# Отчет по практической работе №5 «Одномерные массивы»

## Задание

1. Сформировать массив из n элементов (n задается пользователем с клавиатуры) двумя способами:
   1. с помощью датчика случайных чисел,
   2. элементы вводятся пользователем с клавиатуры.

Способ ввода выбирается пользователем с помощью меню.

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

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Вариант** | **Удаление** | **Добавление** | **Перестановка** | **Поиск** | **Сортировка** |
| ХХ | K элементов, начиная с номера N | К элементов , начиная с номера N | Перевернуть массив | Первый элемента с заданным ключом | Простой обмен |

## Анализ задачи

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№** | **Описание функции** | **Входные данные** | **Классы входных данных** | **Выходные данные** |
| **Основные функции** | | | | |
| 1. | Формирование массива | * Количество элементов в массиве | Целое число> 0 | Массив из целых чисел |
| 1.1. | Формирование массива с помощью ДСЧ | Целое число <=0 | * Сообщение об ошибке * null |
| Символ |
| Вещественное число |
| 1.2. | Формирование массива с клавиатуры | * Число | * Целое число |  |
| * Символ, вещественное число | * Сообщение об ошибке |
| 2 | Печать массива | * Массив * Размер массива | Массив не пустой | Массив напечатан |
| Массив пустой | Сообщение об ошибке |
| 3 | Удаление элементов из массива | * Массив * Размер массива, * Номер, с которого удаляем * Количество удаленных элементов | * Массив не пустой * Размер массива >0 * Номер, с которого удаляем >0 и <Размера массива * Размер массива >Номер, с которого удаляем + Количество удаленных элементов | * Измененный массив * Измененный размер массива |
| Массив пустой | * Сообщение об ошибке, * массив не меняется, * Размер массива не меняется |
| Массив не пустой, из него удаляются все элементы | * Сообщение о том, что массив пустой, * null * Размер массива=0 |
| * Номер, с которого удаляем – целое число <=0 * Номер, с которого удаляем > Размер массива * Номер, с которого удаляем - символ, вещественное число | * Сообщение об ошибке * Массив не меняется * Размер массива не меняется |
| * Количество удаленных элементов <=0 * Размер массива <Номер, с которого удаляем + Количество удаленных элементов * Количество удаленных элементов - символ, вещественное число | * Сообщение об ошибке * Массив не меняется * Размер массива не меняется |
| 4 | Добавление в массив | * Массив * Размер массива * Номер, с которого добавляем * Количество добавленных элементов | * Массив пустой, * номер, с которого добавляем =1, * сколько добавляем >0 | * Измененный массив * Измененный размер массива |
| * Массив не пустой, * номер, с которого добавляем =>1 и <= Размер массива, * Количество добавленных элементов >0 | * Измененный массив * Измененный размер массива |
| * номер, с которого добавляем <1, * номер, с которого добавляем – символ, вещественное число | * Сообщение об ошибке * Массив не меняется * Размер массива не меняется |
| * Количество добавленных элементов >0 * Количество добавленных элементов <= 0 * Количество добавленных элементов – символ, вещественное число |
| 5 | Перестановка | * Массив * Размер массива | Массив пустой | * Сообщение об ошибке * Массив не меняется * Размер массива не меняется |
| Массив не пустой | * Массив меняется * Размер массива не меняется |
| 6 | Сортировка | * Массив * Размер массива | Массив пустой | * Сообщение об ошибке * Массив не меняется * Размер массива не меняется |
| Массив отсортирован | * Сообщение об ошибке * Массив не меняется * Размер массива не меняется |
| Массив не отсортирован | * Массив меняется * Размер массива не меняется |
| 7.1 | Поиск в не отсортированном массиве | * Массив * Размер массива * Элемент для поиска | Массив пустой | Сообщение об ошибке |
| Массив не пустой, нужный элемент есть | * Номер элемента>0 * Количество сравнений>0 |
| Массив не пустой, нужного элемента нет | * Номер элемента<0 * Количество сравнений |
| 7.2 | Поиск в отсортированном массиве | * Массив * Размер массива * Элемент для поиска | Массив пустой | Сообщение об ошибке |
| Массив неотсортирован | Сообщение об ошибке |
| Массив не пустой, нужный элемент есть | * Номер элемента>0 * Количество сравнений |
| Массив не пустой, нужного элемента нет | * Номер элемента<0 * Количество сравнений |
| **Вспомогательные функции** | | | | |
| 8 | Ввод целого числа с клавиатуры | * Строка, которую вводит пользователь * Левая граница диапазона, * Правая граница диапазона | Строка – представляет собой целое число в диапазоне от левой границы до правой границы-1 | Число |
| Не является целым числом | Сообщение об ошибке |
| Очень большое целое число (переполнение) |
| Выходит за пределы диапазона |
| 9 | Проверка массива на пустоту | Массив | Пустой массив | false |
| Не пустой массив | true |
| 10 | Печать диалога для ввода команд пользователя |  |  |  |

