

Рязанский станкостроительный колледж РГРТУ

Основы алгоритмизации и программирования

**Тема 4. Структурированные типы данных.
Массивы.**

Рязань 2022

Оглавление

Массивы	3
Одномерные массивы	3
Действия над элементами одномерного массива	3
Инициализация (присваивание начальных значений элементам массива)	4
Создание массива (присваивание конкретных значений элементам массива)	4
Вывод значений элементов массива	4
Копирование массивов	4
Поиск элементов в массиве	4
Практическая работа №9	8
Двухмерные массивы	10
Действия над элементами двухмерного массива	10
Инициализация (присваивание начальных значений элементам массива)	10
Создание массива (присваивание конкретных значений элементам массива)	11
Вывод значений элементов массива	11
Практическая работа №10	15
Сортировка массивов	17
Алгоритм "Сортировка простым выбором".	17
Алгоритм "Сортировка пузырьком".	18
Алгоритм "Гномья сортировка".	19
Практическая работа №11	20

Массивы

В массивах объединены однотипные (логически однородные элементы), упорядоченные (урегулированные) по индексам, определяющим положение каждого элемента в массиве.

Проще говоря, массив – это набор данных одного типа, имеющий одно общее имя. Например, набор данных целого типа и т.д.

Одномерные массивы

Одномерный массив представляет собой вектор, т.е. линейную таблицу с данными одного типа.

Пример одномерного массива

4	5	2	6
0	1	2	3

- значения элементов массива
- индексы

Объявление массива похоже на объявление переменной за тем исключением, что после указания типа ставятся квадратные скобки:

тип[] имя_массива;

Например, определим массив целых чисел:

int[] mas;

Массивы в C# являются динамическим типом данных, т.е. для использования массива нужно выделить память для нужного количества элементов массива. Делается это следующим образом:

mas = new int[4]; или так **int[] mas = new int[4];**

Для обращения к элементам массива используются индексы. Индекс представляет номер элемента в массиве, при этом нумерация начинается с нуля, поэтому индекс первого элемента будет равен 0. Например, если размер массива определен величиной 4, то в нем можно хранить 4 элементов с индексацией 0, 1, 2, 3.

Например, использование массива может выглядеть следующим образом:

```
int[] nums = new int[4];
nums[0] = 1;
nums[1] = 2;
nums[2] = 3;
nums[3] = 5;
Console.WriteLine(nums[3]); // 5
```

И так как у нас массив определен только для 4 элементов, то мы не можем обратиться, например, к шестому элементу: `nums[5] = 5;`. Если мы так попытаемся сделать, то мы получим исключение *IndexOutOfRangeException*.

Действия над элементами одномерного массива

Так как массив содержит множество значений одного типа, то обычно при выполнении различных действий с массивами, эти действия выполняются над каждым элементом массива по отдельности.

mas[2] = 5 - обращение ко второму элементу массива **Mas**.

Инициализация (присваивание начальных значений элементам массива)

```
//Инициализация элементов массива при описании
int[] mas = new int[4] { 3, 5, 3, 8 };

// Инициализация случайными значениями в диапазоне 0-200
int[] mas = new int[4];
Random rnd = new Random();
for (int i = 0; i < 4; i++) mas[i] = rnd.Next(200);
```

Создание массива (присваивание конкретных значений элементам массива)

```
//Ввод значений с клавиатуры
int[] mas = new int[4];
for (int i = 0; i < 4; i++) mas[i] = Convert.ToInt32(Console.ReadLine());

//Ввод значений с клавиатуры (информативный вариант)

int[] mas = new int[4];
Console.WriteLine("Введите 4 значения элементов массива");
for (int i = 0; i < 4; i++)
{
    Console.Write("Введите значение " + (i+1) + " элемента массива -");
    mas[i] = Convert.ToInt32(Console.ReadLine());
}
```

Вывод значений элементов массива

```
for (int i = 0; i < 4; i++) Console.Write(mas[i] + " ");
```

Копирование массивов

```
//Поэлементно
for (int i = 0; i < 4; i++) mas1[i] = mas[i];

//Целиком из массива в массив, если массивы имеют одинаковый размер и тип
mas1 = mas;
```

Поиск элементов в массиве

Требуется определить количество нулевых элементов в массиве.

```
int kol = 0;
for (int i = 0; i < 4; i++)
    if (mas[i] == 0) kol++;
```

Определение размера массива

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

Пример 1.

Найти минимальный элемент одномерного массива.

Методика решения задачи

4	5	2	6	4	1	3	8
---	---	---	---	---	---	---	---

Имеем массив, в котором мы легко определяем, что минимальный элемент это цифра 1. Теперь надо определить принцип, по которому было найдено это решение, и использовать его для решения задачи.

