

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Санкт-Петербургский государственный технологический институт (технический университет)»

Дисциплина: «Разработка программных систем»

Отчёт по лабораторной работе № 1. Использование массивов.

Выполнил студент группы №475:

Овчинников Р.С.

Проверили:

ст. преп. Федин А. К.

доц. к.т.н. Корниенко И. Г.

Санкт-Петербург

2019**Постановка задачи:**

Необходимо выделить массив требуемого размера, запросив его у пользователя при запуске программы или считав из файла. В программе должны быть предусмотрены три варианта заполнения исходного массива: пользователем с клавиатуры, из файла и случайными числами.

**Конкретное задание:**

Написать программу для циклического сдвига массива вправо на N позиций. **Исходные данные:**

Одномерный массив размером N.

**Особые ситуации:**

Были рассмотрены следующие особые ситуации:

* Цикличное выполнение программы
* Вывод сообщения об ошибке при вводе некорректных данных
* Проверка на запрещённые имена файлов (при работе с файлами)
* Предложение перезаписать файл, если он уже существует.
* Уведомление пользователя о важных действиях программы (уведомления о сохранении)
* Предложение выбора исходных данных
* Предложение сохранения массива и результатов
* Сохранение массива и результатов

**Алгоритм решения:**

Для выполнения поставленной задачи, был разработан алгоритм сдвига массива в права, при этом использовались дополнительная временная переменная tmp.

**Структура программы:**

Программа выполнена в единственном модуле Program.cs

Таблица 1 – Методы.

|  |  |  |
| --- | --- | --- |
| **Название** | **Возвращаемый тип** | **Краткое описание** |
| Main | void | Точка входа в программу |
| Info | void | Выводит в консоль информацию о программе и авторе |
| MainMenu | void | Стартовое меню |
| SecondMenu | void | Выводить в консоль подменю программы |

Продолжение таблицы 1 – Методы

|  |  |  |
| --- | --- | --- |
| **Название** | **Возвращаемый тип** | **Краткое описание** |
| WorkingOnArray | void | Получает и работает над введённым массивом |
| ShowArray | void | Выводит массив на экран |
| RandomInput | int | Реализует алгоритм с случайным заполнением данных |
| FileInput | void | Реализует алгоритм с заполнением данных из файла |
| ManualInput | int | Реализует заполнение массива вручную |
| MainFunction | int[] | Функция циклического сдвига массива |
| CheckInt | int | Проверка вводимого символа на целое число |
| YesNo | bool | Вывод меню выбора да/нет |
| SaveArray | void | Сохраняет результат в файл |

Таблица 2 – Используемые константы (enum)

|  |  |  |  |
| --- | --- | --- | --- |
| **Название** | **Значение (int)** | **enum** | **Краткое описание** |
| Start | 1 | Menu | Пункт меню MainMenu, начало программы |
| Exit | 2 | Menu | Пункт меню MainMenu, выход из программы |
| Yes | 1 | NoYes | Пункт меню YesNo, положительный ответ |
| No | 2 | NoYes | Пункт меню YesNo, отрицательный ответ |
| Manual | 1 | Input | Пункт меню SecondMenu, выбор ручного ввода данных |
| Random | 2 | Input | Пункт меню SecondMenu, выбор случайного заполнения данных |
| File | 3 | Input | Пункт меню SecondMenu, выбор ввода данных из файла |

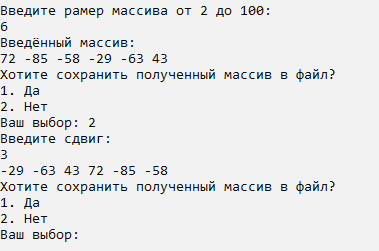
Продолжение таблицы 2 – Используемые константы (enum)

|  |  |  |  |
| --- | --- | --- | --- |
| **Название** | **Значение (int)** | **enum** | **Краткое описание** |
| MinInput | -99 | SizeAndInputArray | Минимально возможное значение в массиве |
| MinShift | 1 | SizeAndInputArray | Минимально возможное значение циклического сдвига массива |
| MinSize | 2 | SizeAndInputArray | Минимально возможный размер массива |
| MaxInput | 99 | SizeAndInputArray | Максимально возможное значение данных в массиве |
| MaxSize | 100 | SizeAndInputArray | Максимально возможный размер массива |

