Федеральное государственное бюджетное образовательное учреждение высшего образования «Пермский национальный исследовательский политехнический университет»

Лабораторная работа №4

По дисциплине

“Объектно-ориентированное программирование”

Тема

“Выполнение программы простой структуры. Вычисление выражений с использованием стандартных функций”

Вариант 13

Выполнил работу

студент группы РИС-19-1б

Мазитов Т.Э.

Проверила

доцент кафедры ИТАС

Викентьева О.Л.

Работу выполнил:

Пермь 2020

**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.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Удаление | Добавление | Перестановка | Поиск | Сортировка |
| N элементов, начиная с номера K | K элементов в начало массива | Поменять местами минимальный и максимальный элементы | Первый четный | Простой обмен |

**Анализ задачи**

Пользовательское меню будет состоять из одного уровня и предлагать выбрать один из пунктов работы с одномерным массивом.

Пункты 2 - 7 не будут работать, пока массив не создан или пуст. После успешного выполнения любого пункта будет выведено меню.

Каждый пункт меню вызывает одну функцию. Если функция изменяет массив, то в качестве аргумента ей нужно передать массив и записать возвращаемый массив. Если функция не изменяет массив, то функции также передается массив, но возвращаемого значения нет. Для создания массива будет вызываться функция без аргументов, которая возвращает массив. Пользователю будет дана возможность выбрать, как заполнить массив: с помощью генератора псевдослучайных чисел или самому ввести все значения с клавиатуры.

Взаимодействия пользователя с программой представляет собой ввод с клавиатуры чисел, которые должны лежать в определенном промежутке значений. Поэтому каждый такой этап включает в себя валидацию этих значений на соответствие критериям. Пока все критерии не будут соблюдены, программа будет предупреждать пользователя и просить ввести значение еще раз.

Работы с одним типом массива не влияет на работу с другим типом массива, т.к. при возвращении в первый уровень меню все переменные и объекты удаляются из памяти.

**Проектирование**

В классе Program описаны методы:

* Main(): void – точка входа в программу. Вызывает метод Menu();
* ConsoleInput(): int – принимает число с консоли и возвращает его;
* InputChoice(): int – выбор способа заполнения массива и вызов ConsoleInput;
* PrintArray(): void – печатает в консоль переданный массив;
* CreateArray(): \*array – создает массив и возвращает его;
* AddElemets(): int[] – добавляет элементы в начало массива. Метод принимает массив, создает новый массив, добавляет в него новые элементы, копирует элементы из старого массива, возвращает новый массив;
* DeleteElements(): int[] – удаляет элементы из массива. Принимает массив, создает новый массив, копирует только нужные строчки, возвращает новый массив;
* ReplaceElements(): void – переставляет местами минимальный и максимальный элементы. Если таких элементов несколько, то спрашивает пользователя;
* FindElement(): void – ищет первое четное число и выводит информацию в консоль;
* SortArray(): void – сортирует массив методом простого обмена (метод пузырька);
* Menu() – выводит основное меню, обрабатывает ввод пользователя и вызывает функции.

