

Практическая работа №4

Работа с одномерными массивами

1. Цель работы:

Получение практических навыков при работе с одномерными массивами.

2. Краткие теоретические сведения

2.1. Массивы

Массив – это упорядоченная последовательность переменных одного типа. Каждому элементу массива отводится одна ячейка памяти. Элементы одного массива занимают последовательно расположенные ячейки памяти. Все элементы имеют одно имя – имя массива и отличаются индексами – порядковыми номерами в массиве. Количество элементов в массиве называется его размером.

Массив относится к ссылочным типам данных, то есть располагается в динамической области памяти, поэтому создание массива начинается с выделения памяти под его элементы с помощью операции `new`. Элементами массива могут быть величины как значимых, так и ссылочных типов (в том числе массивы). Массив значимых типов хранит значения, массив ссылочных типов — ссылки на элементы. Всем элементам при создании массива присваиваются значения по умолчанию: нули для значимых типов и `null` — для ссылочных.

Массивы определяются следующим образом:

```
int [] a= new int[100]; //массив из 100 элементов целого типа
```

Элементы массива всегда нумеруются с 0.

45	352	63		124	значения элементов массива
0	1	2	99	

Чтобы обратиться к элементу массива, надо указать имя массива и номер элемента в массиве (индекс):

`a[0]` – индекс задается как константа,
`a[55]` – индекс задается как константа,
`a[i]` – индекс задается как переменная,
`a[2*i]` – индекс задается как выражение.

2.4. Перебор элементов массива

- 1) Элементы массива можно обрабатывать по одному элементу, двигаясь от начала массива к его концу (или в обратном направлении):

```
for(int i=0;i<n;i++) <обработка a[i]>
```
- 2) Элементы массива можно обрабатывать по два элемента, двигаясь с обеих сторон массива к его середине:

```
int i=0, j=n-1;  
while (j>i){  
<обработка a[i] и a[j]>;  
i++;j++;}
```

- 3) Элементы массива можно обрабатывать по два элемента, двигаясь от начала к концу с шагом 1 (т. е. обрабатываются пары элементов $a[0]$ и $a[1]$, $a[1]$ и $a[2]$ и т. д.)

```
for (i=0; i<n-1; i++)
<обработка a[i] и a[i+1]>
```
- 4) Элементы массива можно обрабатывать по два элемента, двигаясь от начала к концу с шагом 2 (т. е. обрабатываются пары элементов $a[0]$ и $a[1]$, $a[2]$ и $a[3]$ и т. д.)

```
i=1;
while (i<n) {
<обработка a[i] и a[i+1]>
i:=i+2; }
```

2.5. Классы задач по обработке массивов

- 1) К задачам 1 класса относятся задачи, в которых выполняется однотипная обработка всех или указанных элементов массива. Решение таких задач сводится к установлению того, как обрабатывается каждый элемент массива или указанные элементы, затем подбирается подходящая схема перебора, в которую вставляются операторы обработки элементов массива. Примером такой задачи является нахождение среднего арифметического элементов массива.
- 2) К задачам 2 класса относятся задачи, в которых изменяется порядок следования элементов массива. Обмен элементов внутри массива выполняется с использованием вспомогательной переменной:

```
r=a[i]; a[i]=a[j]; a[j]=r; //обмен a[i] и a[j] элементов массива.
```
- 3) К задачам 3 класса относятся задачи, в которых выполняется обработка нескольких массивов или подмассивов одного массива. Массивы могут обрабатываться по одной схеме – синхронная обработка или по разным схемам – асинхронная обработка массивов.
- 4) К задачам 4 класса относятся задачи, в которых требуется отыскать первый элемент массива, совпадающий с заданным значением – поисковые задачи в массиве.

3. Постановка задачи

- 1) Сформировать массив из n элементов с помощью датчика случайных чисел (n задается пользователем с клавиатуры).
- 2) Распечатать массив.
- 3) Выполнить удаление указанных элементов из массива.
- 4) Выполнить добавление указанных элементов в массив.
- 5) Выполнить перестановку элементов в массиве.
- 6) Выполнить поиск указанных в массиве элементов и подсчитать количество сравнений, необходимых для поиска нужного элемента.
- 7) Выполнить сортировку массива указанным методом.
- 8) Выполнить поиск указанных элементов в отсортированном массиве и подсчитать количество сравнений, необходимых для поиска нужного элемента.

