Федеральное агентство связи
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Сибирский государственный университет телекоммуникаций и информатики»

Лабораторная работа №1
Вариант 2

Выполнил:
студенты 4 курса
группы ИП-216
Андрущенко Ф.А.
Литвинов А. Е.
Русецкий А. С.

Проверил:
преподаватель кафедры ПМиК
Агалаков Антон Александрович

Новосибирск, 2025 г.

Задание 1

Разработайте на языке C# класс, содержащий функции в соответствии с вариантом задания. Разработайте тестовые наборы данных по критерию C0 для тестирования функций класса. Протестируйте созданный класс с помощью средств автоматизации модульного тестирования Visual Studio. Напишите отчёт о результатах проделанной работы.

Функции:

- Функция получает два одномерных целочисленных массив a, b одинаковой длины. Возвращает массив, полученный суммированием компонентов массивов a и b с чётными значениями.
- Функция получает одномерный массив вещественных переменных и целое – параметр сдвига. Функция изменяет массив циклическим сдвигом значений его элементов влево на число позиций, равное параметру сдвига.
- Функция находит и возвращает индекс начала первого вхождения последовательности целых чисел, представленных массивом int[] seq в другую последовательность, представленную массивом int[] vec.

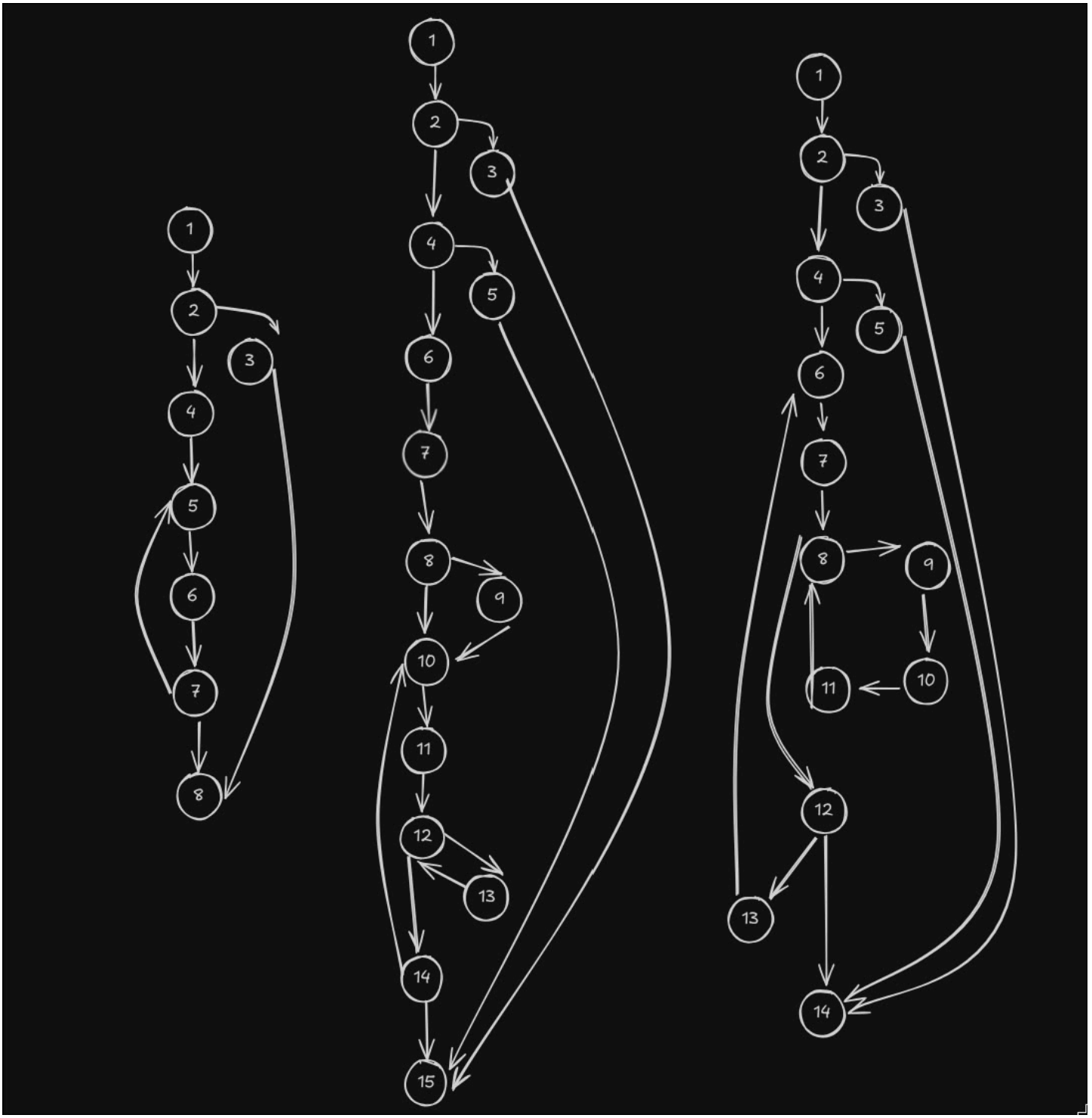
УГП и тестовые наборы данных для тестирования функций класса

