# Лабораторная работа № 4

Работа с однородными структурированными типами данных.  
Элементы пользовательского интерфейса по работе с массивами.

**Цель работы:** получить практические навыки работы с компонентами для обработки однородных структурированных данных: научится работать с компонентом dataGridView для решения задач по обработке одномерных динамических массивов.

**Ход работы:**

1. Прочитайте свой вариант задания (см. ниже). Запишите в отчёт, что дано и что надо найти, с указанием типов исходных данных и результатов.
2. Создайте метод программиста для обработки массива по своему заданию, выполняющий основные вычисления или преобразования в массиве. Метод может находиться в классе формы или в дополнительном классе. В качестве параметра нужно передать методу одномерный массив.

**Пример метода обработки одномерного массива:**

/// <summary>

/// Метод для подсчёта среднего арифметического положительных эл. массива.

/// </summary>

/// <param name="mas">double[] mas исходный массив.</param>

/// <returns>Среднее арифметическое положительных эл. массива.</returns>

public double averagePositiv(double[] mas)

{

double s = 0;

int k = 0;

for (int i = 0; i < mas.Length; i++)

{

if (mas[i] > 0)

{

s += mas[i];

k++;

}

}

if (k == 0)

throw new Exception("В массиве нет положительных!");

return s/k;

}

1. Для того, чтобы протестировать созданный метод, разработайте тест-кейсы. Добавьте в решение тестовый проект. Добавьте в тестовом проекте ссылку на пространство имен исходного проекта. Тестируемый класс следует сделать публичным (на время тестирования).

**Примеры тест-кейсов для проверки метода:**

[TestMethod]

public void TestMethod1()

{

// Исходные данные для теста.

Double[] myArray = { 2.4, -6.1, -3.9, 5.6 };

// Ожидаемое значение (2.4 + 5.6)/2 = 4.0

Double experted = 4.0;

// Вызов тестируемой функции.

Form1 form1 = new Form1();

Double actual = form1.averagePositiv(myArray);

Assert.AreEqual(experted, actual, 0.00, "Ожидаемое среднее арифметическое положительных элементов массива не было получено!");

}

[TestMethod]

public void TestMethod2()

{

// Исходные данные для теста.

Double[] myArray = { -1.2, -9.6, -11.5, -7.8 };

// Вызов тестируемой функции.

Form1 form1 = new Form1();

try

{

Double actual = form1.averagePositiv(myArray);

}

catch (Exception ex)

{

// Содержит ли строка заданное сообщение.

StringAssert.Contains(ex.Message, "В массиве нет положительных!");

return;

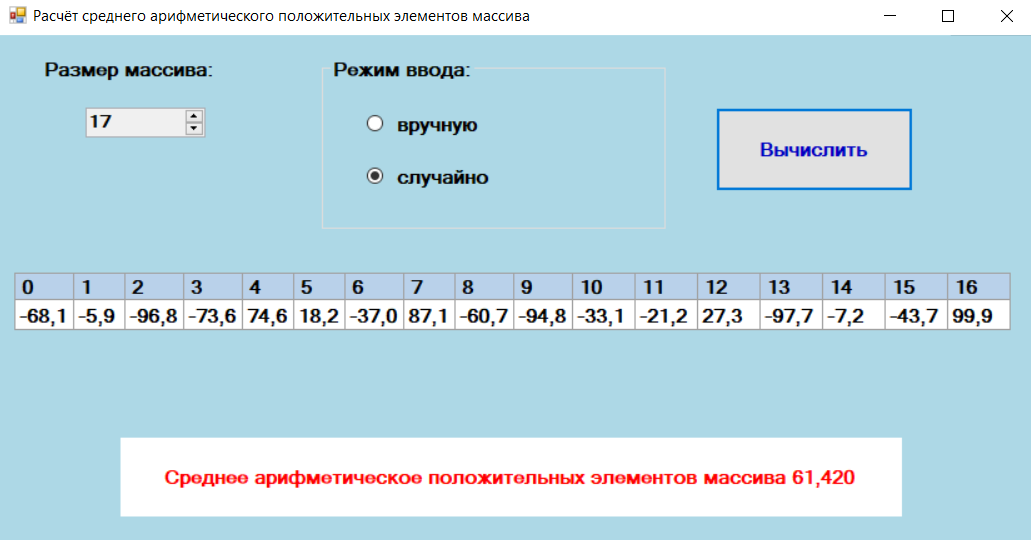
}

Assert.Fail("Ожидаемое исключение не было получено.");

}

1. Сохраните в отчет тест-кейсы и скриншот успешного тестирования.
2. Теперь можно заняться настройкой пользовательского интерфейса. Продумайте, какие элементы интерфейса необходимо разместить на форме, чтобы обеспечить удобную и наглядную работу пользователя. Ввод массива нужно осуществлять с клавиатуры или случайным образом в компонент dataGridView. Способ ввода будем менять с помощью зависимых кнопок выбора.

