

Практическое занятие № 10

Тема практического занятия: Работа с одномерными массивами.

Цель практического занятия: ознакомить с основными понятиями и навыками работы с одномерными массивами в языке программирования.

В результате выполнения данной работы обучающийся должен уметь:

1. Создавать и инициализировать одномерные массивы.
2. Обращаться к элементам массива.
3. Применять различные методы обработки массивов для работы с его элементами.
4. Использовать циклы и условные конструкции для решения задач, связанных с одномерными массивами.
5. Применять полученные знания и навыки для решения практических задач и задач программирования, требующих работу с одномерными массивами.

знать:

1. Основные понятия и термины, связанные с одномерными массивами.
2. Способы объявления и инициализации одномерных массивов.
3. Как обращаться к элементам массива с помощью индексации.
4. Особенности и правила использования циклов и условных конструкций при работе с массивами.

Перечень оборудования, необходимого для выполнения задания:

- Автоматизированные рабочие места по количеству обучающихся (процессор Intel Core i7 или аналогичный, БП 700 Вт, 32 Гб ОЗУ, SSD 512 Гб, HDD 4 ТБ SATA 7200 rpm, RTX 3060 12GB);
- Автоматизированное рабочее место преподавателя (процессор Intel Core i7 или аналогичный, БП 700 Вт, 32 Гб ОЗУ, SSD 512 Гб, HDD 4 ТБ SATA 7200 rpm, RTX 3060 12GB);
- Монитор 34", изогнутый, 3440x1440, 6 ms, 178°/178°, 300 cd/m2, 20M:1, HDMI, DP, USB, регулировка по высоте (1 монитор на одно рабочее место).
- Интерактивная панель 86" с OPS ПК

Программное обеспечение:

- Microsoft Visual Studio.

Общие теоретические сведения:

Одномерный массив - это структура данных, которая позволяет хранить и организовывать набор элементов одного типа в последовательность. Одномерный массив представляет собой непрерывный участок памяти, разделенный на ячейки, каждая из которых содержит один элемент массива. Каждая ячейка имеет свой уникальный индекс.

Одномерные массивы используются для хранения упорядоченных данных одного типа, таких как числа, символы или объекты. За счет

индексации по индексам можно обращаться к элементам массива для чтения и записи значений, а также выполнять различные операции, например, сортировку, поиск, добавление или удаление элементов.

При обработке массивов часто используются различные виды циклов. Особенности использования циклов при работе с массивами включают следующее:

1. Индексация: Циклы используются для обращения к каждому элементу массива по его индексу. Часто индекс начинается с 0 и увеличивается с каждой итерацией цикла.

2. Ограничение: Циклы обычно выполняются до достижения конечного условия. При обработке массивов это может быть количество элементов в массиве.

3. Итерация: Циклы позволяют выполнять однотипные операции над каждым элементом массива.

4. Вывод данных: Циклы, использованные для обработки массива, позволяют выводить данные всех элементов массива или выполнять над ними определенные операции.

5. Управление перебором: Циклы позволяют управлять процессом перебора элементов массива, таким образом, можно пропустить определенные элементы, выполнить операции только для определенных элементов или остановить цикл при выполнении определенного условия.

Через использование циклов в обработке массивов можно эффективно выполнять действия со всеми элементами массива без необходимости ручного обращения к каждому элементу, что значительно упрощает и ускоряет процесс работы с массивами.

Задание:

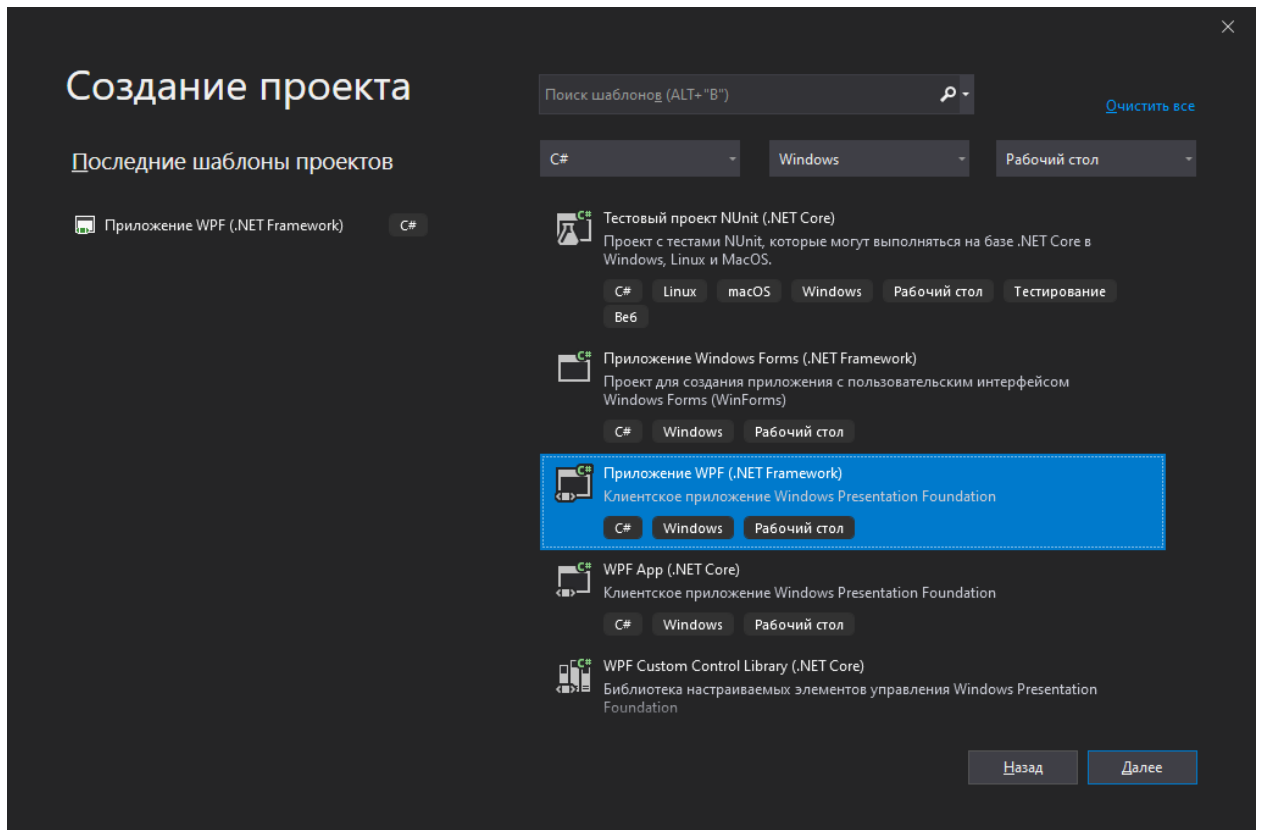
1. Реализуйте в приложении задачу №1: пользователем вводится массив температур воздуха за последние n дней (n – число, заданное пользователем). Необходимо вычислить среднее арифметическое температур. Результат необходимо выдать с округлением до двух знаков после запятой.
2. Реализуйте в приложении задачу №2: пользователем вводятся оценки студента за сессию. Необходимо определить, будет ли студент получать стипендию (условие получения стипендии – нет оценок, ниже 4).
3. Реализуйте в приложении задачу №3: пользователем вводится массив целых чисел. Необходимо записать этот массив в обратном порядке.

Указания по технике безопасности:

Инструкция по технике безопасности при работе в лаборатории, оборудованной компьютерной техникой.

Технология выполнения работы (этапы, последовательность действий):

1. Создайте проект для приложения. Например, это может быть приложение MS Visual Studio



2. Разработайте интерфейс приложения: главное окно и три вспомогательных (по одному окну для каждой задачи). Добавьте все необходимые элементы управления, в том числе и возврат к главному окну.
3. Реализуйте в каждом окне по задаче. При вводе массивов через текстовые поля используйте возможность разделения строки на элементы (в качестве разделителя используйте один пробел).

