# ТЕСТИРОВАНИЕ ВЕБ-ПРИЛОЖЕНИЯ

## Разработка тест-требований

Для разработанного консольного приложения были определены следующие тест-требования:

1. Убедиться, что поля массива заполняются введёнными значениями.
2. Проверить запись введённых значений в файл.
3. Протестировать корректную сортировку массива.
4. Убедиться, что массив правильно сортируется по «числу остановок».
5. Удостовериться, что массив правильно сортируется по «времени в пути».
6. Посмотреть, что при вводе данных нельзя ввести число равное нулю.
7. Протестировать ввод отрицательного числа.

## Тест-план

По тест требованиям был разработан тест-план.

Тест 1. Убедиться, что поля массива заполняются введёнными значениями.

Требования: 1.1.

Описание теста: в данном тесте проверяется, что в полях массива будут содержаться значения, которые ввёл пользователь.

Входные данные:

1. Пользовательские значения, введённые в процессе работы консольного приложения.

Ожидаемые выходные данные: пользовательские значения, записанные в поля массива.

Сценарий теста:

1. Запустить консольное приложение;
2. Последовательно вводить требуемые данные.

Тест 2. Проверить запись введённых значений в файл.

Требования 2.1.

Описание теста: в данном тесте проверяется, что введённые в процессе выполнения программы данные будут записаны в текстовый файл «Route.txt».

Входные данные:

1. Пользовательские значения, введённые в процессе работы консольного приложения.

Ожидаемые выходные данные: текстовый файл «Route.txt», заполненный данными, введёнными пользователем.

Сценарий теста:

1. Запустить консольное приложение.
2. Последовательно вводить требуемые данные.
3. После завершения работы программы открыть текстовый файл «Route.txt».

Тест 3. Протестировать корректную сортировку массива.

Тест 3.1. Убедиться, что массив правильно сортируется по «числу остановок».

Требования 3.1.

Описание теста: в данном тесте проверяется, что данные введённые пользователем в пункт «число остановок» будут корректно отсортированы по возрастанию.

Входные данные:

1. Данные, введённые пользователем в процессе выполнения программы.

Ожидаемые выходные данные: корректно отсортированный массив с пользовательскими данными.

Сценарий теста:

1. Запустить консольное приложение.
2. Последовательно ввести требуемые данные.

Тест 3.2. Удостовериться, что массив правильно сортируется по «времени в пути».

Требования 3.2.

Описание теста: в данном тесте проверяется, что данные введённые пользователем в пункт «время в пути» будут корректно отсортированы по возрастанию.

Входные данные:

1. Данные, введённые пользователем в процессе выполнения программы.

Ожидаемые выходные данные: корректно отсортированный массив с пользовательскими данными.

Сценарий теста:

1. Запустить консольное приложение.
2. Последовательно ввести требуемые данные.

Тест 4. Посмотреть, что при вводе данных нельзя ввести отрицательное число или число, равное нулю.

Требования 4.1.

Описание теста: в данном тесте проверяется, что при вводе числа меньше нуля или равному нулю, программа не вызовет исключение.

Входные данные:

1. Данные, введённые пользователем;

Ожидаемые выходные данные: некорректные данные в виде числа равного нулю.

Сценарий теста:

1. Запустить консольное приложение;
2. Ввести 0.

Тест 5. Протестировать ввод букв.

Требования 5.1.

Описание теста: в данном тесте проверяется ввод букв при вводе данных.

Входные данные:

1. Данные, введённые пользователем.

Ожидаемые выходные данные: некорректные данные в виде отрицательного числа или числа, равного нулю.

Сценарий теста:

1. Запустить консольное приложение.
2. Ввести отрицательное число.

## Результаты тестирования

Тест 1. Убедиться, что поля массива заполняются введёнными значениями.

Согласно сценарию теста, было запущено консольное приложение и введены требуемые данные. Результат тестирования показан на рисунке 1.1.