*Примерный вид формы при запуске:*



1. Выполните настройку свойств компонента dataGridView. Так как массив одномерный, установим количество строк таблицы 1, количество колонок по умолчанию 10. Столбец заголовка нам не нужен, установим свойству **RowHeadersVisible** значение false. Занесите в отчёт в специальную таблицу, у каких компонентов вы изменяли свойства, и какое значение получило теперь каждое свойство.

*Указания:*

**Основные свойства табличных компонентов**

**RowCount, ColumnCount** – задают число строк и столбцов таблицы (нумерация строк и столбцов начинается с нуля). Путем задания новых значений этих свойств можно динамически менять размерность таблицы.

**RowHeadersVisible, ColumnHeadersVisible** – задают будут ли отображаться заголовки строк и столбцов.

**ColumnHeadersHeights, RowHeadersWidths** – задают высоту и ширину ячеек таблицы. Размеры таблицы и её ячеек задаются в пикселях и могут изменяться во время работы программы по содержимому ячеек.

**ColumnHeadersHeightSizeMode** – определяет поведение при изменении высоты заголовков столбцов (установите это свойство в значение EnableResizing, в этом случае высота заголовков столбцов не будет изменяться по содержимому, и мы сможем в конструкторе задать высоту строки заголовка).

**AutoSizeColumnMode** – определяет режим автоматической установки размера для видимых столбцов (установите значение Fill).

**AllowUserToAddRows** – определяет, есть ли возможность добавления строк (установите значение false).

**AllowUserToDeleteRows** – определяет, есть ли возможность удаления строк (установите значение false).

**dataGridView1.Rows[i].Cells[j].Value** – свойство используется для доступа к отдельной ячейке с указанием номера столбца и номера строки. Используя цикл для перебора номеров строк и столбцов таблицы, можно поочередно обращаться к каждой ячейке.

Чтобы пользователь мог редактировать содержимое ячеек таблицы, необходимо свойству **ReadOnly** присвоить значение false.

1. Добавьте на форму поле ввода с кнопкой-переключателем numericUpDown. При нажатии на кнопку-переключатель из поля ввода будет считываться размер массива, выделятся память под динамический массив, устанавливаться соответствующее количество ячеек таблицы и её ширина. Добавьте процедуру нажатия на кнопку numericUpDown и внесите необходимые команды в тело метода.
2. При переключении кнопок-переключателей radioButton в контейнере groupBox будет меняться способ ввода массива. Создайте метод обработки события щелчка по контейнеру. При выборе пользователем кнопки «Вручную» таблица на форме должна заполняться нулями. При выборе пользователем кнопки «Случайно» таблица на форме должна заполняться случайными числами в интервале   
   [-100; 100]. Внесите необходимые команды в тело процедуры.
3. Все расчёты в программе будут происходить при нажатии на кнопку «Вычислить». Создайте соответствующий метод обработки события. При нажатии на кнопку «Вычислить» значения из таблицы должны считываться в массив, а затем с этим массивом можно производить любые действия и вычисления по заданию. Разработайте алгоритм метода нажатия на кнопку «Вычислить» и скопируйте его в отчёт. Внесите необходимые команды в тело процедуры.
4. Запустите проект на компиляцию. При необходимости исправьте синтаксические ошибки, выявленные на этапе компиляции.
5. Запустите проект на выполнение. Протестируйте программу. Отследите, какие ошибки выполнения возникают при недопустимости исходных данных, например, если пользователь оставил одну или несколько ячеек таблицы пустыми, ввел в ячейку недопустимые символы. Подумайте, как можно предотвратить такие ошибки.Добавьте методы проверки правильности ввода данных в таблицу dataGridView.
6. Протестируйте программу при различных значениях размерности массива, в режиме ручного и случайного ввода. Скопируйте в отчёт скриншоты различных вариантов выполнения программы.
7. Скопируйте в отчёт текст программного модуля с комментариями.