## Проектирование

**Данные для проектирования берутся из таблицы для анализа!**

### Проектирование функций

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Функция** | **Входные данные** | **Выходные данные** |
| **Основные функции** | | | |
| 1. | MakeArray  (формирование массива) | int sizeOfArray - количество элементов в массиве | int[] arr – массив  Сообщение об ошибке |
| 1.1. | MakeRandomArray (формирование массива с помощью ДСЧ) |
| 1.2. | ReadArray (формирование массива с помощью ДСЧ) |
| 2 | Печать массива (PrintArray) | int sizeOfArray - количество элементов в массиве  int[] arr - массив | Печать массива  Сообщение об ошибке |
| 3 | DeleteFromArray (удаление из массива) | int[] arr - массив,  int sizeOfArray – размер массива  int firstElem – номер, с которого удаляем  int howMatchElem  - сколько удаляем | int[] arr - массив  int sizeOfArray – размер массива  Сообщение об ошибке |
| 4 | AddToArray (добавление в массив) | int[] arr - массив,  int sizeOfArray – размер массива  int firstElem – номер, с которого добавляем  int howMatchElem  - сколько добавляем | int[] arr - массив  int sizeOfArray – размер массива  Сообщение об ошибке |
| 5 | Reverse  (перестановка) | int[] arr - массив,  int sizeOfArray – размер массива | int[] arr - массив  int sizeOfArray – размер массива  Сообщение об ошибке |
| 6 | Сортировка (Sort) | int[] arr - массив,  int sizeOfArray – размер массива | int[] arr - массивё  int sizeOfArray – размер массива  Сообщение об ошибке |
| 7.1 | FindElem (поиск в неотсортированном массиве) | int[] arr - массив,  int sizeOfArray – размер массива | int number – номер элемента,  int count – количество сравнений  Сообщение об ошибке |
| 7.2 | Поиск в отсортированном массиве (BinarySearch) | int[] arr - массив,  int sizeOfArray – размер массива | int number – номер элемента,  int count – количество сравнений  Сообщение об ошибке |
| **Вспомогательные функции** | | | |
| 8 | ReadIntNumber (ввод целого числа с клавиатуры) | string stringForUser – строка, которую вводит пользователь,  int left – левая граница разрешенного диапазона,  int right – правая граница разрешенного диапазона | Целое число  Сообщение об ошибке |
| 9 | ExistArray (проверка массива на пустоту) | int[] arr - массив | False/true |
| 10 | MakeMenu (диалог для ввода команд пользователя) |  |  |

### Блок-схема функции Main()

Начало

Init (out int[]arr, out int sizeOfArr)

Run (out int[]arr, out int sizeOfArr)

Конец

### Укрупненная блок-схема функции Run

Run (arr, sizeOfArray)

ввод (check)

PrintMenu()

check==1

arr=MakeArray(sizeOfArray)

