# ТЕСТИРОВАНИЕ

## **Тест-требования**

Для создаваемого приложения были созданы следующие тест-требования:

1. Выполнить проверку работоспособность полей для ввода
2. Выполнить проверку сохранения данных в файл
3. Проверить сортировку массива по возрастанию

## Тест-план

Тестовый пример 1.

Требования 1.

Сценарий:

1. Открыть приложение.
2. Ввести входные данные.

Входные данные:

Позитивные данные:

Название города: Moscow;

Количество жителей: 1500000;

Площадь: 300000.

Негативные данные:

Количество жителей: слово.

Ожидаемые выходные данные:

В случае с позитивными данными программа корректно восприняла данные.

В случае с негативными данными программа закроется.

Тестовый пример 2.

Требования 2.

Сценарий:

1. Открыть приложение.
2. Ввести количество студентов 2
3. Ввести входные данные два раза

Входные данные:

Название города: Moscow;

Количество жителей: 1500000;

Площадь: 300000.

Ожидаемые выходные данные:

В результате массив полностью заполнился, и программа не выдала ошибок.

Тестовый пример 3.

Требования 3.

Сценарий:

1. Открыть приложение
2. Ввести количество городов 2
3. Ввести входные данные 2 раза

Входные данные:

Входные данные №1

Название города: Moscow;

Количество жителей: 1500000;

Площадь: 300000.

Входные данные №2

Название города: Pokrov;

Количество жителей: 17345;

Площадь: 10000.

Ожидаемые выходные данные:

## На выходе выведется отсортированный массив.

## Результаты тестирования

Тест пример №1

Сценарий:

1. Открыть приложение.
2. Ввести входные данные.

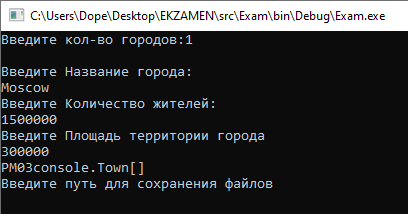


Рисунок 1 – Заполненные корректные данные

Тест пройден успешно.

Тест пример №2

Сценарий:

1. Открыть приложение.
2. Ввести количество городов 2
3. Ввести входные данные два раза

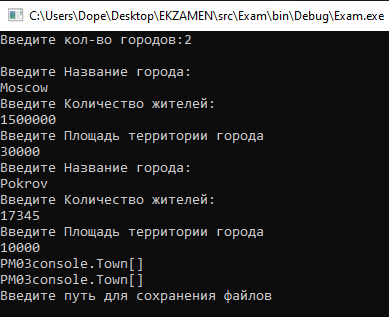


Рисунок 3 – Сохранение файла

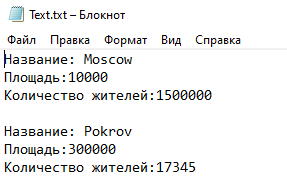


Рисунок 4 – Сохранённый файл

Тест пройден успешно.

Тест пример №3

Сценарий:

1. Открыть приложение
2. Ввести данные №1
3. Ввести данные №2

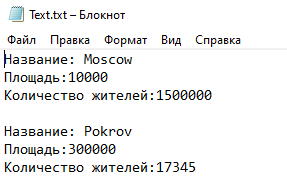


Рисунок 5 – Отсортированы массив

Тест пройден успешно.