Пример тестовых данных:

```
using Lab1;

namespace Lab1
{
    class Program
    {
        static void Main()
        {
            // int[] a = { 1, 2, 3, 4, 5, 6, 7, 8, 9 };
            // int[] b = { 23, 48, 11, 12, 12, 32, 45, 56, 98 };
            // int[] task_1 = ArrayOperations.SumEvenElements(a, b);
            // foreach (int i in task_1)
            // {
            //     Console.WriteLine(i);
            // }

            // double[] task_2 = { 1.0, 2.0, 3.0, 4.0, 5.0 };
            // ArrayOperations.CyclicShiftLeft(task_2, 2);
            // foreach (double i in task_2)
            // {
            //     Console.Write(i + " ");
            // }
            // Console.WriteLine();

            // int[] vec = { 1, 2, 3, 4, 5 };
            // int[] seq = { 2, 3 };
```

```
            // int task_3 = ArrayOperations.FindSequenceIndex(vec, seq);
            // Console.WriteLine(task_3);
        }
    }
}
```

УГП для функций (слева направо) :



Исходные коды программ

Тестируемые функции

```
namespace Lab1
{
    // Вариант 2
    public class ArrayOperations
    {
        /* Функция получает два одномерных целочисленных массив a, b
        одинаковой длины. Возвращает массив, полученный суммированием
        компонентов массивов a и b с чётными значениями. */
        public static int[] SumEvenElements(int[]? a, int[] b)
        {
            if (a == null || b == null || a.Length != b.Length)
                throw new ArgumentException("Массивы должны быть ненулевыми и одинаковой
длины!");

            List<int> sum = new List<int>();

            for (int i = 0; i < a.Length; i++)
                if (a[i] % 2 == 0 && b[i] % 2 == 0)
                    sum.Add(a[i] + b[i]);

            return sum.ToArray();
        }

        /* Функция получает одномерный массив вещественных переменных и
        целое - параметр сдвига. Функция изменяет массив циклическим сдвигом
        значений его элементов влево на число позиций, равное параметру сдвига. */
        public static void CyclicShiftLeft(double[]? array, int shift)
        {
            if (array == null)
                throw new ArgumentNullException(nameof(array));
            if (array.Length == 0 || shift == 0)
                return;

            int arraySize = array.Length;
            shift = shift % arraySize;

            if (shift > 0)
                shift = -shift;

            for (int i = 0; i > shift; i--)
            {
                double tmp = array[0];

                for (int j = 1; j < arraySize; j++)
                {
                    array[j - 1] = array[j];
                }
                array[arraySize - 1] = tmp;
```

```
        }
    }
    /* Функция находит и возвращает индекс начала первого вхождения
    последовательности целых чисел, представленных массивом int[] seq в
    другую последовательность, представленную массивом int[] vec. */
    public static int FindSequenceIndex(int[]? vec, int[]? seq)
    {
        if (vec == null || seq == null)
            throw new ArgumentNullException("Массивы не должны быть нулевыми!");
        if (seq.Length == 0 || seq.Length > vec.Length)
            return -1;

        for (int i = 0; i <= vec.Length - seq.Length; i++)
        {
            bool found = true;
            for (int j = 0; j < seq.Length; j++)
            {
                if (vec[i + j] != seq[j])
                {
                    found = false;
                    break;
                }
            }
            if (found)
                return i;
        }

        return -1;
    }
  }
}
```