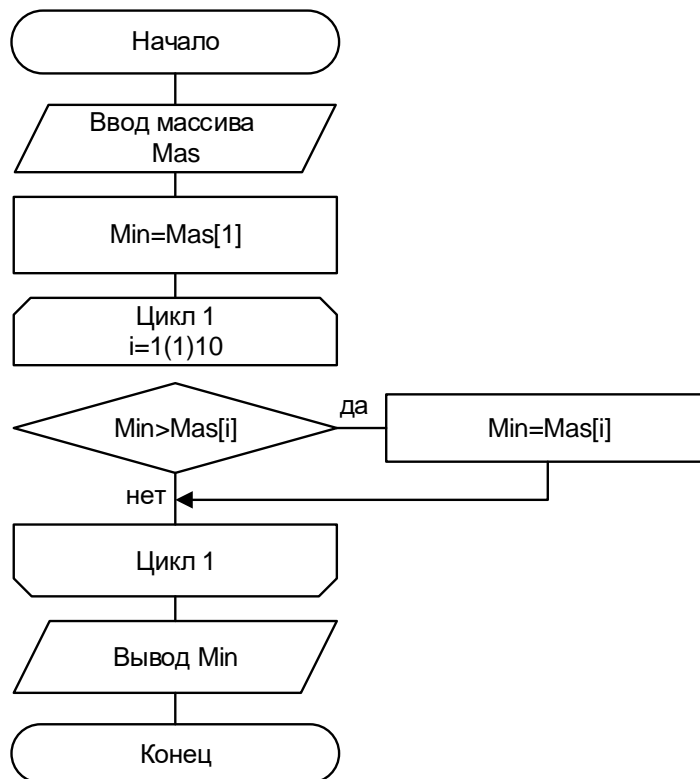
Явно наши действия выглядели так:

1. Мы запомнили первое значение в таблице
2. Далее сравнивали это значение с каждым элементом таблицы, запоминая новое минимальное значение

Распишем эти действия более подробно.

1. Запомнить первое значение в таблице, т.е. мы предполагаем, что это первое значение может быть минимальным **Min=Mas[0];**
2. Просматриваем каждый элемент таблицы. Для этого используем цикл с параметром **for (i=0;i<10;i++)**. Каждый элемент таблицы будет обозначаться **Mas[i]**, где **i** номер элемента таблицы, который просматривается в данный момент.
3. Сравниваем текущий элемент таблицы с запомненным минимальным значением. Если запомненное минимальное значение больше просматриваемого элемента таблицы, то мы нашли новое минимальное значение и должны его запомнить. **If (Min > Mas[i]) Min=Mas[i]**

Блок-схема алгоритма решения задачи:



Код программы

```
//31.03.2022 Сидоров С.С. ИСП-21
//Пример1. Найти минимальный элемент одномерного массива

const int n = 10; //Размер массива
int i, //Счетчик
    min; //Минимальное значение
int[] mas = new int[n]; //Массив

// Инициализация случайными значениями в диапазоне 0-200
Random rnd = new Random();
for (i = 0; i < n; i++) mas[i] = rnd.Next(200);

//Ищем минимальное значение
min = mas[0];
for (i = 1; i < mas.Length; i++) //Просматриваем каждый элемент массива
    if (min > mas[i]) min = mas[i]; //Сравниваем значения и находим мин

//Вывод массива
Console.WriteLine("Исходный массив");
for (i = 0; i < mas.Length; i++) Console.Write(mas[i] + " ");
Console.WriteLine();
Console.WriteLine("Минимальное значение= " + min);
Console.ReadKey(); //Пауза
```

Пример 2.

Вычислить сумму положительных элементов одномерного массива.

Методика решения задачи

4	-2	-2	6	-8	1	-3	8
---	----	----	---	----	---	----	---

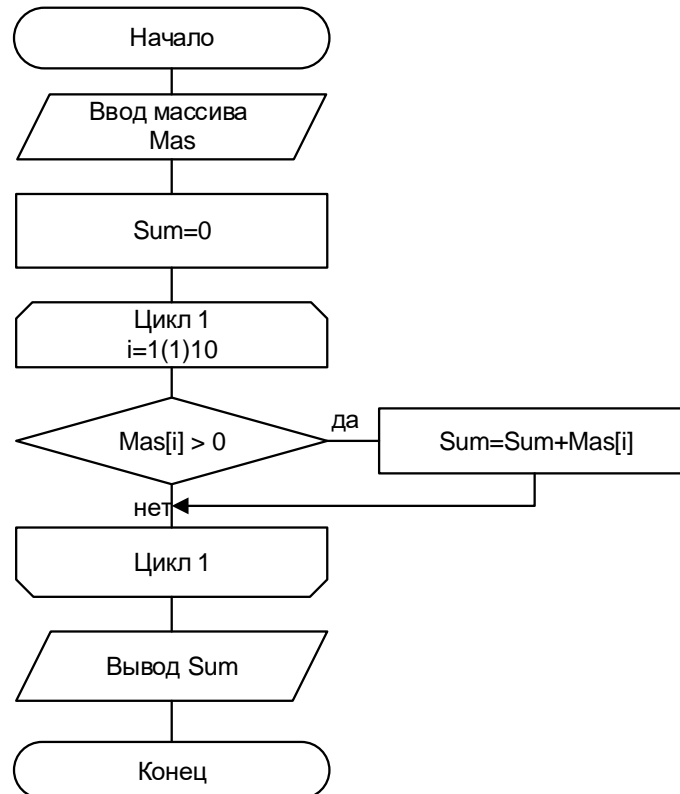
Имеем массив, в котором мы легко определяем, сумму $4+6+1+8=19$. Теперь надо определить принцип, по которому было найдено это решение, и использовать его для решения задачи.