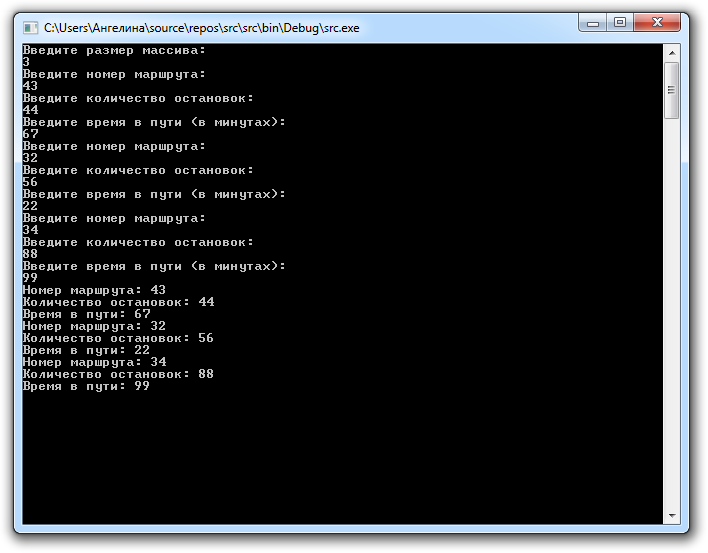


Рисунок 1.1 – Результат заполнения массива

Вывод: полученный результат теста совпадает с ожидаемым результатом.

Тест 2. Проверить запись введённых значений в файл.

Сначала было запущено консольное приложение, введены необходимые данные. Затем после завершения работы программы был открыт текстовый файл «Route.txt». Результат можно увидеть на рисунке 1.2.

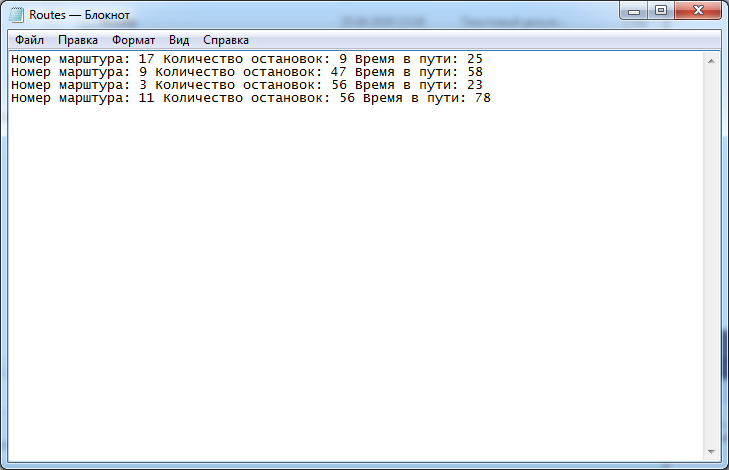


Рисунок 1.2 – Данные, записанные в файл в результате выполнения программы

Вывод: таким образом, можно увидеть, что результат соответствует ожиданиям.

Тест 3. Протестировать корректную сортировку массива.

Тест 3.1. Убедиться, что массив правильно сортируется по пункту «числу остановок».

Сначала было запущено консольное приложение и введены необходимые данные, затем был открыт файл для проверки результатов работы программы. Результат тестирования показан на рисунке 1.3.

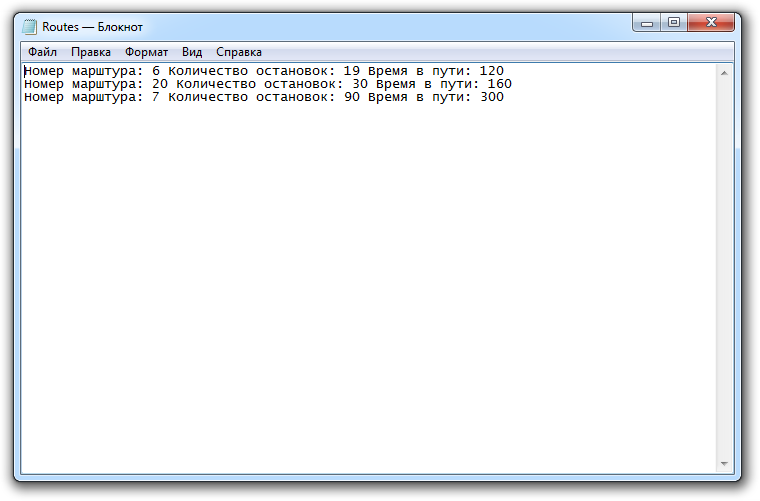


Рисунок 1.3 – Результат сортировки данных, введённых пользователем

В результате тестирования выяснилось, что полученные результаты совпали с ожидаемыми.