**Подведение итогов лабораторной работы:**

Результатом выполнения лабораторной работы должен быть проект, сохранённый в отдельной папке, и отчёт. Отчёт должен содержать:

* название лабораторной работы, цель работы;
* текст вашего задания с указанием номера варианта;
* математическая формулировка задачи (что дано, что найти, с обозначением переменных и их типов);
* код основного метода обработки массива;
* тест-кейсы для проверки метода и скриншот успешного тестирования;
* скриншот формы с надписанными компонентами;
* таблица «Значения свойств, установленные на этапе конструирования интерфейса»;
* текст программного модуля с комментариями;
* скриншоты проекта на этапе выполнения;
* колонтитулы с вашей фамилией и номером группы.

Продемонстрируйте результаты работы преподавателю и получите оценку.

**Контрольные вопросы:**

1. Какие свойства таблицы dataGridView можно установить на этапе конструирования? Поясните назначение этих свойств.
2. Какое свойство таблицы dataGridView позволяет обратиться к определённой ячейке в программе?
3. Сколько методов получилось в классе формы, какие события они обрабатывают?
4. Как организовать подключение основного проекта к тестовому проекту, чтобы можно было вызвать вашу функцию? Как осуществить непосредственно вызов функции из тестового проекта?

# Варианты заданий к лабораторной работе № 4

**Вариант 1.**

В массиве из *n* вещественных чисел отсортировать в порядке возрастания все положительные элементы. Отрицательные и нулевые элементы оставить на своих местах.

**Вариант 2.**

В массиве из *n* вещественных чисел отсортировать в порядке убывания все отрицательные элементы. Положительные и нулевые элементы оставить на своих местах.

**Вариант 3.**

Подсчитать количество нулевых компонент массива из *n* целых чисел. Если нулевых элементов окажется больше двух, то поменять местами нулевые элементы с первым элементом, отличным от нуля.

**Вариант 4.**

Подсчитать количество отрицательных и положительных элементов массива из *n* целых чисел. Если отрицательных элементов окажется больше, то поменять знаки всех элементов на противоположные.

**Вариант 5.**

Переставить в обратном порядке все положительные элементы вещественного массива из *n* элементов.

**Вариант 6.**

Задать случайным образом массив из *n* целых чисел. Перестроить массив, записав в него сначала все положительные, затем все отрицательные элементы и нули.

**Вариант 7.**

Найти номер элемента массива из *n* вещественных чисел, значение которого наиболее близко к среднему арифметическому элементов массива. Если таких элементов несколько, то вывести все номера в порядке их следования.

**Вариант 8.**

Составить программу для ввода двух одномерных массивов *А* и *В* размерностью *n* элементов и вычисления их скалярного произведения по формуле:



**Вариант 9.**

Составить программу для ввода двух одномерных массивов *А* и *В* размерностью *n* элементов и вычисления массива *С* размерностью *n-1*, каждый элемент которого получен как сумма перемноженных пар соседних элементов исходных массивов:



**Вариант 10.**

Дана последовательность вещественных чисел . Вычислить значение *Y* по формуле: 

**Вариант 11.**

Дана последовательность вещественных чисел . Вычислить значения элементов массива *Y* по формулам:



**Вариант 12.**

Для заданных значений  вычислить значение величины *Y* по формуле:



**Вариант 13.**

Для заданных значений  вычислить значение величины *Y* по формуле:



**Вариант 14.**

Для заданных значений  вычислить значение величины *Y* по формуле:



**Вариант 15.**

Для заданных значений  вычислить значение величины *Y* по формуле:



**Вариант 16.**

Для заданных значений  вычислить значение величины *Y* по формуле:



**Вариант 17.**

Составить программу для ввода двух одномерных массивов *A* и *B* размерностью *2n* и вычисления величины *Y* по формуле:



**Вариант 18.**

Для заданных значений  и вычислить значения  по формуле:



**Вариант 19.**

Для заданных значений  и  вычислить значение *Z* по формуле:

, где функция F вычисляется как разность максимального и минимального значения указанного массива:



**Вариант 20.**

Для заданных значений  и  вычислить значение *Z* по формуле:

, где функция F вычисляется как произведение всех элементов указанного массива:

