.1, 10.2, 3.56, 1.2, 0.1, 0.4, 11.4, 90.1, 4.23, 10.24 };

double a = Point.Calculation2(arr, i);

Assert.AreEqual(double.MaxValue, a);

}

**Тесты С2**

[Test] // при min = double.MinValue (>) программа не зашла в условие if и вернула min обратно неизменный

public void CalculationWhenMinEqualsMinValue()

{

double min = double.MinValue;

double[] arr = { 0.1, 10.2, 3.56, 1.2, 0.1, 0.4, 11.4, 90.1, 4.23, 10.24 };

double a = Point.Calculation3(arr, min);

Assert.AreEqual(double.MinValue, a);

}

[Test] // при min = double.MaxValue (<) программа зашла в условие if получила новый min и вернула его

public void CalculationWhenMinEqualsMaxValue()

{

double min = double.MaxValue;

double[] arr = { 0.1, 10.2, 3.56, 1.2, 0.1, 0.4, 11.4, 90.1, 4.23, 10.24 };

double a = Point.Calculation3(arr, min);

Assert.AreEqual(0.5, a);

}

[Test] // при min = arr[i] + arr[i+5] (=) программа не заходит в условие if

// и получает минимум равный последней итерации цикла for в данном случае (0.1+10.24 = 10.34)

public void CalculationWhenMinEqualsArrElementsSum()

{

double min = double.MaxValue;

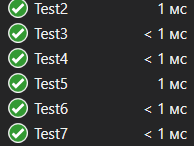
double[] arr = { 0.1, 10.2, 3.56, 1.2, 0.1, 0.4, 11.4, 90.1, 4.23, 10.24 };

double a = Point.Calculation4(arr, min);

Assert.AreEqual(10.34, a);

}

**Результаты тестирования**



**Вывод:** в ходе лабораторной работы были изучены методы тестирования с помощью методов «белого ящика». Данные методы хорошо использовать в ситуациях, когда программа содержит большое количество циклов и условий, так как есть возможность протестировать все различные варианты работы циклов и условий.