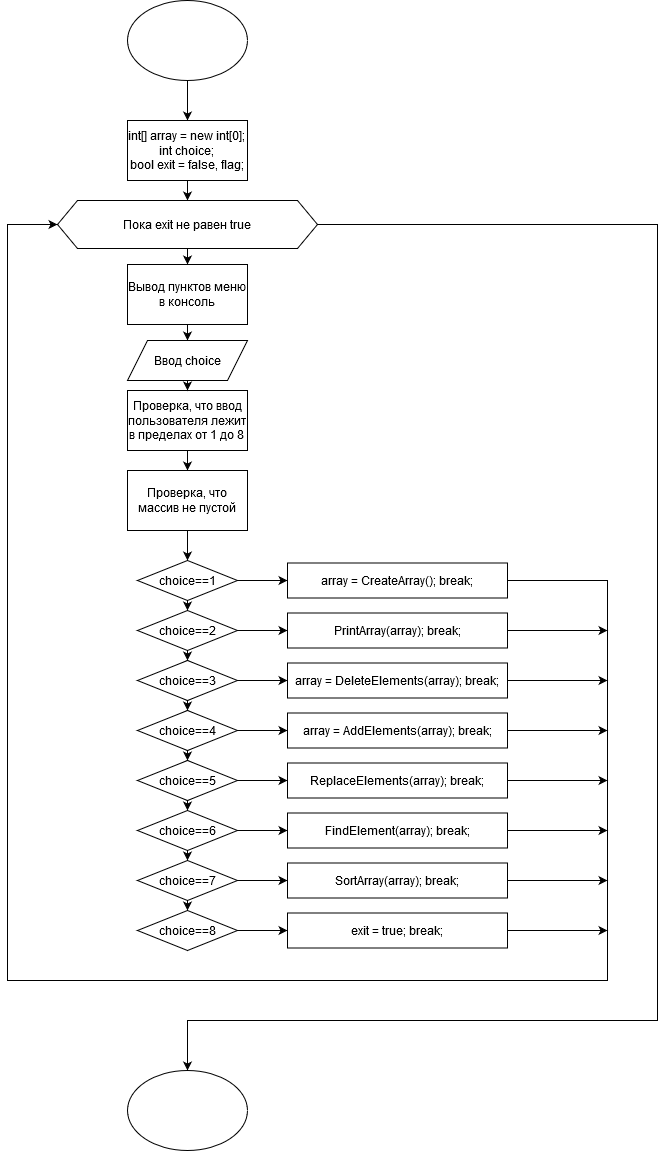
Некоторые встроенные функции, используемые в программе:

* String.Format() – заменяет элементы в строке на строковое представление объекта и возвращает полученную строку;
* Random.Next() – возвращает случайное число из диапазона;
* Math.Abs() – возвращает абсолютное значение числа.
* Convert.ToInt32() – преобразует строковое представление числа в число.

**Проектирование**

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Функция** | **Входные данные** | **Выходные данные** |
| **Основные функции** | | | |
| 1 | CreateArray - формирование одномерного массива. |  | Возвращает одномерный массив. Длину массива запрашивает у пользователя. |
| 2 | PrintArray - печать массива | int [] array - массив | Печать массива |
| 3 | DeleteElements – удаление элементов из массива | int [] oldarray - массив | int[] array - массив |
| 4 | AddElements - добавление элементов в массив | int [] oldarray - массив | int[] array - массив |
| 5 | ReplaceElements – перестановка местами минимального и максимального элементов. | int [] oldarray - массив |  |
| 6 | FindElement – поиск первого четного элемента. | int [] array - массив |  |
| 7 | SortArray – сортировка массива по возрастанию | int [] array - массив |  |
| 8 | ConsoleInput - ввод числа из консоли | string msg – сообщение для ввода,  int beg – начало диапазона допустимых значений,  int end – конец диапазона значений. | Возвращает целое число elem. |
| 9 | InputChoice – выбор метода заполнения массива |  | Вызывает функцию ConsoleInput. Возвращает число elem. |
| **Вспомогательные функции** | | | |
| 10 | Menu – диалог для ввода команд пользователя |  | Вызов функций  Сообщение об ошибке |

**Блок - схема**

****

**Листинг программы**

using System;

using System.Dynamic;

using System.Runtime.CompilerServices;

namespace LR4

{

class Program

{

static Random Rand = new Random();

static void Main() => Menu();

static int ConsoleInput(string msg, int beg, int end)

{

bool flag;

int elem = 0;

do

{

flag = false;

Console.WriteLine();

Console.Write(msg);

try

{

elem = Convert.ToInt32(Console.ReadLine());

if (elem < beg || elem > end) throw new IndexOutOfRangeException();

}

catch

{

flag = true;

Console.WriteLine("\nВы ввели некорректные данные. Повторите ввод.");

}

} while (flag);

return elem;

}

static int InputChoice()

{

Console.WriteLine("\n1. Заполнение массива случайными числами от 1 до 9.\n2. Ввод элементов массива с клавиатуры.");

return ConsoleInput("Ваш выбор: ",1,2);

}

static int[] CreateArray()

{

int l = ConsoleInput("Введите длину массива (от 1 до 10): ",1,10);

int choice = InputChoice();

int[] array = new int[l];

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

{

array[i] = (choice == 1) ? Rand.Next(1, 10) : ConsoleInput(String.Format("Введите элемент [{0}] (от {1} до {2}): ", i + 1, 0, 9), 0, 9);