Явно наши действия выглядели так:

1. Просматриваем каждый элемент таблицы.
2. Если текущий элемент таблицы больше нуля, то такие элементы суммируем.

Распишем эти действия более подробно.

1. Просматриваем каждый элемент таблицы. Для этого используем цикл с параметром **for (i=0;i<8;i++)**. Каждый элемент таблицы будет обозначаться **Mas[i]**, где **i** номер элемента таблицы, который просматривается в данный момент.
2. Если текущий элемент таблицы больше нуля, то такие элементы суммируем **If (Mas[i] > 0) Sum=Sum+Mas[i]**.

Блок-схема алгоритма решения задачи:**Код программы**

```

//31.03.2022 Сидоров С.С. ИСП-21

```

```

//Пример2. Вычислить сумму положительных элементов массива

```

```

const int n = 10; //Размер массива
int i, //Счетчик
    sum; //Сумма
int[] mas = new int[n]; //Массив

```

```

// Инициализация случайными значениями в диапазоне -100 .. 100
Random rnd = new Random();
for (i = 0; i < n; i++) mas[i] = rnd.Next(-100, 100);

```

```

//Ищем минимальное значение
sum = 0;
for (i = 0; i < n; i++) //Просматриваем каждый элемент массива
    if (mas[i] > 0) sum = sum + mas[i]; //Складываем положительные

```

```

//Вывод массива
Console.WriteLine("Исходный массив");
for (i = 0; i < n; i++) Console.Write(mas[i] + " ");
Console.WriteLine();
Console.WriteLine("Сумма положительных элементов= " + sum);
Console.ReadKey(); //Пауза

```

Практическая работа №9.

Обработка одномерных массивов

Задание. Составьте алгоритм и программу выполнения варианта задания для одномерных массивов и вывода исходных данных и результатов работы программы.

1. Дан массив 30 значений в диапазоне $[0;10]$ найти количество значений равных 5 и 7.
2. В одномерном массиве целых чисел найти максимальный среди элементов, являющихся четными, и минимальный среди элементов, кратных А.
3. Дан массив 30 значений в диапазоне $[-100;100]$ найти количество четных и нечетных.
4. Дан массив 30 значений в диапазоне $[-10;10]$ найти количество значений равных 5 и -5.
5. Дан массив 30 значений в диапазоне $[-100;100]$ найти количество положительных и отрицательных.
6. В массиве из 20 целых чисел найти наибольший элемент и поменять его местами с первым элементом.
7. Дан массив 30 значений в диапазоне $[-30;100]$ найти минимальное и максимальное значение и обменять их местами.
8. На пульте имеется n отверстий разных диаметров. Написать программу создания одномерного массива из номеров отверстий, в которые пройдет шарик радиусом г. Результирующий массив не должен содержать лишних элементов.
9. Составьте программу вычисления в массиве суммы всех чисел, кратных 3.
10. Массив содержит элементы, соответствующие росту школьников. Написать программу выбора школьников, рост каждого из которых меньше г.
11. Составьте программу вычисления среднего арифметического отрицательных элементов.
12. Заданы два массива. Проверить, сколько элементов первого массива превосходят соответствующие элементы второго массива.
13. В первом одномерном массиве хранятся затраты на производство продуктов, во втором — цены на эти продукты. Указать номер первого продукта, затраты на производство которого превышают цены.
14. Составьте блок-схему алгоритма и программу вычисления в массиве суммы всех чисел, кратных 7.
15. От «острова Буяна до царства славного Салтана» месяц пути. Капитан корабля записывает в вахтенный журнал расстояние, пройденное за день, в милях. Написать программу, определяющую, в какую десятидневку пройден больший путь.
16. Составьте программу вычисления в массиве максимального среди отрицательных элементов и его номера.
17. Проверить, имеется ли в одномерном массиве хотя бы один элемент, попадающий в интервал $[a; b]$
18. Проверить, имеется ли в одномерном массиве два соседних элемента, отличающихся друг от друга не более чем на два.
19. Проверить, состоит ли заданный одномерный массив только из нечетных чисел.