ввод (sizeOfArray)

check==2

PrintArray(arr, sizeOfArray)

check==3

DeleteFromArray(ref arr, ref sizeOfArray)

1

3

check==4

AddToArray(ref arr, ref sizeOfArray)

check==5

index=

BinarySearch(arr, sizeOfArray, number, out count)

Exist(arr)

вывод («Массив пустой»)

ввод(number)

sortOk

index=

FindElem(arr, sizeOfArray, number, out count)

index<0

вывод («Элемент не найден»)

вывод (index)

1

2

check==6

Sort(arr, sizeOfArray)

sortOk

вывод («Массив отсортирован»)

check==7

Reverse (arr, sizeOfArray)

check<8

3

2

конец

### Программа

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace HsePract\_5

{

class Program

{

//ограничения для размера массива

const int MIN\_LENGTH = 1;

const int MAX\_LENGTH = 100;

//ограничения для элеменов массива

const int MIN\_VALUE = -100;

const int MAX\_VALUE = 100;

// функция для ввода целого числа с проверкой границ и правильности ввода

static int ReadIntNumber(string stringForUser, int left, int right)

{

bool okInput = false;

int number = MIN\_VALUE;

do

{

Console.WriteLine(stringForUser);

try

{

string buf = Console.ReadLine();

number = Convert.ToInt32(buf);// могут возникнуть исключения FormatException, OverflowException

if (number >= left && number < right) okInput = true;

else

{

Console.WriteLine("Не верно введено число!");

okInput = false;

}

}

catch (FormatException)

{

Console.WriteLine("Не верно введено число!");

okInput = false;

}

catch (OverflowException)

{

Console.WriteLine("Не верно введено число!");

okInput = false;

}

} while (!okInput);

return number;

}

static void PrintMenu()

{

Console.WriteLine("1. Формирование массива");

Console.WriteLine("2. Печать массива");

Console.WriteLine("3. Удаление элементов из массива");

Console.WriteLine("4. Добавление элементов в массив");

Console.WriteLine("5. Поиск элемента в массиве");

Console.WriteLine("6. Сортировка массива");

Console.WriteLine("7. Перестановка элементов в массиве");

Console.WriteLine("8. Выход ");

}

static void Run(int[] arr, int sizeOfArray) // вызов функций

{

int check = 0; //номер пункта меню

bool sortOk = false; //проверка отсортирован массив или нет

do

{

PrintMenu();

check = ReadIntNumber("", 1, 9); //9 пунктов меню

switch (check)

{

case 1: //формирование

{

sizeOfArray = ReadIntNumber("Введите размер массива:", MIN\_LENGTH, MAX\_LENGTH);

arr = MakeArray(sizeOfArray);

sortOk = false;//массив не отсортирован

break;

}

case 2: PrintArray(arr, sizeOfArray); break; //печать

case 3: DeleteFromArray(ref arr, ref sizeOfArray); //удаление из массива

sortOk = false;//массив изменился

break;

case 4: AddToArray(ref arr, ref sizeOfArray); //добавление в массив

sortOk = false;//массив изменился

break;

case 5:

int count = 0, //количество сравнений

index = -1;//номер найденного элемента

if (ExistArray(arr, ref sizeOfArray) )//массив не пустой

{

int number = ReadIntNumber("Введите число для поиска:", MIN\_VALUE, MAX\_VALUE);

if (!sortOk) //массив не отсортирован

{

index = FindElem(arr, sizeOfArray, number, out count) + 1;//+1 т.к. нумерация для пользователя с 1

if (index < 0) Console.WriteLine("Элемент {0} в массиве не найден, число сравнений: {1},\nмассив не отсортирован, выполнен линейный поиск", number, count);

else Console.WriteLine("Элемент {0} находится на позиции {1}, число сравнений: {2},\nмассив не отсортирован, выполнен линейный поиск", number, index, count);

