# Практическая работа №4

# Работа с одномерными массивами

## 1. Цель работы:

Получение практических навыков обработки одномерных массивов

## 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 | индексы элементов массива |

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<j){  
   <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.

1. 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.
2. 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:

а) с помощью датчика случайных чисел (количество элементов массива задается пользователем с клавиатуры);

б) пользователь вводит элементы с клавиатуры (количество элементов массива задается пользователем с клавиатуры);

1. Распечатать массив.
2. Выполнить удаление указанных элементов из массива (задание 1).
3. Выполнить добавление указанных элементов в массив (задание 2).
4. Выполнить перестановку элементов в массиве (задание 3).
5. Выполнить поиск в массиве указанных элементов и подсчитать количество сравнений, необходимых для поиска нужного элемента.
6. Выполнить сортировку массива указанным методом.
7. Выполнить поиск элемента, который вводит пользователь с клавиатуры, в отсортированном массиве (бинарный поиск) и подсчитать количество сравнений, необходимых для поиска нужного элемента.

## 4. Варианты

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Вариант | Задание 1 | Задание 2 | Задание 3 | 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 M элементов вправо | Первый отрицательный | Простой выбор |
| 3 | Элемент с заданным номером | N элементов, начиная с номера К | Сдвинуть циклически на M элементов влево | Элемент с заданным ключом (значением) | Простое включение |
| 4 | N элементов, начиная с номера K | 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 номерами | К элементов в конец массива | Поменять местами минимальный и максимальный элементы | Первый отрицательный | Простое включение |
| 7 | Все нечетные элементы | N 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 номерами | 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 M элементов вправо | Первый четный | Простое включение |
| 10 | Максимальный  элемент | К элементов в конец массива | Сдвинуть циклически на M 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 элементов массива | Простое включение |
| 13 | N элементов, начиная с номера K | 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 номерами | 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 M элементов вправо | Элемент равный среднему арифметическому элементов массива | Простой обмен |
| 17 | Все элементы с нечетными номерами | К элементов в начало массива | Сдвинуть циклически на M 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 номера К | Четные элементы переставить в начало массива, нечетные - в конец | Элемент с заданным ключом (значением) | Простой обмен |
| 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 | N элементов, начиная с номера K | К элементов в конец массива | Перевернуть массив | Первый отрицательный | Простой обмен |
| 23 | Все четные элементы | N элементов, начиная с номера К | Сдвинуть циклически на M элементов вправо | Элемент с заданным ключом (значением) | Простой выбор |
| 24 | Все элементы с четными номерами | Элемент с номером К | Сдвинуть циклически на M 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(-100,100);//генерация элемента массива

б) Значения элементов вводятся с клавиатуры.

2. Для пользователя элементы нумеруются с 1.

3. При выводе результатов элементы массива нужно выводить в строчку, разделяя их между собой пробелом.

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

Console.WriteLine();

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

5. 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. Стилевое оформление программы.
  4. Удобный интерфейс в виде двухуровневого текстового меню с выделением задач и подзадач.
  5. Обработка стандартных исключений.
  6. Использование возможностей языка программирования, изучаемых самостоятельно (дополнительные баллы).

## Примерные вопросы для защиты лабораторной работы

* 1. Что такое массив?
  2. Как выделить память под массив?
  3. Можно ли изменить количество выделенной под массив памяти в процессе выполнения программы?
  4. Что такое индекс?
  5. Как выполняется нумерация элементов в массиве?
  6. Каким образом можно перебирать элементы массива?
  7. Как можно заполнить массив данными?
  8. Какие классы задач существуют? Привести примеры задач разных классов. Оценить их сложность.
  9. Какой цикл лучше всего использовать для перебора массива по одному элементу?
  10. Как поменять местами пары элементов в массиве? Оценить сложность задачи.
  11. Как перевернуть массив? Оценить сложность задачи.
  12. Как выполнить удаление элементов из массива? Оценить сложность задачи.
  13. Как выполнить добавление элементов в массив? Оценить сложность задачи.
  14. Как выполнить сортировку массива (простой метод сортировки)? Оценить сложность задачи.
  15. Чем отличается линейный поиск от двоичного? Оценить сложность задачи.

## 8. Содержание отчета

* 1. Постановка задач для конкретного варианта.
  2. Анализ исходных данных и классов исходных данных для каждой задачи.
  3. Алгоритм решения каждой задачи в виде блок-схемы (удаление, добавление, перестановка, поиск, сортировка).
  4. Программа на языке C#.
  5. Тесты (с проверкой достаточности по критериям черного ящика) для каждой задачи.

## Критерии оценки выполнения программы

|  |  |  |
| --- | --- | --- |
| № | Критерий | Баллы |
| 1 | Выполнено формирование массива двумя способами и вывод сформированного массива | 1 балл |
| 2. | Выполнены удаление, добавление, перестановка элементов массива | 2 балла |
| 3 | Выполнены сортировка (простой метод) и поиск элемента массива | 2 балла |
| 4 | В программе имеется текстовое меню, проверка правильности ввода данных и возможность исправления ошибок при вводе данных | 1 балл |
| 5 | Даны полные ответы на теоретические вопросы (п.5) | 1 балл |
| 6 | Написанный код соответствует стиль-гайду | 1 балл |
| 7 | Дополнительные баллы можно получить за реализацию и объяснение одного из быстрых методов сортировки (сортировка Хоара, Шелла, шейкер-сортировка и т.п) | 1 балл |