Задачи повышенной сложности

1. Дан массив $A[1..50]$, элементы которого отличны от нуля. Расположить их в таком порядке, чтобы первыми были все положительные элементы, а затем - все

отрицательные, причем порядок следования как положительных, так и отрицательных элементов должен сохраниться (при решении задачи новый массив не заводить!).

2. Упорядочить по возрастанию массив, содержащий 20 целых чисел.
3. Упорядочить по убыванию массив, содержащий 10 целых чисел.
4. Натуральные числа вводятся с клавиатуры до тех пока не будет введено число -
1. Написать программу, выводящую введенные цифры: сначала выводится наиболее часто встречающаяся цифра, в конце — наименее встречающаяся цифра.
5. В массиве заданы цены продажи доллара в банках города N. Определить, по скольким различным ценам продается здесь Доллар. Распечатать эти цены.

Двухмерные массивы

Двухмерный массив представляет собой список одномерных массивов, т.е. другими словами состоит из нескольких строк. Визуально двухмерный массив представляет собой таблицу (матрицу).

Пример двумерного массива

3	5	6	7
4	5	7	4
5	9	7	3

Объявление двухмерного массива осуществляется аналогично одномерному массиву:

тип[,] имя_массива;

Например, определим массив целых чисел:

int[,] matr;

Массивы в C# являются динамическим типом данных, т.е. для использования массива нужно выделить память для нужного количества элементов массива. Делается это следующим образом:

matr = new int[3, 4]; или так **int[,]** matr = new int[3, 4];

При выделении памяти массиву первая цифра задает количество строк, вторая цифра – количество столбцов.

Для обращения к элементам массива используются индексы. Для двухмерного массива первый индекс обозначит строку, второй индекс – столбец. Индексы задают ячейку массива.

Например, **matr[1, 2]** означает, что обращение осуществляется в ячейку с индексом строки 1 и индексом столбца 2, другими словами вторую строку и третий столбец.

Действия над элементами двухмерного массива

Так как массив содержит множество значений одного типа, то обычно при выполнении различных действий с массивами, эти действия выполняются над каждым элементом массива по отдельности.

Matr[2][5] - обращение к элементу находящемуся на пересечении второй строки и пятого столбца.

Инициализация (присваивание начальных значений элементам массива)

//Инициализация элементов массива при объявлении

```
int[,]
```

 matr = new int[3,4] {{2,4,3,5},{5,6,3,7},{7,5,3,5}};

// Инициализация случайными значениями в диапазоне 0-200

```
Random rnd = new Random();
```

```
int[,]
```

 matr = new int[3, 4];

```
for (int i = 0; i < 3; i++)
```

```
    for (int j = 0; j < 4; j++)
```

```
        matr[i, j] = rnd.Next(200);
```

Создание массива (присваивание конкретных значений элементам массива)

Для двумерного массива (упрощенный вариант): (используется вложенный оператор for):

```
for (int i = 0; i < 3; i++)
{
    for (int j = 0; j < 4; j++)
    {
        matr[i, j] = Convert.ToInt32(Console.ReadLine());
    }
}
```

Для двумерного массива (информативный вариант): (используется вложенный оператор for):

```
Console.WriteLine("Введите таблицу 3*4");
for (int i = 0; i < 3; i++)
{
    Console.WriteLine("Ввод строки " + i);
    for (int j = 0; j < 4; j++)
    {
        Console.Write("Введите значение " + (j+1) + " столбца -");
        matr[i, j] = Convert.ToInt32(Console.ReadLine());
    }
}
```

Вывод значений элементов массива

Для двумерного массива (упрощенный вариант): (используется вложенный оператор for):

```
for (int i = 0; i < 3; i++)
{
    for (int j = 0; j < 4; j++)
    {
        Console.Write(matr[i, j] + " ");
    }
}
```

Для двумерного массива (информативный вариант): (используется вложенный оператор for):

```
Console.WriteLine("Исходный массив");
for (int i = 0; i < 3; i++)
{
    for (int j = 0; j < 4; j++)
    {
        Console.Write("{0,5}", matr[i, j]);
    }
    Console.WriteLine();
}
```

Определение размера массива

```
int[,] matr = new int[3, 4];
for (int i = 0; i < matr.GetLength(0); i++)
    for (int j = 0; j < matr.GetLength(1); j++)
        matr[i, j] = rnd.Next(200);
```

Пример 3.

Вычислить номер столбца матрицы, состоящей из нулей и единиц, с максимальным количеством единиц.