}

else

{

index = BinarySearch(arr, sizeOfArray, number, out count) + 1;//+1 т.к. нумерация для пользователя с 1

if (index < 0) Console.WriteLine("Элемент {0} в массиве не найден, число сравнений: {1},\nмассив отсортирован, выполнен бинарный поиск", number, count);

else Console.WriteLine("Элемент {0} находится на позиции {1}, число сравнений: {2},\n массив отсортирован, выполнен бинарный поиск", number, index, count);

}

}

else Console.WriteLine("Массив пустой ");

break;

case 6: if (!sortOk) //не отсортирован

Sort(arr, sizeOfArray);

sortOk = true;//изменить флаг сортировки

break;

case 7: Reverse(arr, sizeOfArray);

sortOk = false;//массив изменился

break;

}

} while (check < 8);

}

static bool ExistArray(int[] arr, ref int sizeOfArray)

{

if (arr == null)

{

if (sizeOfArray!=0) sizeOfArray=0;

return false;

}

return true;

}

static void PrintFormMenu()

{

Console.WriteLine("1. Формирование ДСЧ");

Console.WriteLine("2. Ввод с клавиатуры");

}

static int[] MakeArray(int sizeOfArray) //для формирования массива двумя способами

{

int[] arr = null;

int check = 0;

PrintFormMenu();

check = ReadIntNumber("", 1, 3);

switch (check)

{

case 1: arr = MakeRandomArray(sizeOfArray);

break;

case 2: arr = ReadArray(sizeOfArray);

break;

}

Console.WriteLine("Массив сформирован");

return arr;

}

static int[] ReadArray(int sizeOfArray)//для ввода элементов массива указанного размера с клавиатуры с проверкой правильности ввода

{

int[] arr = new int[sizeOfArray]; //выделить память под массив

for (int i = 0; i < sizeOfArray; i++)

{

arr[i] = ReadIntNumber("Введите элемент массива из диапазона от -100 до 100", MIN\_VALUE, MAX\_VALUE);

}

return arr;

}

static int[] MakeRandomArray(int sizeOfArray)//для формирования элементов массива указанного размера с помощью ДСЧ

{

Random rnd = new Random();

int[] arr = new int[sizeOfArray];//выделить память под массив

for (int i = 0; i < sizeOfArray; i++)

{

arr[i] = rnd.Next(MIN\_VALUE, MAX\_VALUE);

}

return arr;

}

static void PrintArray(int[] arr, int sizeOfArray)

{

if (ExistArray(arr, ref sizeOfArray))

{

foreach (int elem in arr)

Console.Write(elem + " ");

Console.WriteLine();

}

else Console.WriteLine("Массив пустой!");

}

static void DeleteFromArray(ref int[] arr, ref int sizeOfArray)

{

if (!ExistArray(arr, ref sizeOfArray)) //массив пустой

{

Console.WriteLine("Удаление невозможно!");

return;//массив не меняется, можно завершить функцию

}

int firstElem = ReadIntNumber("Введите номер элемента,с котрого элемента нужно удалять элементы из массива:", 0, MAX\_LENGTH);

//проверяем корректность данных для удаления

if (firstElem <= 0 || firstElem > sizeOfArray)

{

Console.WriteLine("Не правильно задан номер, с которого нужно удалять элементы массива!");

return;//массив не меняется, можно завершить функцию

};

int howMatchElem = ReadIntNumber("Введите сколько элементов нужно удалять из массива:", 0, MAX\_LENGTH);

if (firstElem + howMatchElem > sizeOfArray + 1)

{

Console.WriteLine("Не правильно задано количество удаляемых элементов массива!");

return;//массив не меняется, можно завершить функцию

};

Console.WriteLine("Удаляем {0} элементов, начиная с {1} ", howMatchElem, firstElem);

int newSize = sizeOfArray - howMatchElem;

if (newSize == 0) //из массива удаляются все элементы

{

arr = null;

sizeOfArray = 0;

return;

