Пермский филиал федерального государственного автономного образовательного учреждения высшего образования «Национальный исследовательский университет «Высшая школа экономики»

*Факультет экономики, менеджмента и бизнес-информатики*

Шульжик Кирилл Владимирович

**Отчёт по практической работе №6 (Вариант 16)**

студента образовательной программы бакалавриата «Программная инженерия» по направлению подготовки 09.03.04 Программная инженерия

Преподаватель

В.Д Марквирер

**Оглавление**

[Задание 3](#_Toc90818661)

[Постановка задачи 1. 3](#_Toc90818662)

[Постановка задачи 2. 3](#_Toc90818663)

[Анализ 4](#_Toc90818664)

[Проектирование 6](#_Toc90818665)

[Листинг 7](#_Toc90818666)

[Тесты 8](#_Toc90818667)

[ПРИЛОЖЕНИЕ А *Листинг* 23](#_Toc90818668)

Задание

Постановка задачи 1.

Постановка задачи 2 выглядит следующим образом:

1. Создать динамический массив (одномерный, двумерный, рваный) из элементов заданного типа. При заполнении массива использовать 2 способа (ручной и с помощью ДСЧ).
2. Массив вывести на печать.
3. Выполнить операции с массивом, указанные в варианте, используя, по возможности, методы класса Array.
4. Результаты обработки вывести на печать.

Постановка задачи 2.

Постановка задачи 2 выглядит следующим образом:

1. Ввести строку символов (с клавиатуры или из массива заранее сформированных тестовых строк). Строка состоит из слов, разделенных пробелами (пробелов может быть несколько) и знаками препинания (, ;:). В строке может быть несколько предложений, в конце каждого предложения стоит один знак препинания (.!?).
2. Выполнить обработку строки в соответствии с вариантом, используя по возможности, методы класса String.
3. Результаты обработки вывести на печать.

Таблица 1 – Формулировка 1 задачи для 16 варианта

|  |  |  |  |
| --- | --- | --- | --- |
| 16 | Одномерный | Double | Отсортировать по убыванию только отрицательные элементы массива, остальные элементы остаются на своих местах. |

Формулировка 1 задачи 16 варианта представлено в таблице 1

Таблица 2 – Формулировка 2 задачи 16 варианта

|  |  |
| --- | --- |
| 16 | Определить есть ли в строке ключевые слова C#. Если есть, то напечатать сколько раз встречается каждое слово. |

Формулировка 2 задачи 16 варианта представлена в таблице 2

Анализ

Таблица 3 – Анализ

| **№** | **Описание функции** | **Входные данные** | **Классы входных данных** | **Выходные данные** |
| --- | --- | --- | --- | --- |
| **Основные функции** | | | | |
| **1** | формирование случайного одномерного массива с помощью ДСЧ | Одномерный массив arr (ссылка на массив) | Массив типа double |  |
| **2** | формирование одномерного массива с клавиатуры | Одномерный массив arr (ссылка на массив) | Массив типа double |  |
| **3** | Печать одномерного массива | Одномерный массив arr | Массив типа double |  |
| Строка-сообщение msg для взаимодействия с пользователем  (умалчиваемый параметр) | Строковый |
| **4** | Сортировка отрицательных элементов массива | Одномерный массив arr | Массив типа double | Отсортированный одномерный массив типа double |
| **5** | Поиск ключевых слов С# в строке | Строка phrase – строка, в которой будет осуществляться поиск ключевых слов С# | Строковый |  |
| Одномерный массив KeyWords (содержит все ключевые слова С#) | Строковый |
| Вспомогательные функции | | | | |
| **1** | Ввод целочисленной переменной | Выходной параметр x | Целочисленный |  |
| Левая граница left вводимой переменной  (умалчиваемый параметр) | Целочисленный |
| Правая граница right вводимой переменной  (умалчиваемый параметр) | Целочисленный |
| Строка-сообщение msg для взаимодействия с пользователем (умалчиваемый параметр) | Строковый |
| **2** | Ввод переменной типа double | Выходной параметр x | вещественный (тип double) |  |
| Строка-сообщение msg для взаимодействия с пользователем (умалчиваемый параметр) | строковый |
| **3** | Печать стартового меню |  |  |  |
| **4** | Печать меню, предназначенного для работы с первым заданием |  |  |  |
| **5** | Печать меню, предназначенного для работы с вторым заданием |  |  |  |

Анализ представлен в таблице 3

Проектирование

Таблица 4 – Проектирование

| **№** | **Функция** | **Входные данные** | **Выходные данные** |
| --- | --- | --- | --- |
| **Основные функции** | | | |
| **1** | MakeRandomMas (формирование одномерного массива с помощью ДСЧ) | ref double[] arr – одномерный массив типа double (ссылка на массив) |  |
| **2** | MakeManualMas (формирование одномерного массива с клавиатуры) | ref double[] arr – одномерный массив типа double (ссылка на массив) |  |
| **3** | PrintMas (печать одномерного массива) | ref double[] arr – одномерный массив типа double (ссылка на массив)  string msg = “” – строка-сообщение для взаимодействия с пользователем (умалчиваемый параметр) |  |
| **4** | SortOtriz (сортировка отрицательных элементов одномерного массива) | double[] arr– одномерный массив типа double | double[] arr – отсортированный одномерный массив типа double |
| **5** | keywords (поиск ключевых слов С# в строке) | string phrase – строка, в которой будет осуществляться поиск ключевых слов С#  string[] KeyWords – массив, содержащий все ключевые слова C# |  |
| Вспомогательные функции | | | |
| **1** | CheckInt (ввод целочисленной переменной) | out int x – выходной параметр |  |
| string msg = “” – строка-сообщение для взаимодействия с пользователем (умалчиваемый параметр) |
| int left (левая граница (умалчиваемый параметр)) |
| int right (правая граница (умалчиваемый параметр)) |
| **2** | CheckDouble ввод переменной типа double) | out double x – выходной параметр |  |
| string msg = “”– строка-сообщение для взаимодействия с пользователем (умалчиваемый параметр) |
| **3** | MenuStart (печать стартового пеню) |  |  |
| **4** | MenuFirstExercise (печать меню, предназначенного для работы с первым заданием) |  |  |
| **5** | MenuSecondExercise (Печать меню, предназначенного для работы с вторым заданием) |  |  |