Тест 3.2. Удостовериться в том, что массив правильно сортируется по пункту «время в пути».

Сначала было запущено консольное приложение и введены необходимые данные, затем был открыт файл для проверки результатов работы программы. Результат действий отображён на рисунке 1.4.

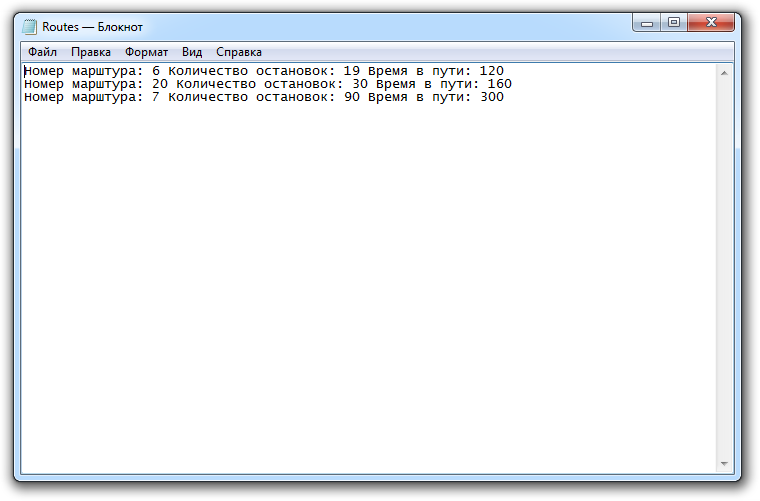


Рисунок 1.4 – Результат сортировки массива по пункту «время в пути»

Вывод: в ходе тестирования итоговые результаты оправдали ожидаемые.

Тест 4. Посмотреть, что при вводе данных нельзя ввести отрицательное число или число, равное нулю.

Согласно сценарию теста, было запущено консольное приложение. Затем при вводе данных был введён ноль. Результат показан на рисунке 1.5.

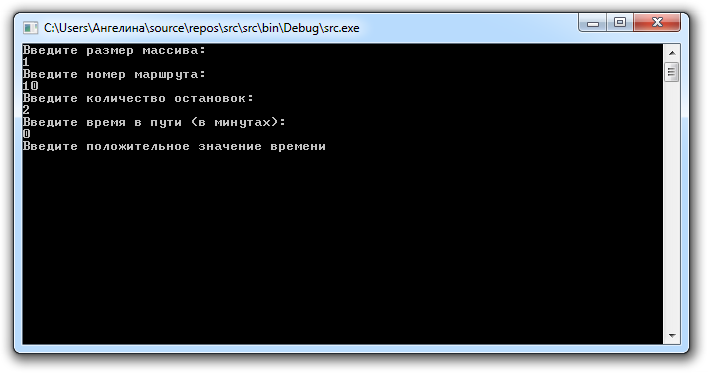


Рисунок 1.5 – Результат ввода некорректного числа

Тест 5. Посмотреть, что при вводе данных нельзя ввести отрицательное число.

Согласно сценарию теста, было запущено консольное приложение. Затем при вводе данных было введено отрицательное число. Результат показан на рисунке 1.6.

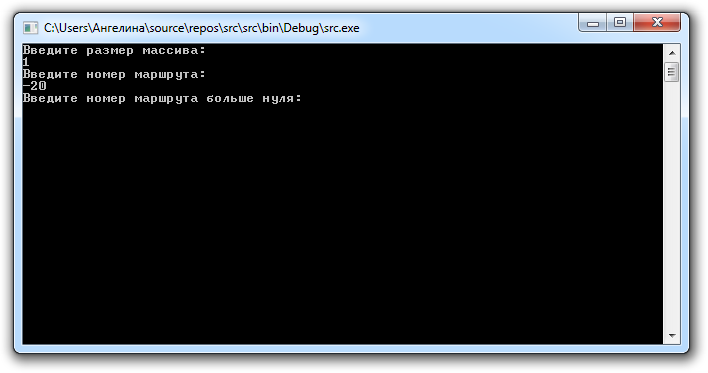


Рисунок 1.6 – Результат ввода отрицательного числа

Вывод: можно сделать вывод, что ожидаемые результаты подтверждаются.