1 Тест-требования

Для создаваемого приложения были разработаны следующие тест-требования:

Проверка поведения работоспособности программы:

1. Проверка поведения программы при введении отрицательного значения размера массива.
2. Проверка поведения программы при вводе текста в значение размера массива.
3. Проверка поведения программы при введении нулевого значения размера массива.
4. Проверка поведения программы при введении положительного значения и заполнение всех параметров.
5. Проверка вывода результата в файл.
6. Проверка отсортированного массива по убыванию.

2 Разработка тест плана

Тестовый пример 1.

Номер тест требования 1.

Описание теста: проверка ввода отрицательного значения.

Входные данные:

* «Размер массива» – «-3».

Ожидаемые выходные данные:

* Сообщение об ошибке «Введите число больше 0!».

Сценарий:

* Запустить программу.
* Ввести размер массива «-3».
* Нажать кнопку Enter.

Тестовый пример 2.

Номер тест требования 2.

Описание теста: проверка ввода текста в значение размера массива.

Входные данные:

* «Размер массива» – «Карина».

Ожидаемые выходные данные:

* Сообщение об ошибке «Введите число!».

Сценарий:

* Запустить программу.
* Ввести размер массива «Карина».
* Нажать кнопку Enter.

Тестовый пример 3.

Требования: 3.

Описание теста: проверка ввода нуля в значение размера массива.

Входные данные:

* «Размер массива» – «0».

Ожидаемые выходные данные:

* Сообщение об ошибке «Введите число больше 0!».

Сценарий:

* Запустить программу.
* Ввести размер массива «0».
* Нажать кнопку Enter.

Тестовый пример 4.

Требования: 4.

1. Описание теста: в данном тесте проверяется работоспособность программы при вводе положительного значения размера массива и заполнение всех параметров.

Входные данные:

* «Размер массива» – «3».
* «Наименование» – «Мы из будущего».
* «Автор» – «Брунов».
* «Жанр» – «Боевик».
* «Наименование» – «5 сантиметров в секунду».
* Автор» – «Макото».
* «Жанр» – «Аниме».
* «Наименование» – «Книга Генри».
* Автор» – «Осл».
* «Жанр» – «Драма».

Ожидаемые выходные данные:

* Сообщение «Заполнение записи №».
* Сообщение «Наименование:».
* Сообщение «Автор:».
* Сообщение «Жанр:».

Сценарий:

* Запустить программу.
* Ввести размер массива «3».
* Нажать кнопку Enter.
* Заполнить в соответствии с входными данными.
* Нажать кнопку Enter.

Тестовый пример 5.

Требования: 5.

Описание теста: в данном тесте проверяется работоспособность вывода данных в файл.

Входные данные:

* «Размер массива» – «3».
* «Наименование» – «Мы из будущего».
* «Автор» – «Брунов».
* «Жанр» – «Боевик».
* «Наименование» – «5 сантиметров в секунду».
* Автор» – «Макото».
* «Жанр» – «Аниме».
* «Наименование» – «Книга Генри».
* Автор» – «Осл».
* «Жанр» – «Драма».

Ожидаемые выходные данные:

* Записи в файле test.txt.

Сценарий:

* Запустить программу.
* Ввести размер массива «3».
* Нажать кнопку Enter.
* Заполнить в соответствии с входными данными.
* Нажать кнопку Enter.
* Открыть файл test.txt по пути Book\bin\Debug\netcoreapp3.1\.

Тестовый пример 6.

Требования: 6.

Описание теста: в данном тесте проверяется работоспособность сортировки по убыванию.

Входные данные:

* «Размер массива» – «3».
* «Наименование» – «».
* «Автор» – «».
* «Жанр» – «».
* «Наименование» – «».
* Автор» – «».
* «Жанр» – «».
* «Наименование» – «».
* Автор» – «».
* «Жанр» – «».

Ожидаемые выходные данные:

* Отсортированные записи в файле test.txt.

Сценарий:

* Запустить программу.
* Ввести размер массива «3».
* Нажать кнопку Enter.
* Заполнить в соответствии с входными данными.
* Нажать кнопку Enter.
* Открыть файл test.txt по пути Book\bin\Debug\netcoreapp3.1\.

3 Результат тестирования

Тестовый пример 1.

Входные данные:

* Запустить программу.
* Ввести размер массива «-3».
* Нажать кнопку Enter.

Полученные выходные данные:

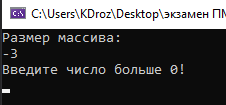


Рисунок 1 – Отрицательное значение

Вывод: соответствует ожидаемым выходным данным.

Тестовый пример 2.

Входные данные:

* Запустить программу.
* Ввести размер массива «Карина».
* Нажать кнопку Enter.

Полученные выходные данные:

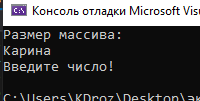


Рисунок 2 – Текстовое значение

Вывод: результат соответствует ожидаемым выходным данным.

Тестовый пример 3.

Входные данные:

* Запустить программу.
* Ввести размер массива «0».
* Нажать кнопку Enter.

Полученные выходные данные:

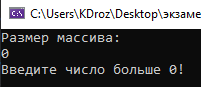


Рисунок 3 – Значение 0

Вывод: результат соответствует ожидаемым выходным данным.

Тестовый пример 4.

Входные данные:

* Запустить программу.
* Ввести размер массива «3».
* Нажать кнопку Enter.
* «Наименование» – «Мы из будущего».
* «Автор» – «Брунов».
* «Жанр» – «Боевик».
* «Наименование» – «5 сантиметров в секунду».
* Автор» – «Макото».
* «Жанр» – «Аниме».
* «Наименование» – «Книга Генри».
* Автор» – «Осл».
* «Жанр» – «Драма».
* Нажать кнопку Enter.

Полученные выходные данные:

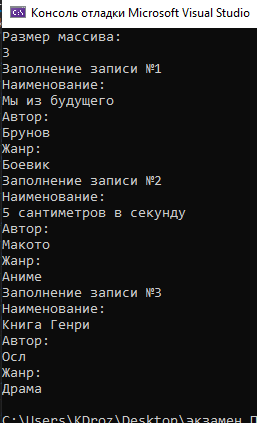


Рисунок 4 – Заполнение

Вывод: результат соответствует ожидаемым выходным данным.

Тестовый пример 5.

Входные данные:

* Запустить программу.
* Ввести размер массива «3».
* Нажать кнопку Enter.
* «Наименование» – «Мы из будущего».
* «Автор» – «Брунов».
* «Жанр» – «Боевик».
* «Наименование» – «5 сантиметров в секунду».
* Автор» – «Макото».
* «Жанр» – «Аниме».
* «Наименование» – «Книга Генри».
* Автор» – «Осл».
* «Жанр» – «Драма».
* Нажать кнопку Enter.
* Открыть файл test.txt по пути Book\bin\Debug\netcoreapp3.1\.

Полученные выходные данные:

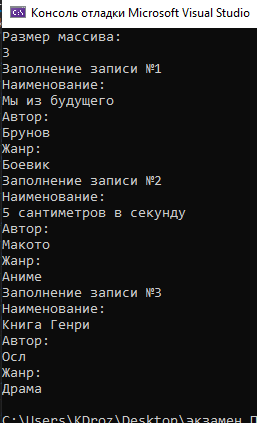


Рисунок 5 – Заполнение

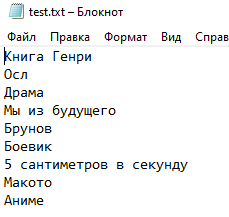


Рисунок 6 – Данные в файле

Вывод: результат соответствует ожидаемым выходным данным.

Тестовый пример 6.

Входные данные:

* Запустить программу.
* Ввести размер массива «3».
* Нажать кнопку Enter.
* Нажать кнопку Enter.
* Открыть файл test.txt по пути Book\bin\Debug\netcoreapp3.1\.

Полученные выходные данные:

Рисунок 7 – Заполнение

Рисунок 7 – Отсортированные данные

Вывод: результат соответствует ожидаемым выходным данным.