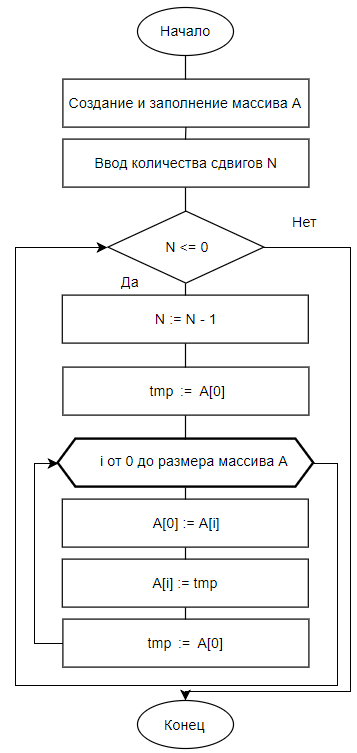
Таблица 3 – Переменные и массивы

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Краткое описание** |
| array | int[] | Исходный массив |
| arr | int[] | Переделанный массив |
| shift | int | Значение сдвига |
| caseSwitch | int | Выбор пункта (кейса) меню |
| size | int | Размер массива |
| path | string | Путь к файлу |
| cycle | bool | Зацикливание функций |
| num | int | Число вводимое пользователем |

**Результат работы программы:**

В результате работы программы в консоль будет выведен исходным массив и переделанный массив.

**Блок-схема:**



**Исходный код:**Модуль Program.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.IO;

namespace ConsoleApp1

{

public class Lab1

{

public enum Menu : int

{

Start = 1,

Exit,

}

public enum NoYes : int

{

Yes = 1,

No,

}

public enum Input : int

{

Manual = 1,

Random,

File,

}

public enum SizeAndInputArray : int

{

MinInput = -99,

MinShift = 1,

MinSize = 2,

MaxInput = 99,

MaxSize = 100,

}

static void Main()

{

Info();

MainMenu();

}

/// <summary>

/// Вывод информации о программе

/// </summary>

public static void Info()

{

Console.WriteLine("Лабораторная работа №1");

Console.WriteLine("Выполнил студент 475 группы: Овчинников Роман");

Console.WriteLine("Написать программу для циклического сдвига массива вправо на N позиций.");

}

/// <summary>

/// Стартовое меню

/// </summary>

public static void MainMenu()

{

Console.WriteLine("1. Начать программу");

Console.WriteLine("2. Выход из программы");

int caseSwitch = CheckInt((int)Menu.Start, (int)Menu.Exit);

switch (caseSwitch)

{

case (int)Menu.Start:

SecondMenu();

break;

case (int)Menu.Exit:

break;

}

}

/// <summary>

/// Получение массива

/// </summary>

public static void WorkingOnArray(int[] array)

{

Console.WriteLine("Введённый массив:");

ShowArray(array);

SaveArray(array);

Console.WriteLine("Введите сдвиг:");

int shift = CheckInt((int)SizeAndInputArray.MinShift, array.Length - 1);

ShowArray(MainFunction(array, shift));

SaveArray(MainFunction(array, shift), array, shift);

}

/// <summary>

/// Выбор способа ввода

/// </summary>

public static void SecondMenu()

{

Console.WriteLine("Выбор способа ввода:");

Console.WriteLine("1. Ручное заполнение");

Console.WriteLine("2. Случайное заполнение");

Console.WriteLine("3. Заполнение из файла");

int caseSwitch = CheckInt((int)Input.Manual, (int)Input.File);

switch (caseSwitch)

{

case (int)Input.Manual:

WorkingOnArray(ManualInput());

MainMenu();

break;

case (int)Input.Random:

WorkingOnArray(RandomInput());

MainMenu();

break;

case (int)Input.File:

FileInput();

MainMenu();

break;

}

}

/// <summary>

/// Вывод массива на экран

/// </summary>

public static void ShowArray(int[] array)

{

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

{

Console.Write(array[i] + " ");

}

Console.WriteLine(" ");

}

/// <summary>

/// Ручной ввод

/// </summary>

public static int[] ManualInput()

{

Console.WriteLine("Введите рамер массива от 2 до 100:");

int size = CheckInt((int)SizeAndInputArray.MinSize, (int)SizeAndInputArray.MaxSize);

int[] array = new int[size];

Console.WriteLine("Введите массив (значение от -99 до 99) через Enter:");

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

{

array[i] = CheckInt((int)SizeAndInputArray.MinInput, (int)SizeAndInputArray.MaxInput);

}

return array;

}

/// <summary>

/// Случайное заполнение

/// </summary>

public static int[] RandomInput()

{

Console.WriteLine("Введите рамер массива от 2 до 100:");

int size = CheckInt((int)SizeAndInputArray.MinSize, (int)SizeAndInputArray.MaxSize);

int[] array = new int[size];

Random random = new Random();

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

{

array[i] = random.Next(-100, 100);

}

return array;

}

/// <summary>

/// Файловый ввод

/// </summary>

public static void FileInput()

{

try

{

Console.WriteLine("Введите путь к файлу или имя файла:");

string path = Console.ReadLine();

string[] array = File.ReadAllLines(path);

int[] arr = new int[array.Length];

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

{

arr[i] = Convert.ToInt32(array[i]);

}

Console.WriteLine("Введённый массив:");

ShowArray(arr);

Console.WriteLine("Введите сдвиг:");

int shift = CheckInt((int)SizeAndInputArray.MinShift, arr.Length);

ShowArray(MainFunction(arr, shift));

SaveArray(MainFunction(arr, shift), arr, shift);

}

catch (Exception e)

{

Console.WriteLine("Ошибка! " + e.Message);

}

}

/// <summary>

/// Функция циклического сдвига

/// </summary>

public static int[] MainFunction(int[] array, int shift)

{

while (shift != 0)

{

int tmp = array[0];

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

{

array[0] = array[i];// сохраняем текущий элемент во временную ячейку 1

array[i] = tmp; // вставляем в текущую ячейку предыдущий

tmp = array[0];

}

shift--;

}

return array;

}

/// <summary>

/// Возврат целого числа

/// </summary>

static int CheckInt(int start, int finish)

{

bool cycle = true;

int num = 0;

while (cycle)

{

string strPick = Console.ReadLine();

if (int.TryParse(strPick, out num))

{

if ((num >= start) && (num <= finish))

{

cycle = false;

}

else

{

Console.WriteLine("Ошибка! Вводите числа от " + start + " до " + finish);

}

}

else

{

Console.WriteLine("Некорректный ввод! Вводите цифры!");

}

}

return num;

}

/// <summary>

/// Меню выбора Да\Нет

/// </summary>

public static bool YesNo()

{

Console.WriteLine("1. Да");

Console.WriteLine("2. Нет");

Console.Write("Ваш выбор: ");

int UserChoice = CheckInt((int)NoYes.Yes, (int)NoYes.No);

if (UserChoice == (int)NoYes.Yes)

return true;

else

return false;

}

/// <summary>

/// Сохранение в файл основого массива

/// </summary>

public static void SaveArray(int[] array)

{

Console.WriteLine("Хотите сохранить полученный массив в файл?");

if (YesNo())

{

Console.WriteLine("Введите путь к файлу или имя файла:");

string path = "";

try

{

path = Console.ReadLine();

using (StreamWriter sw = new StreamWriter(path)) {; }

}

catch (Exception e)

{

Console.WriteLine("Ошибка! " + e.Message);

Console.WriteLine("Попробовать ещё раз?");

SaveArray(array);

return;

}

if (File.Exists(path))

{

Console.WriteLine("Данный файл уже существует. Пезаписать?");

if (YesNo())

{

Save(path, array);

}

else

{

SaveArray(array);

}

}

else

{

Save(path, array);

}

}

}

/// <summary>

/// Сохрарение в файл переделанного массива

/// </summary>

public static void SaveArray(int[] arr, int[] array, int shift)

{

Console.WriteLine("Хотите сохранить полученный массив в файл?");

if (YesNo())

{

Console.WriteLine("Введите путь к файлу или имя файла:");

string path = "";

try

{

path = Console.ReadLine();

using (StreamWriter sw = new StreamWriter(path)) {; }

}

catch (Exception e)

{

Console.WriteLine("Ошибка! " + e.Message);

Console.WriteLine("Попробовать ещё раз?");

SaveArray(arr, array, shift);

return;

}

if (File.Exists(path))

{

Console.WriteLine("Данный файл уже существует. Пезаписать?");

if (!YesNo())

{

Save(path, arr, array, shift);

}

else

{

SaveArray(arr, array, shift);

}

}

else

{

File.WriteAllText(path, "");

Save(path, arr, array, shift);

}

}

}

/// <summary>

/// Само сохранение

/// </summary>

public static void Save(string path, int[] arr, int[] array, int shift)

{

using (StreamWriter sw = new StreamWriter(path))

{

sw.WriteLine("Сдвиг на " + shift);

sw.WriteLine("Изначальный массив:");

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

{

sw.Write(array[i] + " ");

}

sw.WriteLine("");

sw.WriteLine("Переделанный массив:");

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

{

sw.Write(arr[i] + " ");

}

}

Console.WriteLine("Файл успешно записан!");

}

/// <summary>

/// Само сохранение

/// </summary>

public static void Save(string path, int[] array)

{

using (StreamWriter sw = new StreamWriter(path))

{

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

{

sw.WriteLine(array[i]);

}

}

Console.WriteLine("Файл успешно записан!");

}

}

}

Модуль UninTest.cs

using System;

using ConsoleApp1;

using Microsoft.VisualStudio.TestTools.UnitTesting;

namespace UnitTestProject1

{

[TestClass]

public class UnitTest1

{

[TestMethod]

public void TestMethod1()

{

int[] array = new int[3];

array[0] = 1; array[1] = 2; array[2] = 3;

int shift = 1;

int[] result = new int[3];

result[0] = 3; result[1] = 1; result[2] = 2;

CollectionAssert.AreEqual(Lab1.MainFunction(array, shift), result);

}

}

}