При реализации можно использовать примеры, описанные ниже.

Одномерный массив объявляется следующим образом:

<тип элементов> [] <имя массива>=new <тип элементов> [<число элементов>];

Например, ниже приведено описание массива целых чисел из десяти элементов

```
int[] a = new int[10];
```

Количество элементов может быть задано и с помощью заранее введенной переменной, например:

```
int[] a = new int[n];
```

Значения массиву можно присвоить сразу при объявлении. При этом не требуется указывать количество значений и выделять память, так как все значения будут сразу перечислены. Например, в массиве ниже сразу будет 3 элемента:

```
int[] a = { 1, 2, 3 };
```

Для определения длины такого массива, где число элементов напрямую не задавалось, можно воспользоваться свойством Length:

```
int k = a.Length;
```

Чтобы пользователь сам вводил массив с клавиатуры, можно воспользоваться методом Split, который будет строку из TextBox-а разбивать на массив строк, используя в качестве разделителя заданный (заданные) пользователем символы. Например, это может быть пробел:

```
//строковый массив, куда будут помещены числа из TextBox-а в строковом представлении
string[] sNums = T.Text.Split(' ');    //разделитель - пробел

//определение числа элементов в массиве
int n = sNums.Length;

//объявление массива целых чисел
int[] nums = new int[n];

//перенос элементов из строкового массива в целочисленный
for (int i = 0; i < n; i++)
    nums[i] = Convert.ToInt32(sNums[i]);
```

Пример кода с обработкой массива (меняются местами первый и последний элементы):

```

static void Main(string[] args)
{
    int[] a = new int[10000];
    Random rnd = new Random();
    for (int i = 0; i < 10000; i++)
        a[i] = rnd.Next(-1000, 1001);
    int m = a[0];
    int posMax = 0;
    for (int i = 0; i < a.Length; i++)
        if (a[i] > m)
        {
            m = a[i];
            posMax = i;
        }
    m = a[0];
    int posMin = 0;
    for (int i = 0; i < a.Length; i++)
        if (a[i] < m)
        {
            m = a[i];
            posMin = i;
        }
    a[posMin] = a[posMax];
    a[posMax] = m;
    foreach (int x in a)
        Console.WriteLine(x);
    Console.ReadKey();
}

```

Требование к отчету:

1. Приложение разработано, компилируется, запускается.
2. В приложении реализована корректная навигация между окнами.
3. Задача №1 решена корректно: средняя температура выдается верно, результат выдается с точностью до двух знаков после запятой.
4. Задача №2 решена корректно: наличие стипендии определяется верно.
5. Задача №3 решена корректно: массив записывается в обратном порядке.

Контрольные вопросы:

1. Что такое одномерный массив и как он организован в памяти?
2. Как объявить и инициализировать одномерный массив?
3. Как обратиться к элементу массива по его индексу?
4. Каковы способы чтения и записи значений в элементы массива?
5. Что такое индексация массива и с какого значения индекс начинается?
6. Какие операции можно выполнять над элементами массива?
7. Какие виды циклов используются для обработки массивов и в чем их особенности?
8. Какие контрольные механизмы можно использовать при работе с массивами?

9. Какие практические задачи могут быть решены с использованием одномерных массивов?

Основные и дополнительные источники, электронные ресурсы:

1. Подбельский, В. В. Язык C#. Базовый курс: учебное пособие / В. В. Подбельский. - 2-е изд., перераб. и доп. - Москва: Финансы и статистика, 2022. - 408 с. - ISBN 978-5-00184-079-4. - Текст: электронный. - URL: <https://znanium.com/catalog/product/1913989>
2. Гуриков, С. Р. Основы алгоритмизации и программирования на Python: учебное пособие / С.Р. Гуриков. — Москва: ИНФРА-М, 2023. — 343 с. — (Среднее профессиональное образование). - ISBN 978-5-16-016906-4. - Текст: электронный. - URL: <https://znanium.com/catalog/product/1927269>.