}

Console.WriteLine("Создан одномерный массив длиной {0}", l);

return array;

}

static void PrintArray(int[] array)

{

Console.WriteLine("\nВывод элементов массива.\n");

foreach (var x in array) Console.Write(" {0} ", x);

Console.WriteLine();

for (int i = 1; i <= array.Length; i++) Console.Write("[{0}] ", i);

Console.WriteLine();

}

static int[] DeleteElements(int[] oldarray)

{

int begin = ConsoleInput("Включая с какого элемента производить удаления: ", 1, oldarray.Length);

int toDel = ConsoleInput("Сколько элементов удалить: ", 1, oldarray.Length - begin + 1);

int[] array = new int[oldarray.Length - toDel];

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

{

if (i < begin-1) array[i] = oldarray[i];

if (i >= begin + toDel - 1) array[i-toDel] = oldarray[i];

}

Console.WriteLine($"Из массива было удалено {toDel} элементов.");

return array;

}

static int[] AddElements(int[] oldarray)

{

int toAdd = ConsoleInput("Сколько элементов добавить в начало массива (от 1 до 10): ",1,10);

int[] array = new int[oldarray.Length + toAdd];

int choice = InputChoice();

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

{

if (i < toAdd) array[i] = (choice == 1) ? Rand.Next(1, 10) : ConsoleInput(String.Format("Введите элемент [{0}] (от {1} до {2}): ", i + 1, 0, 9), 0, 9);

else array[i] = oldarray[i - toAdd];

}

Console.WriteLine("\nВ массив было добвалено элементов: {0}", toAdd);

return array;

}

static void ReplaceElements(int[] oldarray)

{

int max = oldarray[0], min = oldarray[0];

int[,] positions = new int[2, oldarray.Length];

positions[0, 0] = 1;

positions[1, 0] = 1;

int max\_count = 1;

int min\_count = 1;

int max\_ch = 0, min\_ch = 0, temp;

for (int i = 1; i< oldarray.Length; i++)

{

if (oldarray[i] > max)

{

max = oldarray[i];

for (int j = 0; j < max\_count; j++) positions[0, j] = 0;

positions[0, 0] = i+1;

max\_count = 1;

}

else if (oldarray[i] == max)

{

positions[0, max\_count] = i+1;

max\_count++;

}

if (oldarray[i] < min)

{

min = oldarray[i];

for (int j = 0; j < min\_count; j++) positions[1, j] = 0;

positions[1, 0] = i+1;

min\_count = 1;

}

else if (oldarray[i] == min)

{

positions[1, min\_count] = i+1;

min\_count++;

}

}

for (int i = 0; i < 2; i++)

{

for (int j = 0; j < positions.GetLength(1); j++)

{

Console.Write(" "+positions[i, j]);

}

Console.WriteLine();

}

if (max\_count > 1)

{

Console.WriteLine($"\nКоличество максимальных элементов: {max\_count}");

max\_ch = ConsoleInput($"Какой максимальный элемент будем менять (от {1} до {max\_count}): ", 1, max\_count);

}

else max\_ch = 1;

if (min\_count > 1)

{

Console.WriteLine($"\nКоличество минимальных элементов: {min\_count}");

min\_ch = ConsoleInput($"Какой минимальный элемент будем менять (от {1} до {min\_count}): ", 1, min\_count);

}

else min\_ch = 1;

temp = oldarray[positions[0,max\_ch-1]-1];

oldarray[positions[0, max\_ch - 1]-1] = oldarray[positions[1,min\_ch-1]-1];

oldarray[positions[1, min\_ch - 1]-1] = temp;

}

static void FindElement(int[] array)

{

int count = 0, first\_even = 0, pos = 0;

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

{

count++;

if (array[i] % 2 == 0)

{

first\_even = array[i];

pos = i+1;

break;

}

}

Console.WriteLine($"\nПервое четное число {first\_even} в позиции {pos} было найднено за {count} {(count > 1 ? count > 4 ? "сравнений" : "сравнения" : "сравнение")}");

}

static void SortArray(int[] array)

{

int t;

for (int i=0; i < array.Length - 1; i++)