Тесты

```
using Lab1;

namespace Lab1Tests
{
    [TestClass]
    public class ArrayOperationsTests
    {
        // Тесты для SumEvenElements
        [TestMethod]
        public void SumEvenElements_BothEven_ReturnsSum()
        {
            // Проверяет успешное суммирование чётных элементов
            int[] a = { 2, 4 };
            int[] b = { 2, 6 };
            int[] expected = { 4, 10 };
            int[] result = ArrayOperations.SumEvenElements(a, b);
            CollectionAssert.AreEqual(expected, result);
        }
```

```
[TestMethod]
[ExpectedException(typeof(ArgumentException))]
public void SumEvenElements_NullArray_ThrowsException()
{
    // Проверяет выброс исключения при null
    ArrayOperations.SumEvenElements(null, new int[] { 1, 2 });
}

[TestMethod]
public void SumEvenElements_NoEvenPairs_ReturnsEmpty()
{
    // Проверяет случай без чётных пар
    int[] a = { 1, 3 };
    int[] b = { 1, 3 };
    int[] expected = { };
    int[] result = ArrayOperations.SumEvenElements(a, b);
    CollectionAssert.AreEqual(expected, result);
}

// Тесты для CyclicShiftLeft
[TestMethod]
public void CyclicShiftLeft_ValidShift_ShiftsArray()
{
    // Проверяет сдвиг влево
    double[] array = { 1.0, 2.0, 3.0 };
    double[] expected = { 3.0, 1.0, 2.0 };
    ArrayOperations.CyclicShiftLeft(array, 2);
    CollectionAssert.AreEqual(expected, array);
}

[TestMethod]
[ExpectedException(typeof(ArgumentNullException))]
public void CyclicShiftLeft_NullArray_ThrowsException()
{
    // Проверяет выброс исключения при null
    ArrayOperations.CyclicShiftLeft(null, 1);
}

[TestMethod]
public void CyclicShiftLeft_EmptyArray_NoChange()
{
    // Проверяет обработку пустого массива
    double[] array = { };
    double[] expected = { };
    ArrayOperations.CyclicShiftLeft(array, 5);
    CollectionAssert.AreEqual(expected, array);
}

// Тесты для FindSequenceIndex
[TestMethod]
public void FindSequenceIndex_SequenceFound_ReturnsIndex()
```

```
        {
            // Проверяет успешное нахождение последовательности
            int[] vec = { 1, 2, 3, 4 };
            int[] seq = { 2, 3 };
            int expected = 1;
            int result = ArrayOperations.FindSequenceIndex(vec, seq);
            Assert.AreEqual(expected, result);
        }

        [TestMethod]
        [ExpectedException(typeof(ArgumentNullException))]
        public void FindSequenceIndex_NullVec_ThrowsException()
        {
            // Проверяет выброс исключения при null
            ArrayOperations.FindSequenceIndex(null, new int[] { 1 });
        }

        [TestMethod]
        public void FindSequenceIndex_NoSequence_ReturnsMinusOne()
        {
            // Проверяет случай, когда последовательность не найдена
            int[] vec = { 1, 2, 3 };
            int[] seq = { 4 };
            int expected = -1;
            int result = ArrayOperations.FindSequenceIndex(vec, seq);
            Assert.AreEqual(expected, result);
        }
    }
}
```

Результаты выполнения модульных тестов

```
❯ dotnet test Labs.sln
Restore complete (0.6s)
  Lab1 succeeded (1.6s) → Lab 1/bin/Debug/net9.0/Lab1.dll
  Tests succeeded (0.4s) → Lab 1 Tests/bin/Debug/net9.0/Tests.dll
Test Parallelization enabled for /home/dev/Documents/Sibsutis/Курс 4/Семестр 1/МРТ/Lab 1/Lab 1 Tests/bin/Debug/net9.0/Tests.dll (Workers: 16, Scope: MethodLevel)
  Tests test succeeded (0.6s)

Test summary: total: 9, failed: 0, succeeded: 9, skipped: 0, duration: 0.6s
Build succeeded in 3.6s
```

Выводы по выполненной работе

В ходе выполнения практической работы был успешно разработан и протестирован класс ArrayOperations, содержащий три статические функции для работы с массивами: SumEvenElements, CyclicShiftLeft, FindSequenceIndex. Для каждой функции были построены УГП и разработаны тестовые наборы данных по критерию C0, обеспечивающие полное покрытие всех ветвей кода, включая проверку корректной обработки нормальных данных, граничных случаев (пустые массивы, массивы с одним элементом, все нулевые элементы) и исключительных ситуаций (нулевые ссылки, недопустимые индексы). Модульное тестирование с помощью MSTest подтвердило корректность работы всех функций, а анализ покрытия кода показал 100% покрытие тестируемого кода, что свидетельствует о полноте разработанных тестов и качестве реализации класса.