Проектирование представлено в таблице 4

Листинг

См. [ПРИЛОЖЕНИЕ А](#Приложение).

Тесты

ПРИМЕЧАНИЕ. Тестирование ввода не чисел в таких входных данных как длина массива в дальнейший функциях тестироваться не будет, так как за это отвечает функция для ввода целочисленной переменной ChekInt (соответственно тестирование ввода не чисел будет производиться в этой функции). Тестирование ввода не чисел в таких входных данных как элементы массива типа Double в дальнейших функциях также не будет, так как за это отвечает функция для ввода переменной типа Double CheckDouble (соответственно тестирование ввода не чисел будет производиться в этой функции)

Таблица 5 – Набор тестов для функции, предназначенной для формирования одномерного массива с помощью ДСЧ

| №Теста | Входные данные | Ожидаемый результат | Выходные данные |
| --- | --- | --- | --- |
| 1 | -5 | Ошибка | Введите целое число, большее или равное 0 |
| 2 | 0 | Пустой массив/Повторный ввод | Массив пустой, дальнейшие действия не имеют смысла, введите кол-во элементов заново |
| 3 | 4 | Массив сформирован | Массив сформирован. Нажмите любую клавишу, чтобы продолжить |

Набор тестов для функции, предназначенной для формирования одномерного массива с помощью ДСЧ представлен в таблице 5.

Таблица 6 – Тестирование функции, предназначенной для формирования одномерного массива с помощью ДСЧ, по критериям черного ящика

| Критерии | Т1 | Т2 | Т3 |
| --- | --- | --- | --- |
| Длина массива меньше нуля | + |  |  |
| Длина массива больше нуля |  |  | + |
| Длина массива равна нулю |  | + |  |

Тестирование функции, предназначенной для формирования одномерного массива с помощью ДСЧ, по критериям черного ящика представлено в таблице 6.

Таблица 7 – Набор тестов для функции, предназначенной для формирования одномерного массива с клавиатуры

| №Теста | Входные данные | Ожидаемый результат | Выходные данные |
| --- | --- | --- | --- |
| 1 | -5 | Ошибка | Введите целое число, большее или равное 0 |
| 2 | 0 | Пустой массив/Повторный ввод | Массив пустой, дальнейшие действия не имеют смысла, введите кол-во элементов заново |
| 3 | 4 | Ввод 1 элемента массива | Введите значение элемента под номером 1 |
| -5,4 | Ввод 2 элемента массива | Введите значение элемента под номером 2 |
| 4,4 | Ввод 3 элемента массива | Введите значение элемента под номером 3 |
| -3 | Ввод 4 элемента массива | Введите значение элемента под номером 4 |
| 5 | Массив сформирован | Массив сформирован. Нажмите любую клавишу, чтобы продолжить |

Набор тестов для функции, предназначенной для формирования одномерного массива с клавиатуры представлен в таблице 7.

Таблица 8 – Тестирование функции, предназначенной для формирования одномерного массива с клавиатуры, по критериям черного ящика

| Критерии | Т1 | Т2 | Т3 |
| --- | --- | --- | --- |
| Длина массива меньше нуля | + |  |  |
| Длина массива больше нуля |  |  | + |
| Длина массива равна нулю |  | + |  |
| Элементы массива – целые числа |  |  | + |
| Элементы массива – вещественный числа (тип Double) |  |  | + |

Тестирование функции, предназначенной для формирования одномерного массива с клавиатуры, по критериям черного ящика представлено в таблице 8.

ПРИМЕЧАНИЕ. Тестирование печати пустого массива производиться не будет, так как сформировать пустой массив невозможно (при вводе 0 элементов обязателен повторный ввод).

Таблица 9 – Набор тестов для функции, предназначенной для печати одномерного массива

| №Теста | Входные данные | Ожидаемый результат | Выходные данные |
| --- | --- | --- | --- |
| 1 | 4 -5,4 4,4 -3 5 | Распечатанный одномерный массив | 4, -5,4 4,4 -3,5 |
| 2 | 12324324324 1 5345435 | Распечатанный одномерный массив | 12324324324 1 5345435 |

Набор тестов для функции, предназначенной для печати одномерного массива представлен в таблице 9.

Таблица 10 – Тестирование функции, предназначенной для печати одномерного массива, по критериям черного ящика

| Критерии | Т1 | Т2 |
| --- | --- | --- |
| Разница длины максимального и минимального элемента – значительное число (больше 3) |  | + |
| Разница длины максимального и минимального элемента <= 3 | + |  |

Тестирование функции, предназначенной для печати одномерного массива, по критериям черного ящика представлено в таблице 10.

Таблица 11 – Набор тестов для функции, предназначенной для сортировки отрицательных элементов одномерного массива

| №Теста | Входные данные | Ожидаемый результат | Выходные данные |
| --- | --- | --- | --- |
| 1 | 23,4 -4,1 53,3 -98,1 15,3 | Массив с отсортированными отрицательными элементами | 23,4 -98,1 53,3 -4,1 15,3 |
| 2 | -34,4 -93,1 -23 -43,3 -99 | Отсортированный массив | -99 -93,1 -43,3 -34,4 -23 |
| 3 | 24,1 34,4 12 1 43,2 | Исходный массив | 24,1 34,4 12 1 43,2 |

Набор тестов для функции, предназначенной для сортировки отрицательных элементов одномерного массива представлен в таблице 11.

Таблица 12 – Тестирование функции, предназначенной для сортировки отрицательных элементов одномерного массива, по критериям черного ящика

| Критерии | Т1 | Т2 | Т3 |
| --- | --- | --- | --- |
| Массив состоит только из отрицательных неотсортированных элементов |  | + |  |
| Массив состоит только из положительных неотсортированных элементов |  |  | + |
| Массив состоит и из отрицательных неотсортированных элементов, и из положительных неотсортированных элементов | + |  |  |

Тестирование функции, предназначенной для сортировки отрицательных элементов одномерного массива, по критериям черного ящика представлено в таблице 12.

ПРИМЕЧАНИЕ. Тестирование ввода различных знаков препинания (?!.) в конце не тестируется, так как в моей функции признак окончания последовательности – предварительно добавленный пробел, стоящий на последнем месте. Также подробно не тестируется отделение ключевых слов такими знаками, как ;:, поскольку для них условия в функции те же, что и для запятой.

Таблица 13 – Набор тестов для функции, предназначенной для поиска ключевых слов С# в строке

| №Теста | Входные данные | Ожидаемый результат | Выходные данные |
| --- | --- | --- | --- |
| 1 | А уж Онегин вышел вон; домой одеться едет ulong о return н. | 2 ключевых слова в строке | Найдено ключевых слов C# - 2 |
| 2 | Купаться в грязной реке decimal идея. | 1 ключевое слово в строке | Найдено ключевых слов C# - 1 |
| 3 | Я, int, а вы? | 1 ключевое слово в строке | Найдено ключевых слов C# - 1 |
| 4 | Ключевое слово отделено, ulong , образом | 1 ключевое слово в строке | Найдено ключевых слов C# - 1 |
| 5 | А вдруг пользователь решит ввести,double, вот так. | 1 ключевое слово в строке | Найдено ключевых слов C# - 1 |
| 6 | А теперь, continue; проверим, int , ,break: всеми возможными ,double способами! | 4 ключевых слова в строке | Найдено ключевых слов C# - 4 |

Набор тестов для функции, предназначенной для поиска ключевых слов С# в строке представлен в таблице 13.

Таблица 14 – Тестирование функции, предназначенной для поиска ключевых слов С# в строке, по критериям черного ящика

| Критерии | Т1 | Т2 | Т3 | Т4 | Т5 | Т6 |
| --- | --- | --- | --- | --- | --- | --- |
| Ключевое слово отделено несколькими пробелами | + |  |  |  |  |  |
| Ключевое слово отделено 1 пробелом слева и 1 пробелом справа |  | + |  |  |  |  |
| Ключевое слово отделено следующим образом: запятая, пробел, ключевое слово, запятая |  |  | + |  |  |  |
| Ключевое слово отделено следующим образом: запятая, пробел, ключевое слово, пробел, запятая |  |  |  | + |  |  |
| Ключевое слово отделено следующим образом: запятая, ключевое слово, запятая |  |  |  |  | + |  |
| Ключевое слово отделено разными знаками (,:;) и разными способами |  |  |  |  |  | + |

Тестирование функции, предназначенной для поиска ключевых слов С# в строке, по критериям черного ящика представлено в таблице 14.

Таблица 15 – Набор тестов для функции, предназначенной для ввода целочисленной переменной

| №Теста | Входные данные | Ожидаемый результат | Выходные данные |
| --- | --- | --- | --- |
| 1 | left = 2  right = 5  msg = “Ввод числа x” | Вывод сообщения | “Ввод числа x” |
| “фыва” | Ошибка/повторный ввод | Введите целое число, большее или равное 2 и меньшее или равное 5 |
| 0 | Ошибка/повторный ввод | Введите целое число, большее или равное 2 и меньшее или равное 5 |
| 7 | Ошибка/повторный ввод | Введите целое число, большее или равное 2 и меньшее или равное 5 |
| 3 | Присвоение переменной значение 3 | int x = 3 |
| 2 | left = 2  right = 3  msg: “Ввод числа х” | Вывод сообщения | “Ввод числа x” |
| 1 | Ошибка/повторный ввод | Введите либо 2, либо 3 |
| 2 | Присвоение переменной значение 2 | int x = 2 |
| 3 | left = 2  msg: “Ввод числа x” | Вывод сообщения | “Ввод числа x” |
| 1 | Ошибка/повторный ввод | Введите целое число, большее или равное 2 |
| 3 | Присвоение переменной значение 3 | int x = 3 |
| 4 | right = 5  msg: “Ввод числа x” | Вывод сообщения | “Ввод числа x” |
| 8 | Ошибка/повторный ввод | Введите целое число, меньшее или равное 5 |
| 3 | Присвоение переменной значение 3 | int x = 3 |
| 5 | msg: “Ввод числа x” | Вывод сообщения | “Ввод числа x” |
| “dsg” | Ошибка/повторный ввод | Введите целое число |
| 7 | Присвоение переменной значение 7 | int x = 7 |
| 6 | msg: “Ввод x”  left = 5  right = 5 | Вывод сообщения | “Ввод числа x” |
| 4 | Ошибка/повторный ввод | Введите число 5 |
| 5 | Присвоение переменной значение 5 | int x = 5 |
| 7 | left = 10  rigt = 5 | Сообщение об ошибке | Неверно заданы границы диапазона |
| 8 | msg = “” | 5 | int x = 7 |

Набор тестов для функции, предназначенной для ввода целочисленной переменной представлен в таблице 15

Таблица 16 – Тестирование функции, предназначенной для ввода целочисленной переменной, по критериям черного ящика

| Критерии | Т1 | Т2 | Т3 | Т4 | Т5 | Т6 | Т7 | Т8 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Ввод не целого числа | + |  |  |  | + |  |  |  |
| Ввод числа, не входящего в заданный диапазон | + | + | + | + |  | + |  |  |
| Задана только левая граница диапазона |  |  | + |  |  |  |  |  |
| Задана только правая граница диапазона |  |  |  | + |  |  |  |  |
| Границы диапазона не заданы |  |  |  |  | + |  |  |  |
| Заданный диапазон состоит из 1 числа |  |  |  |  |  | + |  |  |
| Левая граница превышает правую границу |  |  |  |  |  |  | + |  |
| Сообщение msg пустое |  |  |  |  |  |  |  | + |

Тестирование функции, предназначенной для ввода целочисленной переменной, по критериям черного ящика представлено в таблице 16.

Таблица 17 – Набор тестов для функции, предназначенной для ввода переменной типа Double

| №Теста | Входные данные | Ожидаемый результат | Выходные данные |
| --- | --- | --- | --- |
| 1 | msg = “Введите значение элемента под номером 1” | Вывод строки-сообщения | Введите значение элемента под номером 1 |
| “dfasf” | Сообщение об ошибке | Введите число! |
| 4,5 | Присвоение переменной числа 4,5 | float x = 4,5 |
| 2 | 4 | Присвоение переменной числа 4 | float x = 4 |

Набор тестов для функции, предназначенной для ввода переменной типа double представлен в таблице 17.

Таблица 18 – Тестирование функции, предназначенной для ввода переменной типа Double, по критериям черного ящика

| Критерии | Т1 | Т2 |
| --- | --- | --- |
| Ввод переменной типа double с строкой-сообщением для взаимодействия с пользователем | + |  |
| Ввод переменной типа double без строки-сообщения для взаимодействия с пользователем |  | + |
| Ввод не числа | + |  |

Тестирование функции, предназначенной для ввода переменной типа double, по критериям черного ящика представлено в таблице 18.

Таблица 19 – Набор тестов для функции, предназначенной для печати стартового меню

| №Теста | Входные данные | Ожидаемый результат | Выходные данные |
| --- | --- | --- | --- |
| 1 |  | Печать стартового меню | 1. Задача номер 1 (массивы)  2. Задача номер 2 (строки)  0. Выход из программы |
| 0 | Выход из программы | Благодарим за использование программы! |
| 2 |  | Печать стартового меню | 1. Задача номер 1 (массивы)  2. Задача номер 2 (строки)  0. Выход из программы |
| 1 | Вывод меню, предназначенного для работы с 1 заданием | 1. Сформировать одномерный массив типа Double с помощью ДСЧ  2. Сформировать одномерный массив типа Double с клавиатуры  3. Распечатать массив  4. Отсортировать только отрицательные элементы массива  0. Назад |
| 3 |  | Печать стартового меню | 1. Задача номер 1 (массивы)  2. Задача номер 2 (строки)  0. Выход из программы |
| 2 | Вывод меню, предназначенного для работы с 2 заданием | 1. Формирование случайной строки  2. Формирование строки с клавиатуры  3. Распечатать строку  4. Проверка на содержание ключевых слов в строке  0. Назад |
| 4 |  | Печать стартового меню | 1. Задача номер 1 (массивы)  2. Задача номер 2 (строки)  0. Выход из программы |
| 5 | Ошибка/повторный ввод | Введите целое число от 0 до 2! |
| “sfsf” | Ошибка/повторный ввод | Введите целое число от 0 до 2! |

Набор тестов для функции, предназначенной для печати стартового меню представлен в таблице 19.

Таблица 20 – Тестирование функции, предназначенной для печати стартового меню, по критериям черного ящика

| Критерии | Т1 | Т2 | Т3 | Т4 |
| --- | --- | --- | --- | --- |
| Выбор 1 пункта меню |  | + |  |  |
| Выбор 2 пункта меню |  |  | + |  |
| Выбор 3 пункта меню | + |  |  |  |
| Ввод символа, не соответствующего ни одному пункту меню |  |  |  | + |

Тестирование функции, предназначенной для печати стартового меню, по критериям черного ящика представлено в таблице 20.

Таблица 21 – Набор тестов для функции, предназначенной для печати меню, предназначенного для работы с 1 заданием

| №Теста | Входные данные | Ожидаемый результат | Выходные данные |
| --- | --- | --- | --- |
| 1 |  | Печать меню, предназначенного для работы с 1 заданием | 1. Сформировать одномерный массив типа Double с помощью ДСЧ  2. Сформировать одномерный массив типа Double с клавиатуры  3. Распечатать массив  4. Отсортировать только отрицательные элементы массива  0. Назад |
| 0 | Возвращение в стартовое меню | 1. Задача номер 1 (массивы)  2. Задача номер 2 (строки)  0. Выход из программы |
| 2 |  | Печать меню, предназначенного для работы с 1 заданием | 1. Сформировать одномерный массив типа Double с помощью ДСЧ  2. Сформировать одномерный массив типа Double с клавиатуры  3. Распечатать массив  4. Отсортировать только отрицательные элементы массива  0. Назад |
| 1 | Переход в функцию, предназначенную для формирования одномерного массива типа Double с помощью ДСЧ | Введите кол-во элементов массива |
| 3 |  | Печать меню, предназначенного для работы с 1 заданием | 1. Сформировать одномерный массив типа Double с помощью ДСЧ  2. Сформировать одномерный массив типа Double с клавиатуры  3. Распечатать массив  4. Отсортировать только отрицательные элементы массива  0. Назад |
| 2 | Переход в функцию, прденазначенную для формирования одномерного массива типа Double с клавиатуры | Введите кол-во элементов массива |
| 4 |  | Печать меню, предназначенного для работы с 1 заданием | 1. Сформировать одномерный массив типа Double с помощью ДСЧ  2. Сформировать одномерный массив типа Double с клавиатуры  3. Распечатать массив  4. Отсортировать только отрицательные элементы массива  0. Назад |
| 3 | Переход в функцию, предназначенную для печати одномерного массива типа Double | Массив представлен ниже  3 4 4,4 3 3,4 |
| 5 |  | Печать меню, предназначенного для работы с 1 заданием | 1. Сформировать одномерный массив типа Double с помощью ДСЧ  2. Сформировать одномерный массив типа Double с клавиатуры  3. Распечатать массив  4. Отсортировать только отрицательные элементы массива  0. Назад |
| 4 | Переход в функцию, предназначенную для сортировки только отрицательных элементов одномерного массива типа Double | Полученный массив представлен ниже:  -7 61,1 -5,9 -1  Исходный массив:  -1 61,1 -7 -5,9 |
| 6 |  | Печать меню, предназначенного для работы с 1 заданием | 1. Сформировать одномерный массив типа Double с помощью ДСЧ  2. Сформировать одномерный массив типа Double с клавиатуры  3. Распечатать массив  4. Отсортировать только отрицательные элементы массива  0. Назад |
| 5 | Сообщение об ошибке/повторный ввод | Введите целое число от 0 до 4! |
| “sdf" | Сообщение об ошибке/повторный ввод | Введите целое число от 0 до 4! |

Набор тестов для функции, предназначенной для печати меню, предназначенного для работы с 1 заданием представлен в таблице 21.

Таблица 22 – Тестирование функции, предназначенной для печати меню, предназначенного для работы с 1 заданием, по критериям черного ящика

| Критерии | Т1 | Т2 | Т3 | Т4 | Т5 | Т6 |
| --- | --- | --- | --- | --- | --- | --- |
| Выбор 1 пункта меню |  | + |  |  |  |  |
| Выбор 2 пункта меню |  |  | + |  |  |  |
| Выбор 3 пункта меню |  |  |  | + |  |  |
| Выбор 4 пункта меню |  |  |  |  | + |  |
| Выбор 5 пункта меню | + |  |  |  |  |  |
| Ввод символа, не соответствующего ни одному пункту меню |  |  |  |  |  | + |

Тестирование функции, предназначенной для печати меню, предназначенного для работы с 1 заданием, по критериям черного ящика представлено в таблице 22.

Таблица 23 – Набор тестов для функции, предназначенной для печати меню, предназначенного для работы с 2 заданием

| №Теста | Входные данные | Ожидаемый результат | Выходные данные |
| --- | --- | --- | --- |
| 1 |  | Печать меню, предназначенного для работы с 2 заданием | 1. Формирование случайной строки  2. Формирование строки с клавиатуры  3. Распечатать строку  4. Проверка на содержание ключевых слов в строке  0. Назад |
| 0 | Возвращение в стартовое меню | 1. Задача номер 1 (массивы)  2. Задача номер 2 (строки)  0. Выход из программы |
| 2 |  | Печать меню, предназначенного для работы с 2 заданием | 1. Формирование случайной строки  2. Формирование строки с клавиатуры  3. Распечатать строку  4. Проверка на содержание ключевых слов в строке  0. Назад |
| 1 | Формирование случайной строки | Строка сформирована |
| 3 |  | Печать меню, предназначенного для работы с 1 заданием | 1. Формирование случайной строки  2. Формирование строки с клавиатуры  3. Распечатать строку  4. Проверка на содержание ключевых слов в строке  0. Назад |
| 2 | Формирование строки с клавиатуры | Введите строку |
| 4 |  | Печать меню, предназначенного для работы с 2 заданием | 1. Формирование случайной строки  2. Формирование строки с клавиатуры  3. Распечатать строку  4. Проверка на содержание ключевых слов в строке  0. Назад |
| 3 | Вывод сформированной строки | Строка представлена ниже:  То стан совьет, то разовьет и быстрой ножкой ножку бьет. |
| 5 |  | Печать меню, предназначенного для работы с 2 заданием | 1. Формирование случайной строки  2. Формирование строки с клавиатуры  3. Распечатать строку  4. Проверка на содержание ключевых слов в строке  0. Назад |
| 4 | Переход в функцию, предназначенную для поиска ключевых слов С# в строке | В строке нет ключевых слов C# |
| 6 |  | Печать меню, предназначенного для работы с 2 заданием | 1. Формирование случайной строки  2. Формирование строки с клавиатуры  3. Распечатать строку  4. Проверка на содержание ключевых слов в строке  0. Назад |
| 5 | Сообщение об ошибке/повторный ввод | Введите целое число от 0 до 4! |
| “sdf" | Сообщение об ошибке/повторный ввод | Введите целое число от 0 до 4! |

Набор тестов для функции, предназначенной для печати меню, предназначенного для работы с 2 заданием представлен в таблице 23.

Таблица 24 – Тестирование функции, предназначенной для печати меню, предназначенного для работы с 2 заданием, по критериям черного ящика

| Критерии | Т1 | Т2 | Т3 | Т4 | Т5 | Т6 |
| --- | --- | --- | --- | --- | --- | --- |
| Выбор 1 пункта меню |  | + |  |  |  |  |
| Выбор 2 пункта меню |  |  | + |  |  |  |
| Выбор 3 пункта меню |  |  |  | + |  |  |
| Выбор 4 пункта меню |  |  |  |  | + |  |
| Выбор 5 пункта меню | + |  |  |  |  |  |
| Ввод символа, не соответствующего ни одному пункту меню |  |  |  |  |  | + |

Тестирование функции, предназначенной для печати меню, предназначенного для работы с 2 заданием, по критериям черного ящика представлено в таблице 24.

ПРИЛОЖЕНИЕ А

**Листинг**

using System;

namespace Лаба\_6

{

class Program

{

static void CheckInt(out int x, string msg = "", int left = Int32.MinValue, int right = Int32.MaxValue)

{

x = 0;

if (right < left && right != Int32.MaxValue && left != Int32.MinValue)

Console.WriteLine("Неверно заданы границы диапазона");

else

{

if (msg != "")

Console.WriteLine(msg);

string bug = Console.ReadLine();

while ((!int.TryParse(bug, out x) || int.Parse(bug) < left || int.Parse(bug) > right))

{

Console.WriteLine();

if (left != Int32.MinValue && right != Int32.MaxValue && left == right)

{

Console.WriteLine($"Введите число {left}");

bug = Console.ReadLine();

}

else if (left != Int32.MinValue && right != Int32.MaxValue && right - left == 1)

{

Console.WriteLine($"Введите либо {left}, либо {right}");

bug = Console.ReadLine();

}

else if (left != Int32.MinValue && right != Int32.MaxValue)

{

Console.WriteLine($"Введите целое число, большее или равное {left} и меньшее или равное {right}");

bug = Console.ReadLine();

}

else if (left != Int32.MinValue)

{

Console.WriteLine($"Введите целое число, большее или равное {left}");

bug = Console.ReadLine();

}

else if (right != Int32.MaxValue)

{

Console.WriteLine($"Введите целое число, меньшее или равное {right}");

bug = Console.ReadLine();

}

else

{

Console.WriteLine($"Введите целое число");

bug = Console.ReadLine();

}

}

x = int.Parse(bug);

}

}

static void CheckDouble(out double x, string msg = "")

{

x = 0;

if (msg != "")

Console.WriteLine(msg);

string bug = Console.ReadLine();

while (!Double.TryParse(bug, out x))

{

Console.WriteLine($"Введите число!");

bug = Console.ReadLine();

}

x = Double.Parse(bug);

}

static void MakeRandomMas(ref double[] arr)

{

CheckInt(out int size, msg: "Введите кол-во элементов массива", left: 0);

while (size == 0)

CheckInt(out size, msg: "Массив пустой, дальнейшие действия не имеют смысла." +

" Введите кол-во элементов заново", left: 0);

Random rand = new();

arr = new double[size];

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

arr[i] = Math.Round(rand.NextDouble() \* rand.Next(-100,100), 1);

Console.WriteLine("\nМассив сформирован");

}

static void MakeManualMas(ref double[] arr)

{

CheckInt(out int size, msg: "Введите кол-во элементов массива", left: 0);

while (size == 0)

CheckInt(out size, msg: "Массив пустой, дальнейшие действия не имеют смысла." +

" Введите кол-во элементов массива заново", left: 0);

int k = 1;

arr = new double[size];

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

{

CheckDouble(out double x, msg: $"\nВведите значение элемента под номером {k}");

arr[k - 1] = x;

k += 1;

}

Console.WriteLine("\nМассив сформирован");

}

static void PrintMas(double[] arr, string msg = "")

{

if (msg != "")

Console.WriteLine(msg + "\n");

foreach (double x in arr)

{

Console.Write(x + " ");

}

Console.WriteLine("\n");

}

static double[] SortOtriz(double[] arr)

{

int poloz = 0;

int k = 0;

foreach (double x in arr)

{

if (x > 0)

k += 1;

}

poloz = k;

k = 0;

double[] ArrOfOtriz = new double[arr.Length - poloz];

k = 0;

foreach (double x in arr)

{

if (x < 0)

{

ArrOfOtriz[k] = x;

k += 1;

}

}

double[] ubuv = new double[ArrOfOtriz.Length];

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

ubuv[i] = ArrOfOtriz[i];

Array.Sort(ubuv);

Array.Reverse(ubuv);

Array.Sort(ArrOfOtriz);

double[] copy\_arr = new double[arr.Length];

Console.WriteLine("1. Отсортировать по возрастанию");

Console.WriteLine("2. Отсортировать по убыванию");

Console.WriteLine("0. Назад");

string otv = Console.ReadLine();

while (otv != "1" && otv != "2" && otv != "0")

{

Console.WriteLine("Введите целое число от 0 до 2");

otv = Console.ReadLine();

}

Console.Clear();

k = 0;

if (otv == "1")

{

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

{

if (arr[i] < 0)

{

copy\_arr[i] = ArrOfOtriz[k];

k++;

}

else

{

copy\_arr[i] = arr[i];

}

}

}

else if (otv == "2")

{

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

{

if (arr[i] < 0)

{

copy\_arr[i] = ubuv[k];

k++;

}

else

{

copy\_arr[i] = arr[i];

}

}

}

else

copy\_arr = arr;

return copy\_arr;

}

static void keywords(string phrase, string[] KeyWords)

{

while (phrase.IndexOf(" ") != -1)

phrase = phrase.Replace(" ", " ");

phrase += " ";

string ch = "";

int k = 0;

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

{

if (phrase[i] == ' ' && i == 0 && i != phrase.Length - 1)

continue;

else if (phrase[i] != ' ' && (phrase[i] == ',' | phrase[i] == ';' | phrase[i] == ':') && i != 0)

{

if (phrase[i + 1] == ' ')

{

ch += phrase[i];

continue;

}

else

{

if (phrase[i-1] == ' ')

{

ch = "";

}

foreach (string x in KeyWords)

{

if (x == ch.Substring(0, ch.Length))

{

k += 1;

ch = "";

break;

}

}

ch = "";

}

}

else if (phrase[i] != ' ' && phrase[i] != ',' && phrase[i] != ';' && phrase[i] != ':')

ch += phrase[i];

else

{

if (ch.Length == 0)

continue;

else if (ch[ch.Length - 1] != ',' && ch[ch.Length - 1] != ';' && ch[ch.Length - 1] != ':' &&

ch[ch.Length - 1] != '!' && ch[ch.Length - 1] != '?' && ch[ch.Length - 1] != '.')

{

foreach (string x in KeyWords)

{

if (x == ch)

{

k += 1;

ch = "";

break;

}

}

ch = "";

}

else

{

foreach (string x in KeyWords)

{

if (x == ch.Substring(0, ch.Length - 1))

{

k += 1;

ch = "";

break;

}

}

}

ch = "";

}

}

if (k == 0)

Console.WriteLine("В строке нет ключевых слов C#");

else

Console.WriteLine($"Найдено ключевых слов C# - {k}");

}

static string MenuStart()

{

Console.WriteLine(@"1. Задача номер 1 (массивы)

2. Задача номер 2 (строки)

0. Выход из программы");

string bug = Console.ReadLine();

while (bug != "0" && bug != "1" && bug != "2")

{

Console.WriteLine("\nВведите целое число от 0 до 2!");

bug = Console.ReadLine();

}

return bug;

}

static string MenuFirstExercise()

{

Console.WriteLine(@"1. Сформировать одномерный массив типа Double с помощью ДСЧ

2. Сформировать одномерный массив типа Double с клавиатуры

3. Распечатать массив

4. Отсортировать только отрицательные элементы массива

0. Назад");

string bug = Console.ReadLine();

while (bug != "0" && bug != "1" && bug != "2" && bug != "3" && bug != "4")

{

Console.WriteLine("\nВведите целое число от 0 до 4!");

bug = Console.ReadLine();

}

return bug;

}

static string MenuSecondExercise()

{

Console.WriteLine(@"1. Формирование случайной строки

2. Формирование строки с клавиатуры

3. Распечатать строку

4. Проверка на содержание ключевых слов в строке

0. Назад");

string bug = Console.ReadLine();

while (bug != "0" && bug != "1" && bug != "2" && bug != "3" && bug != "4")

{

Console.WriteLine("\nВведите целое число от 0 до 4!");

bug = Console.ReadLine();

}

return bug;

}

static void Main(string[] args)

{

double[] arr = null;

string[] stroki\_random = { "Но, боже мой, какая скука, с больным сидеть и день и ночь, не отходя ни шагу прочь!",

"Так думал молодой повеса, летя в пыли на почтовых, всевышней волею Зевеса наследник всех\nсвоих родных. Друзья Людмилы и Руслана!",

"Мог изъясняться и писал; continue легко мазурку танцевал decimal и кланялся непринужденно;\nчего ж вам больше fixed?",

"Как томно был он молчалив, как пламенно красноречив, в сердечных письмах public как небрежен!",

"Одним д set ыша, одно любя, как он умел ,sizeof, забыть себя!",

"Как взор его был быстр и нежен, стыдлив и дерзок, а порой блистал послушною слезой!",

"И недоверчивый старик, и рогоносец величавый, всегда довольный сам собой, своим обедом и женой.",

"И там;void, гуляет на просторе, пока недремлющий брегет не прозвонит ему обед.",

"Французской, virtual кухни лучший цвет, и Страсбурга пи, ulong рог нетленный меж сыром лимбургским\nжи while, вым и ананасом золотым.",

"Там и Дидло virtual ,венчался славой, там, там под сению кулис младые дни мои неслись.",

"И, устремив на; ;uint; ; ; ; ; ;ч;уждый свет разочарованный лорнет, веселья зритель\nравнодушный,throw безмолвно буду я зевать и о былом воспоминать?",

"И вдруг прыжок unsafe, и вдруг летит, летит, как пух от уст Эола.",

"То стан совьет, то разовьет и быстрой ножкой ножку бьет.",

"Еще амуры, черти, зм value еи на сцене скачут и шумят.",

"Сморкаться, кашлять, шикать, хлопать.",

"А уж Онегин вышел вон; домой одеться едет ulong о,return ,н." };

string[] KeyWords = {"abstract", "as", "base", "bool", "break", "byte", "case", "catch", "char", "checked", "class", "const", "continue", "decimal", "default", "delegate", "do",

"double", "else", "enum", "event", "explicit", "extern", "false", "finally", "fixed", "float", "for", "foreach", "goto", "if", "implicit", "in", "int", "interface", "internal",

"is", "lock", "long", "namespace", "new", "null", "object", "operator", "out", "override", "params", "private", "protected", "public", "readonly", "ref", "return", "sbyte",

"sealed", "short", "sizeof", "stackalloc", "static", "string", "struct", "switch", "this", "throw", "true", "try", "typeof", "uint", "ulong", "unchecked", "unsafe", "ushort",

"using", "virtual", "void", "volatile", "while"};

int ll = 0; int ll1 = 0; int ll2 = 0; bool massive = false; string stroka = ""; Random rand = new(); bool strok = false;

do

{

string Menu = MenuStart();

switch (Menu)

{

case "0":

ll = -4;

Console.Clear();

Console.WriteLine("Благодарим за использование программы!");

break;

case "1":

ll1 = 0;

do

{

Console.Clear();

string Menu1 = MenuFirstExercise();

switch (Menu1)

{

case "0":

Console.Clear();

ll1 = -4;

break;

case "1":

Console.Clear();

massive = true;

MakeRandomMas(ref arr);

Console.WriteLine("\nНажмите любую клавишу, чтобы продолжить");

Console.ReadKey();

break;

case "2":

Console.Clear();

massive = true;

MakeManualMas(ref arr);

Console.WriteLine("\nНажмите любую клавишу, чтобы продолжить");

Console.ReadKey();

break;

case "3":

if (massive)

{

Console.Clear();

PrintMas(arr, msg: "Массив представлен ниже");

Console.WriteLine("Нажмите любую клавишу, чтобы продолжить");

Console.ReadKey();

}

else

{

Console.Clear();

Console.WriteLine("Сначала нужно сформировать массив!");

Console.WriteLine("\nНажмите любую клавишу, чтобы продолжить");

Console.ReadKey();

}

break;

case "4":

if (massive)

{

Console.Clear();

double[] arr\_copy = arr;

arr = SortOtriz(arr);

if (arr == arr\_copy)

break;

PrintMas(arr, msg: "Полученный массив представлен ниже:");

PrintMas(arr\_copy, msg: "Исходный массив:");

Console.WriteLine("Нажмите любую клавишу, чтобы продолжить");

Console.ReadKey();

}

else

{

Console.Clear();

Console.WriteLine("Сначала нужно сформировать массив!");

Console.WriteLine("\nНажмите любую клавишу, чтобы продолжить");

Console.ReadKey();

}

break;

}

} while (ll1 != -4);

break;

case "2":

ll2 = 0;

do

{

Console.Clear();

string Menu1 = MenuSecondExercise();

switch (Menu1)

{

case "0":

Console.Clear();

ll2 = -4;

break;

case "1":

Console.Clear();

strok = true;

stroka = stroki\_random[rand.Next(0, stroki\_random.Length)];

Console.WriteLine("Строка сформирована");

Console.WriteLine("\nНажмите любую клавишу, чтобы продолжить");

Console.ReadKey();

break;

case "2":

Console.Clear();

strok = true;

Console.WriteLine("Введите строку");

stroka = Console.ReadLine();

while(stroka.Length == 0)

{

Console.WriteLine("\nСтрока пуста. Дальнейшие действия не имеют смысла. Введите строку заново");

stroka = Console.ReadLine();

}

Console.WriteLine("\nСтрока сформирована");

Console.WriteLine("\nНажмите любую клавишу, чтобы продолжить");

Console.ReadKey();

break;

case "3":

if (strok)

{

Console.Clear();

Console.WriteLine($"Строка представлена ниже:\n\n{stroka}");

Console.WriteLine("\nНажмите любую клавишу, чтобы продолжить");

Console.ReadKey();

}

else

{

Console.Clear();

Console.WriteLine("Сначало нужно сформировать строку!");

Console.WriteLine("\nНажмите любую клавишу, чтобы продолжить");

Console.ReadKey();

}

break;

case "4":

if (strok)

{

Console.Clear();

keywords(stroka, KeyWords);

Console.WriteLine("\nНажмите любую клавишу, чтобы продолжить");

Console.ReadKey();

}

else

{

Console.Clear();

Console.WriteLine("Сначала нужно сформировать строку!");

Console.WriteLine("\nНажмите любую клавишу, чтобы продолжить");

Console.ReadKey();

}

break;

}

} while (ll2 != -4);

break;

}

}

while (ll != -4);

}

}

}