Методика решения задачи

		j, m							
i, n		0	1	0	1	0	1	0	1
	1	1	0	0	1	0	1	0	0
	2	1	1	0	1	1	1	1	0
	3	1	0	0	0	1	0	1	0
	4	0	1	0	0	1	1	1	1

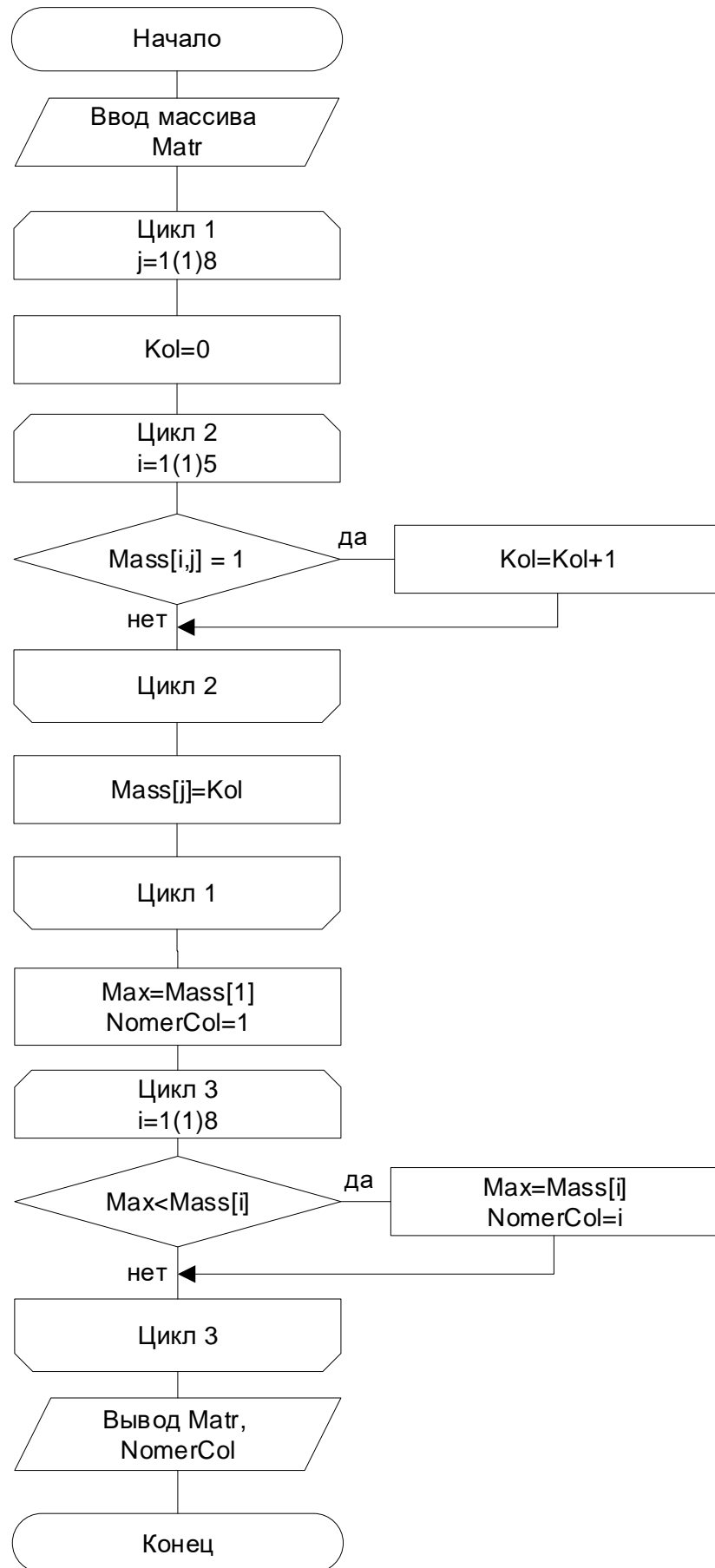
Имеем матрицу, в которой мы легко определяем, что 6 столбец имеет максимальное количество единиц. Теперь надо определить принцип, по которому было найдено это решение, и использовать его для решения задачи.

Явно наши действия выглядели так:

1. Просматриваем каждый столбец таблицы, подсчитываем в нем количество единиц и сравниваем полученные значения, находим и запоминаем номер столбца с максимальным количеством единиц.

Распишем эти действия более подробно.

1. Просматриваем каждый столбец таблицы
for (j=0;j<8;j++)
2. Подсчитываем в нем количество единиц
Kol=0;
for (i=0;i<5;i++) If (Matr[i,j] = 1) Kol++;
3. Записываем найденное количество в дополнительный одномерный массив.
Mass[j]=Kol;
4. Определяем в массиве mass номер столбца с максимальным количеством единиц. Выполнение смотри пример 1.
Max=Mass[0];
for (j=0;j<8;j++) If (Max < Mass[j]) Max=Mass[j]

Блок-схема алгоритма решения задачи:

Код программы

```
//31.03.2022 Сидоров С.С. ИСП-21
//Пример3. Вычислить номер столбца матрицы, состоящей из нулей
//и единиц, с максимальным количеством единиц

const int n = 5, m = 8; //Размер массива строки и столбцы
int i, j, //Индексы массива
    Kol, //Счетчик
    Max, //Максимальное значение
    Pos; //Номер столбца с max значением
int[] Mas = new int[m]; //Массив
int[,] Matr = new int[n, m]; //Матрица

// Инициализация случайными значениями в диапазоне 0-1
Random rnd = new Random();
for (i = 0; i < n; i++)
    for (j = 0; j < m; j++) Matr[i, j] = rnd.Next(2);

// Подсчитывание единиц в каждом столбце с сохранением результата
for (j = 0; j < m; j++)
{
    //Просматриваем каждый столбец
    Kol = 0; //Обнуляет количество для каждого столбца
    for (i = 0; i < n; i++) //Просматриваем каждую строку в столбце
        if (Matr[i, j] == 1) Kol++; //Если текущее значение 1, то счетчик +1
    Mas[j] = Kol; //записываем кол 1 во вспомогательный массив
}

//Ищем максимальное значение (max столбец)
Max = Mas[0]; Pos = 0; //Запоминаем первое значение как max
for (j = 1; j < m; j++) //Просматриваем каждый элемент массива
    if (Max < Mas[j])
    {
        //Сравниваем значения и находим max
        Max = Mas[j]; //Записываем новое max значение
        Pos = j; //Запоминаем номер столбца где нашли max значение
    }

//Вывод массива
Console.WriteLine("Исходный массив");
for (i = 0; i < n; i++)
{
    for (j = 0; j < m; j++)
        Console.Write("{0,5}", Matr[i, j]);
    Console.WriteLine();
}
Console.WriteLine();
Console.WriteLine("Столбец с max значением единиц= " + (Pos + 1));
Console.ReadKey(); //Пауза
```

Практическая работа №10.

Обработка двумерных массивов

Задание. Составьте алгоритм и программу обработки матрицы произвольного размера. Выведите на экран исходные данные в виде матрицы с заголовком и результаты работы программы.

1. Преобразовать матрицу, разделив каждый элемент матрицы на ее минимальный элемент
2. Вычислить количество нечетных элементов в каждой строке матрицы.
3. Сформировать одномерный массив из количества ненулевых элементов строк матрицы.
4. Записать элементы матрицы построчно в одномерный массив и найти его минимальный элемент.
5. Вычислить суммы элементов матрицы, лежащих выше, ниже и на главной диагонали.
6. Сформировать одномерный массив из максимальных элементов столбцов матрицы.
7. Вычислить количество повторений каждого элемента вектора. Сообщения о повторяющихся элементах должны выводиться только один раз.
8. Вычислить координаты максимального элемента главной диагонали матрицы.
9. Сформировать одномерный массив из сумм элементов строк матрицы.
10. Вычислить номер строки и номер столбца минимального по абсолютной величине элемента матрицы.
11. Вычислить номер столбца матрицы с максимальным произведением элементов.
12. Вычислить минимальный элемент матрицы, лежащий ниже главной диагонали.
13. Поменять местами k -ю строку и k -й столбец квадратной матрицы.
14. Вычислить номер строки матрицы, состоящей из нулей и единиц, с максимальным количеством нулей.
15. Сформировать одномерный массив из количества положительных элементов столбцов матрицы.
16. Вычислить количества нулевых элементов матрицы, лежащих выше, ниже и на главной диагонали.
17. Записать элементы матрицы по столбцам в одномерный массив и найти его максимальный элемент.
18. Сформировать одномерный массив из сумм элементов столбцов матрицы.
19. Поменять местами минимальный и максимальный элементы матрицы.
20. Поменять местами k -ю и n -ю строки матрицы.
21. Вычислить количество элементов каждой строки матрицы A , удовлетворяющих условию $0 \leq A_{ij} \leq 10$.
22. Вычислить номер столбца матрицы, состоящей из нулей и единиц, с максимальным количеством единиц.
23. Сформировать одномерный массив из координат нулевых элементов матрицы.
24. Определить, сколько элементов матрицы попадает в каждый из заданных интервалов: (n_1, k_1) ; (n_2, k_2) . Границы интервалов n_i, k_i выбрать произвольно, $i = 1, 2$.

Контрольные вопросы

1. Каким образом определяются переменные типа массив (одномерный и двумерный)?
2. Как осуществляется доступ к отдельному элементу одномерного и двумерного массива?
3. Каким образом выводятся элементы массива на экран?
4. Приведите пример фрагмента программы, который выводит на экран двумерный массив в виде матрицы.
5. Дайте определение массива.
6. Верно ли, что массив — это структурированный тип данных, состоящий из фиксированного числа элементов, имеющих один и тот же тип?
7. Где и как определяется общее число элементов массива?
8. Дайте определение индекса. Какие типы данных можно использовать при описании индекса?
9. Как осуществляется доступ к каждому элементу массива?
10. Может ли индекс быть выражением целого типа?
11. Может ли индекс быть выражением вещественного типа?
12. Какой массив называется одномерным?
13. Какой массив называется двумерным?
14. Верно ли, что одномерный массив соответствует понятию линейной таблицы (вектора)?
15. Верно ли, что двумерный массив соответствует понятию прямоугольной таблицы (матрице, набору векторов)?
16. Можно ли в описании массива использовать предварительно определенные константы?
17. В какой последовательности располагаются в памяти элементы одномерного массива?
18. Может ли оператор `cout` ввести весь массив `x` целиком?
19. Что такое размерность массива?
20. Где и как определяется размерность массива?
21. Как в описании массива использовать предварительно определенные константы? Приведите пример.

Сортировка массивов

Сортировкой называется процесс расположения элементов массива в порядке убывания (возрастания) из значений.

Пример:

Д	1	2	3	4	5	6
	2	4	1	5	7	9

Д	1	2	3	4	5	6
	9	7	5	4	2	1

Алгоритм выполнения сортировки называется методом сортировки. К наиболее распространенным методам относятся:

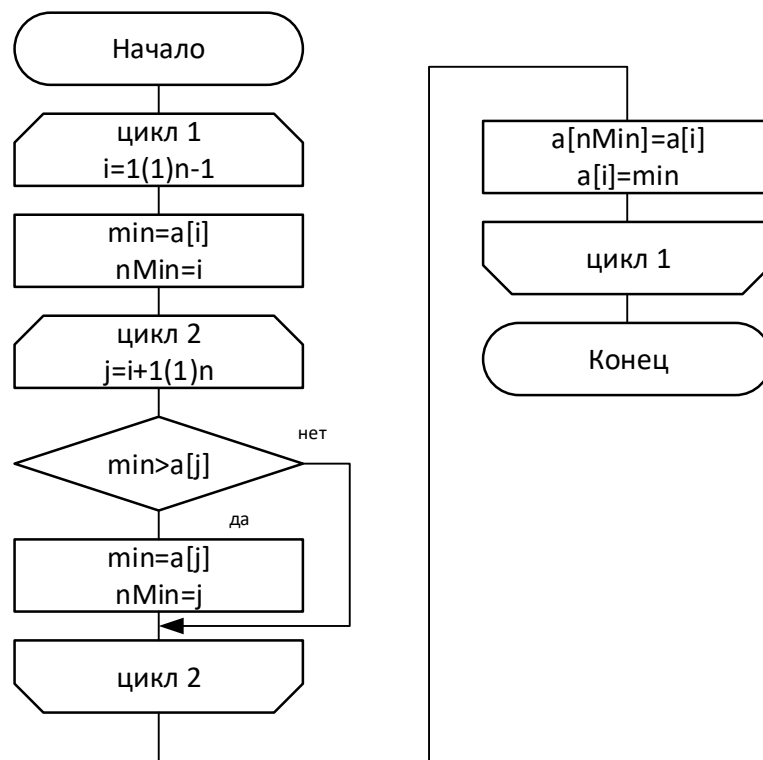
1. Простым выбором
2. Обменом (пузырьковый метод)

Алгоритм "Сортировка простым выбором".

Является одним из самых простых алгоритмов сортировки массива. Смысл в том, чтобы идти по массиву и каждый раз искать минимальный элемент массива, обменивая его с начальным элементом неотсортированной части массива. На небольших массивах может оказаться даже эффективнее, чем более сложные алгоритмы сортировки, но в любом случае проигрывает на больших массивах. Число обменов элементов по сравнению с "пузырьковым" алгоритмом N^2 , где N - число элементов массива.

Алгоритм:

1. Находим минимальный элемент в массиве
2. Меняем местами минимальный и первый элемент массива.
3. Опять ищем минимальный элемент в неотсортированной части массива
4. Меняем местами уже второй элемент массива и минимальный найденный, потому как первый элемент массива является отсортированной частью.
5. Ищем минимальные значения и меняем местами элементы, пока массив не будет отсортирован до конца.



Пример сортировки методом простого выбора.

3 7 15 5 9 11 2

Наименьший элемент 2 меняем местами с первым.

2 7 15 5 9 11 3

Наименьший элемент теперь 3. Меняем местами со вторым.

2 3 15 5 9 11 7

5 ставим на место третьего элемента, а третий на место пятёрки.

2 3 5 15 9 11 7

Далее

2 3 5 7 9 11 15

2 3 5 7 9 11 15

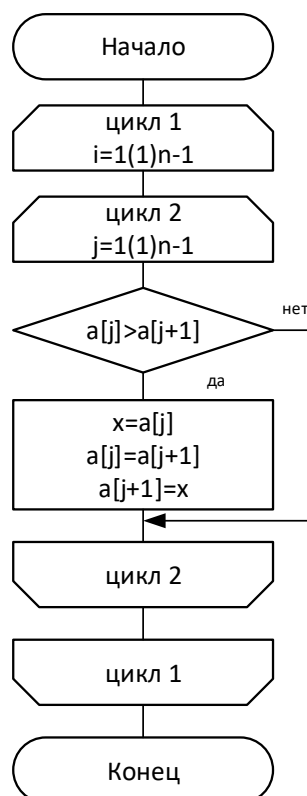
2 3 5 7 9 11 15

Алгоритм "Сортировка пузырьком".

Самый известный алгоритм, применяемый в учебных целях, для практического же применения является слишком медленным. Алгоритм лежит в основе более сложных алгоритмов: "Шейкерная сортировка", "Пирамидальная сортировка", "Быстрая сортировка". Примечательно то, что один из самых быстрых алгоритмов "Быстрый алгоритм" был разработан путем модернизации одного из самых худших алгоритмов "Сортировки пузырьком". Смысл алгоритма заключается в том, что самые "легкие" элементы массива как бы "всплывают", а самые "тяжелые" "тонут". Отсюда и название "Сортировка пузырьком"

Алгоритм:

1. Каждый элемент массива сравнивается с последующим и если элемент[i] > элемент[$i+1$] происходит замена. Таким образом самые "легкие" элементы "всплывают" - перемещаются к началу списка, а самые тяжелые "тонут" - перемещаются к концу.
2. Повторяем Шаг 1 $n-1$ раз, где n – количество элементов массива. Так же можно досрочно закончить сортировку, если перестановок не было.



Алгоритм "Гномья сортировка".

Алгоритм так странно назван благодаря голландскому ученому Дику Груну.

Гномья сортировка основана на технике, используемой обычным голландским садовым гномом (нидерл. *tuinkabouter*). Это метод, которым садовый гном сортирует линию цветочных горшков. По существу он смотрит на следующий и предыдущий садовые горшки: если они в правильном порядке, он шагает на один горшок вперёд, иначе он меняет их местами и шагает на один горшок назад. Граничные условия: если нет предыдущего горшка, он шагает вперёд; если нет следующего горшка, он закончил.

Вот собственно и все описание алгоритма "Гномья сортировка". Что интересно, алгоритм не содержит вложенных циклов, а сортирует весь массив за один проход.

Практическая работа №11.

Сортировка массивов

Задание. Составьте алгоритм и программу сортировки массива. Выведите на экран исходные данные и результаты работы программы.

1. Выполнить сортировку только четных элементов массива (нечетные элементы остаются на своих местах).
2. Выполнить сортировку элементов массива кратных 3 и 2 (остальные элементы остаются на своих местах).
3. Дан массив 15 целых чисел на отрезке $[-5;5]$. Упорядочить массив по возрастанию, удалив повторяющиеся элементы.
4. Выполнить сортировку элементов, записанных на нечетных местах. Использовать сортировку простым выбором.
5. Даны натуральные n, a_1, \dots, a_n ($n \geq 4$). Числа a_1, \dots, a_n — это измеренные в сотых долях секунды результаты n спортсменов в беге на 100 м. Составить команду из четырех лучших бегунов для участия в эстафете 4x100, т. е. указать одну из четверок натуральных чисел i, j, k, l такую, что сумма $a_i + a_j + a_k + a_l$ имеет наименьшее значение.
6. Дано n независимых случайных точек, с координатами $(x_i; y_i)$, равномерно распределенных в круге радиуса R с центром в начале координат. Напишите программу, располагающую точки в порядке возрастания расстояния от центра.
7. Выполнить сортировку элементов, записанных на нечетных местах. Использовать «Пузырьковую» сортировку.
8. Дан массив n целых чисел на отрезке $[-5;5]$. Отсортировать положительные элементы массива.
9. В матрице $n*m$ отсортировать столбцы в порядке возрастания.
10. Найти так называемую медиану ряда, т. е. такой его элемент, который больше любого из одной половины элементов и меньше любого из другой (если число элементов ряда четно, следует взять среднее значение из двух значений, обладающих указанным свойством).
11. Массив размером m , где m — натуральное число, заполнен случайным образом. Найдите в массиве моду. Модой называется элемент ряда, который встречается наиболее часто.
12. Дан двумерный квадратный массив размером $n*n$, заполненный числами в случайном порядке. Упорядочить значения в порядке возрастания при просмотре элементов в строках слева направо, а при просмотре строк — сверху вниз.
13. Выполнить сортировку элементов, записанных на четных местах. Использовать сортировку простым выбором.
14. В матрице $n*m$ отсортировать строки в порядке убывания.
15. Выполнить сортировку элементов, записанных на четных местах. Использовать «Пузырьковую» сортировку.