{

for (int j=0; j < array.Length - i -1; j++)

{

if (array[j] > array[j + 1])

{

t= array[j];

array[j] = array[j + 1];

array[j + 1] = t;

}

}

}

}

static void Menu()

{

int[] array = new int[0];

int choice;

bool exit = false, flag;

while (!exit)

{

Console.WriteLine();

Console.WriteLine(

@"Меню:

1. Создать одномерный массив;

2. Распечатать массив;

3. Удалить N элементов, начиная с номера K;

4. Добавить К элементов в начало массива;

5. Поменять местами минимальный и максимальный элементы;

6. Найти первый четный элемент;

7. Отсортировать методом простой сортировки;

8. Выход из программы.");

do

{

Console.Write("\nВаш выбор: ");

flag = int.TryParse(Console.ReadLine(), out choice);

if (!(1 <= choice && choice <= 8) || !flag) Console.WriteLine("\nВы ввели некорректные данные. Повторите ввод.");

} while (!(1 <= choice && choice <= 8) || !flag);

if (array.Length == 0 && choice >= 2 && choice <=7)

{

Console.WriteLine("\nСоздайте массив!");

}

else

{

switch (choice)

{

case 1: array = CreateArray(); break;

case 2: PrintArray(array); break;

case 3: array = DeleteElements(array); break;

case 4: array = AddElements(array); break;

case 5: ReplaceElements(array); break;

case 6: FindElement(array); break;

case 7: SortArray(array); break;

case 8: exit = true; break;

}

}

}

Console.WriteLine("\nЗавершение работы программы");

}

}

}

**Тестирование**

**Навигация по меню**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Ввод | ER | AR | Примечание |
| 1 | 0 | Сообщение об ошибке | Некорректные данные.  Повторите ввод. |  |
| 2 | 5 | Сообщение об ошибке | Некорректные данные. Повторите ввод. |
| 3 | -5 | Сообщение об ошибке | Некорректные данные. Повторите ввод. |
| 4 | f | Сообщение об ошибке | Некорректные данные. Повторите ввод. |
| 5 | abc | Сообщение об ошибке | Некорректные данные. Повторите ввод. |
| 7 | 1 | Создание массива | Создание массива |  |
| 8 | 2 | Печать массива | Печать массива |
| 9 | 8 | Выход из программы | Выход из программы |

**Формирование одномерного массива**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Ввод | ER | AR | Примечание |
| 1 | 1 | 1 число | 6 | Ввод длины массива |
| 2 | 5 | 5 чисел | 1 2 3 4 5 |
| 3 | 99 | 99 чисел | 99 чисел |
| 4 | 0 | Сообщение об ошибке | Некорректные данные. Повторите ввод. |
| 5 | -5 | Сообщение об ошибке | Некорректные данные. Повторите ввод. |
| 6 | 1.23 | Сообщение об ошибке | Некорректные данные. Повторите ввод. |
| 7 | 1,23 | Сообщение об ошибке | Некорректные данные. Повторите ввод. |
| 8 | abc | Сообщение об ошибке | Некорректные данные. Повторите ввод. |
| 9 | 100 | Сообщение об ошибке | Некорректные данные. Повторите ввод. |
| 10 | 1 | 1 | 1 | Ввод элементов массива с клавиатуры |
| 11 | -100 | -100 | -100 |
| 12 | 0 | 0 | 0 |
| 13 | 43567897656, 1.4, “abc”, “ ” | Сообщение об ошибке | Некорректные данные. Повторите ввод. |

**Удаление элементов массива**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Ввод | ER | AR | Примечание |
| 1 | 1 2 3 4 5  3 1 | 4 числа | 1 2 4 5 | Удалить с номера K удалить N элементов. |
| 2 | 1 2 3 4 5  4 2 | 3 числа | 1 2 3 |
| 3 | 1 2 3 4 5  5 2 | Сообщение об ошибке | Некорректные данные. Повторите ввод. |
| 4 | 1 2 3 4 5  1 6 | Сообщение об ошибке | Некорректные данные. Повторите ввод. |
| 5 | 1 2 3 4 5  -1 2 | Сообщение об ошибке | Некорректные данные. Повторите ввод. |
| 6 | 1 2 3 4 5  0 2 | Сообщение об ошибке | Некорректные данные. Повторите ввод. |
| 7 | 1 2 3 4 5  2 -2 | Сообщение об ошибке | Некорректные данные. Повторите ввод. |

**Добавление элементов в массив**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Ввод | ER | AR | Примечание |
| 1 | 1 2 3  2 | 5 чисел | 1 2 3 | Задан массив.  Вводится количества элементов.  Заполнение с помощью ДСЧ. |
| 2 | 1 2 3  4 | 7 чисел | 3 5 1 1 1 2 3 |
| 3 | 1 2 3  -1 | Сообщение об ошибке | Некорректные данные. Повторите ввод. |
| 4 | 1 2 3  0 | Сообщение об ошибке | Некорректные данные. Повторите ввод. |
| 5 | 1 2 3  11 | Сообщение об ошибке | Некорректные данные. Повторите ввод. |
| 6 | 1 2 3  abc | Сообщение об ошибке | Некорректные данные. Повторите ввод. |
| 7 | 1 2 3  1.2 | Сообщение об ошибке | Некорректные данные. Повторите ввод. |

**Перестановка элементов массива**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Ввод | ER | AR | Примечание |
| 1 | 1 2 3 4 5 | 5 2 3 4 1 | 5 2 3 4 1 | Когда минимальных или  максимальных элементов  больше 1, требуется ввод  пользователя |
| 2 | 1 | 1 | 1 |
| 3 | 1 1,  1 2 | 1 1 | 1 1 |
| 4 | 1 2 | 2 1 | 2 1 |
| 5 | 1 1 5,  1 | 5 1 1 | 5 1 1 |
| 6 | 1 1 5,  2 | 1 5 1 | 1 5 1 |
| 7 | 1 5 5,  1 | 5 1 5 | 5 1 5 |
| 8 | 1 5 5,  2 | 5 5 1 | 5 5 1 |

**Поиск первого четного элемента**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Ввод | ER | AR | Примечание |
| 1 | 1 5 2 7 | 2 в позиции 3 за 3 сравнения | 2 в позиции 3 за 3 сравнения |  |
| 2 | 2 2 | 2 в позиции 1 за 1 сравнение | 2 в позиции 1 за 1 сравнение |
| 3 | 0 0 0 | 0 в позиции 1 за 1 сравнение | 0 в позиции 1 за 1 сравнение |
| 4 | 1 3 5 3 2 | 2 в позиции 5 за 5 сравнений | 2 в позиции 5 за 5 сравнений |
| 5 | 1 3 5 | Нет четных чисел | Нет четных чисел |

**Сортировка массива**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Ввод | ER | AR | Примечание |
| 1 | 5 3 1 2 4 | 1 2 3 4 5 | 1 2 3 4 5 |  |
| 4 | 5 | 5 | 5 |
| 5 | 0 0 0 0 | 0 0 0 0 | 0 0 0 0 |
| 6 | 2 2 2 1 | 1 2 2 2 | 1 2 2 2 |

**Бинарный поиск элемента**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Ввод | ER | AR | Примечание |
| 1 | 1 2 2 5 6  1 | 1 в позиции 1 | 1 в позиции 1 |  |
| 2 | 1 1 1 5 5 5  5 | 5 в позиции 4 | 5 в позиции 4 |
| 3 | 0  0 | 0 в позиции 1 | 0 в позиции 1 |
| 4 | 8 8 8 8 8 8  8 | 8 в позиции 1 | 8 в позиции 1 |
| 5 | 1 3 5 6 7  8 | Сообщение об ошибке | Некорректные данные.  Повторите ввод. |
| 6 | 2 3 4 5 6  1 | Сообщение об ошибке | Некорректные данные.  Повторите ввод. |

**Печать массива**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Ввод | ER | AR | Примечание |
| 1 | Пустой массив | Сообщение об ошибке | Некорректные данные.  Повторите ввод. | Любой массив |
| 2 | Пустой массив  После удаления  элементов | Сообщение об ошибке | Некорректные данные.  Повторите ввод. | Двумерный массив |
| 3 | 1 5 2 7 8 9 | 1 5 2 7 8 9 | 1 5 2 7 8 9 | Одномерный массив |
| 4 | 5 | 5 | 5 |