4. Варианты

Вариант	Удаление	Добавление	Перестановка	Поиск	Сортировка
1	Максимальный элемент	К элементов в начало массива	Перевернуть массив	Первый четный	Простой обмен
2	Минимальный элемент	К элементов в конец массива	Сдвинуть циклически на М элементов вправо	Первый отрицательный	Простой выбор
3	Элемент с заданным номером	Н элементов, начиная с номера К	Сдвинуть циклически на М элементов влево	Элемент с заданным ключом (значением)	Простое включение
4	Н элементов, начиная с номера К	Элемент с номером К	Поменять местами элементы с четными и нечетными номерами	Элемент равный среднему арифметическому элементов массива	Простой обмен
5	Все четные элементы	К элементов в начало массива	Четные элементы переставить в начало массива, нечетные - в конец	Первый четный	Простой выбор
6	Все элементы с четными индексами	К элементов в конец массива	Поменять местами минимальный и максимальный элементы	Первый отрицательный	Простое включение
7	Все нечетные элементы	Н элементов, начиная с номера К	Положительные элементы переставить в начало массива, отрицательные - в конец	Элемент с заданным ключом (значением)	Простой обмен
8	Все элементы с нечетными индексами	Элемент с номером К	Перевернуть массив	Элемент равный среднему арифметическому элементов массива	Простой выбор
9	Все элементы больше среднего арифметического элементов массива	К элементов в начало массива	Сдвинуть циклически на М элементов вправо	Первый четный	Простое включение

10	Максимальный элемент	К элементов в конец массива	Сдвинуть циклически на М элементов влево	Первый отрицательный	Простой обмен
11	Минимальный элемент	N элементов, начиная с номера К	Поменять местами элементы с четными и нечетными номерами	Элемент с заданным ключом (значением)	Простой выбор
12	Элемент с заданным номером	Элемент с номером К	Четные элементы переставить в начало массива, нечетные - в конец	Элемент равный среднему арифметическому у элементов массива	Простое включение
13	N элементов, начиная с номера К	К элементов в начало массива	Поменять местами минимальный и максимальный элементы	Первый четный	Простой обмен
14	Все четные элементы	К элементов в конец массива	Положительные элементы переставить в начало массива, отрицательные - в конец	Первый отрицательный	Простой выбор
15	Все элементы с четными индексами	N элементов, начиная с номера К	Перевернуть массив	Элемент с заданным ключом (значением)	Простое включение
16	Все нечетные элементы	Элемент с номером К	Сдвинуть циклически на М элементов вправо	Элемент равный среднему арифметическому у элементов массива	Простой обмен
17	Все элементы с нечетными индексами	К элементов в начало массива	Сдвинуть циклически на М элементов влево	Первый четный	Простой выбор
18	Все элементы больше среднего арифметического элементов массива	К элементов в конец массива	Поменять местами элементы с четными и нечетными номерами	Первый отрицательный	Простое включение
19	Максимальный элемент	N элементов, начиная с	Четные элементы переставить в	Элемент с заданным ключом	Простой обмен

		номера К	начало массива, нечетные - в конец	(значением)	
20	Минимальный элемент	Элемент с номером К	Поменять местами минимальный и максимальный элементы	Элемент равный среднему арифметическому элементов массива	Простой выбор
21	Элемент с заданным номером	К элементов в начало массива	Положительные элементы переставить в начало массива, отрицательные - в конец	Первый четный	Простое включение
22	Н элементов, начиная с номера К	К элементов в конец массива	Перевернуть массив	Первый отрицательный	Простой обмен
23	Все четные элементы	Н элементов, начиная с номера К	Сдвинуть циклически на М элементов вправо	Элемент с заданным ключом (значением)	Простой выбор
24	Все элементы с четными индексами	Элемент с номером К	Сдвинуть циклически на М элементов влево	Элемент равный среднему арифметическому элементов массива	Простое включение
25	Все нечетные элементы	К элементов в начало массива	Поменять местами элементы с четными и нечетными номерами	Первый четный	Простой обмен

5. Методические указания

1. Формирование массива осуществляется двумя способами:

а) С помощью датчика случайных чисел. Для этого используется класс Random.

```
Random a=new Random(0); //инициализация ДСЧ
```

```
. . . .
```

```
arr[i] = a.Next(0,100); //генерация элемента массива
```

б) Значения элементов вводятся с клавиатуры.

2. При выводе результатов элементы массива рекомендуется выводить в строку, разделяя их между собой пробелом.

```
for ( i = 0; i < n; i++) Console.Write(arr[i] + " ");
Console.WriteLine();
```

3. Диалог с пользователем осуществляется с помощью текстового меню. Для организации меню использовать оператор switch.

4. Класс Array для работы с массивом НЕ использовать.

6. Требования к программе

1. Реализация основных функций задачи (создание, обработка в соответствии с вариантом, вывод полученных результатов).
2. Дополнительные функции (проверка правильности вводимых данных и т.д.)
3. Стилизовое оформление программы.
4. Удобный интерфейс.
5. Использование исключений.
6. Использование возможностей языка программирования, изучаемых самостоятельно.

7. Содержание отчета

1. Описание этапа анализа (входные и выходные данные, исключительные ситуации).
2. Описание этапа проектирования (описание функций)
3. Листинг программы.
4. Тесты с проверкой полноты по критериям черного ящика.