}

//из массива удаляются не все элементы

int[] temp = new int[sizeOfArray - howMatchElem];//для переноса элементов, которые останутся после удаления

int j = 0, i = 0;

for (; i < sizeOfArray; i++)

if (i < firstElem - 1 || i >= (firstElem + howMatchElem) - 1)//номера элементов для пользователя считаются с 1, а не с 0

{

temp[j] = arr[i];

j++;

}

sizeOfArray = sizeOfArray - howMatchElem;

arr = temp;

}

static void AddToArray(ref int[] arr, ref int sizeOfArray)

{

int firstElem = ReadIntNumber("Введите номер, с которого нужно добавлять элементы в массив:", 0, MAX\_LENGTH);

//проверяем корректность данных для добавления

if (firstElem <= 0 || firstElem > sizeOfArray + 1)// +1 т.к. можно добавить элементы после последнего элемента массива

{

Console.WriteLine("Не правильно задан номер, с которого нужно добавлять элементы массива!");

return;//массив не меняется, можно завершить функцию

};

int howMatchElem = ReadIntNumber("Введите сколько элементов нужно добавлять в массив:", 0, MAX\_LENGTH);

if (sizeOfArray + howMatchElem > MAX\_LENGTH)

{

Console.WriteLine("Не правильно задано количество добавляемых элементов массива!");

return; //массив не меняется, можно завершить функцию

};

int[] temp;

if (ExistArray(arr, ref sizeOfArray)) temp = new int[sizeOfArray + howMatchElem];//для переноса элементов, которые будут в массиве после добавления

else temp = new int[howMatchElem];// в новом массиве будут только новые элементы

int[] elemForAdding = MakeArray(howMatchElem);//для формирования массива с элементами для добавления

if (!ExistArray(elemForAdding,ref howMatchElem)) //пользователь пропустил формирование массива и вернулся

{

Console.WriteLine("Элементы для добавления не сформированы!");

return; //массив не меняется, можно завершить функцию

}

int j = 0, //счетчик в новом массиве

i1 = 0, //счетчик в исходном массиве

i2 = 0;//счетчики в массиве с элементами для добавления

for (; i1 < firstElem - 1; i1++) //номера элементов для пользователя считаются с 1, а не с 0

{

temp[j] = arr[i1]; //перенос элементов из старого массива в новый до элемента, с которого начинается добавление

j++;

}

for (; i2 < howMatchElem; i2++)

{

temp[j] = elemForAdding[i2];//перенос добавляемых элементов в новый массив

j++;

}

for (i1 = firstElem - 1; i1 < sizeOfArray; i1++) //номера элементов считаются с 1, а не с 0 (как в машинном массиве)

{

temp[j] = arr[i1];//перенос оставшихся элементов из старого массива в новый

j++;

}

Console.WriteLine("Элементы в массив добавлены");

sizeOfArray = sizeOfArray + howMatchElem; //изменить количество элеменов в новом массиве

arr = temp;

}

static int FindElem(int[] arr, int sizeOfArray, int number, out int count)

{

int index = -1; //начальное значение != индексу в массиве

count = 0;

for (int i = 0; i < sizeOfArray; i++)

{

count++;

if (arr[i] == number)

{

index = i; break;

}

}

Console.WriteLine("Поиск элемента выполнен");

return index; //отрицательное значение, если число не найдено

}

static int BinarySearch(int[] arr, int sizeOfArray, int number, out int count)

{

int left = 0, right = sizeOfArray - 1, middle = 0;

count = 0;

do

{

middle = (left + right) / 2;

if (arr[middle] < number) left = middle + 1;

else right = middle;

count++;

} while (left != right);

Console.WriteLine("Поиск элемента выполнен");

if (number == arr[left]) return left;

else return -1;

}

static void Sort(int[] arr, int sizeOfArray) //метод простого обмена

{

if (ExistArray(arr, ref sizeOfArray))

{

for (int i = 1; i < sizeOfArray; i++)

for (int j = sizeOfArray - 1; j >= i; j--)

if (arr[j] < arr[j - 1])

{

int temp = arr[j];

arr[j] = arr[j - 1];

arr[j - 1] = temp;

}

Console.WriteLine("Сортировка выполнена");

}

else

Console.WriteLine("Сортировка не выполнена, массив пустой");

}

static void Reverse(int[] arr, int sizeOfArray)

{

if (ExistArray(arr, ref sizeOfArray))

{

int left = 0, right = sizeOfArray - 1;

while (left < right)

{

int temp = arr[left];

arr[left] = arr[right];

arr[right] = temp;

left++; right--;

}

Console.WriteLine("Перестановка элементв выполнена");

}

else

Console.WriteLine("Перестановка элементв не выполнена, массив пустой");

}

static void Init(out int sizeOfArray, out int[] arr)

{

sizeOfArray = 0;

arr = null;

}

static void Main(string[] args)

{

int sizeOfArray;

int[] arr;

Init(out sizeOfArray, out arr); //инициализация

Run(arr, sizeOfArray);//выполнение функций

}

}

}

### Тестирование

**Данные для тестов берутся из таблицы для анализа!**

1. Формирование массива

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№ теста** | **Входные данные** | **Ожидаемый результат** | **Реальный результат** | **Примечание** |
| 1 | 5 | 5 чисел | -90 0 27 54 -67 | Ввод размера массива |
| 2 | 0 | Сообщение об ошибке | Неверно введено число |
| 3 | -3 | Сообщение об ошибке | Неверно введено число |
| 4 | 1,34 | Сообщение об ошибке | Неверно введено число |
| 5 | q | Сообщение об ошибке | Неверно введено число |
| 6 | 1 | 1 число | 10 |
| 7 | 1 | 1 | 1 | Ввод элементов массива с клавиатуры |
| 8 | 1111111111111 | Сообщение об ошибке | Неверно введено число |
| 9 | 100 | Сообщение об ошибке | Неверно введено число |
| 10 | -100 | -100 | -100 |
| 11 | 0 | 0 | 0 |

1. Печать массива

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№ теста** | **Входные данные** | **Ожидаемый результат** | **Реальный результат** | **Примечание** |
| 1 | пустой массив | Сообщение об ошибке | Массив пустой |  |
| 2 | -1 53 13 | -1 53 13 | -1 53 13 |  |

1. Удаление из массива

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№ теста** | **Входные данные** | **Ожидаемый результат** | **Реальный результат** | **Примечание** |
| 1 | -1, 53, -13, удалить с 1, удалить 2 элемента | -13 | -13 | Массив не пустой, данные для удаления верные |
| 2 | -1, 53, -13, 15, -7, удалить с 2, удалить 2 элемента | -1, 15,-7 | -1, 15,-8 |
| 3 | -1, 53, -13, удалить с 5 | Сообщение об ошибке | Не правильно задан номер, с которого нужно удалять элементы | Массив не пустой, данные для удаления не верные |
| 4 | -1, 53, -13 удалить с 1, удалить 10 элементов | Сообщение об ошибке | Не правильно задано количество удаляемых элементов |
| 5 | -1, 53, -13 удалить с 1,5 | Сообщение об ошибке | не верно введено число |
| 6 | -1, 53, -13 удалить с q | Сообщение об ошибке | не верно введено число |
| 7 | -1, 53, -13 удалить с 1, удалить 1,5 элементов | Сообщение об ошибке | не верно введено число |
| 8 | -1, 53, -13 удалить с 1, удалить q элементов | Сообщение об ошибке | не верно введено число |
| 9 | -1, 53, -13 удалить с -1 | Сообщение об ошибке | не верно введено число |
| 10 | -1, 53, -13 удалить с 1, удалить -5 элементов | Сообщение об ошибке | не верно введено число |
| 11 | -1, 53, -13 удалить с 1, удалить 3 элемента | Массив пустой | Массив пустой | Массив не пустой, удалить все элементы |
| 12 | Массив пустой | Массив пустой | Массив пустой | Массив пустой |