**Тест-план:**

Тестовый пример: Корректный ввод информации о маршруте

Тест-требование: тест-требование 1

Описание: Тест проверяет, что при корректном вводе данных о маршруте, маршрут успешно сохраняется в массив.

Входные параметры: Количество добавляемых маршрутов – «1», номер маршрута – «152T», количество остановок – «5», время в пути – «145».

Ожидаемый результат: После ввода всех данных, программа уведомит пользователя об успешном добавлении маршрута.

Шаги выполнения:

1. Запуск программы.
2. В строке для ввода количества маршрутов ввод «1». Нажатие клавиши Enter.
3. В строке для ввода номера маршрута ввод «152T». Нажатие клавиши Enter.
4. В строке для ввода количества остановок ввод «5». Нажатие клавиши Enter.
5. В строке для ввода времени у пути ввод «145». Нажатие клавиши Enter.
6. Проверка наличия сообщения об успешном добавлении маршрута.

Тестовый пример: Некорректный ввод информации о маршруте 1

Тест-требование: тест-требование 1

Описание: Тест проверяет, что при некорректном вводе номера маршрута, выводится сообщение об ошибке и производится повторный ввод.

Входные параметры: – Количество добавляемых маршрутов – «1».

Ожидаемый результат: После ввода пустой строки в строке ввода номера, программа уведомит пользователя о том, что номер введен некорректно и предоставит возможность ввести номер еще раз.

Шаги выполнения:

1. Запуск программы.
2. В строке для ввода количества маршрутов нажатие клавиши Enter.
3. Проверка наличия сообщения о некорректно введенном номере. Нажатие клавиши Enter.
4. Проверка возможности повторного ввода номера.

Тестовый пример: Некорректный ввод информации о маршруте 2

Тест-требование: тест-требование 1

Описание: Тест проверяет, что при некорректном вводе количества остановок маршрута, выводится сообщение об ошибке и производится повторный ввод.

Входные параметры: Количество добавляемых маршрутов – «1», номер маршрута – «152T», количество остановок – «пять».

Ожидаемый результат: После ввода строкового значения в строке ввода количества остановок, программа уведомит пользователя о том, что количество введено некорректно и предоставит возможность ввести количество еще раз.

Шаги выполнения:

1. Запуск программы.
2. В строке для ввода количества маршрутов ввод «1». Нажатие клавиши Enter.
3. В строке для ввода номера маршрута ввод «152T». Нажатие клавиши Enter.
4. В строке для ввода количества остановок ввод «пять». Нажатие клавиши Enter.
5. Проверка наличия сообщения о некорректно введенном количестве. Нажатие клавиши Enter.
6. Проверка возможности повторного ввода количества.

Тестовый пример: Некорректный ввод информации о маршруте 3

Тест-требование: тест-требование 1

Описание: Тест проверяет, что при некорректном вводе времени в пути маршрута, выводится сообщение об ошибке и производится повторный ввод.

Входные параметры: Количество добавляемых маршрутов – 1, номер маршрута – «152T», количество остановок – «5», время в пути – «пол часа».

Ожидаемый результат: После ввода строкового значения в строке ввода времени в пути, программа уведомит пользователя о том, что время введено некорректно и предоставит возможность ввести время еще раз.

Шаги выполнения:

1. Запуск программы.
2. В строке для ввода количества маршрутов ввод «1». Нажатие клавиши Enter.
3. В строке для ввода номера маршрута ввод «152T». Нажатие клавиши Enter.
4. В строке для ввода количества остановок ввод «5». Нажатие клавиши Enter.
5. В строке для ввода времени в пути ввод «пол часа». Нажатие клавиши Enter.
6. Проверка наличия сообщения о некорректно введенном количестве. Нажатие клавиши Enter.
7. Проверка возможности повторного ввода количества.

Тестовый пример: Проверка сортировки списка маршрутов

Тест-требование: тест-требование 2

Описание: Тест проверяет, что список маршрутов сортируется по возрастанию по сочетанию значений «число остановок» и «время в пути».

Входные параметры: Количество добавляемых маршрутов – «3», номер маршрута 1 – «152T», количество остановок 1– «5», время в пути 1– «145», номер маршрута 2 – «46T», количество остановок 2 – «10», время в пути 2– «120», номер маршрута 1 – «11T», количество остановок 1– «6», время в пути 1– «163».

Ожидаемый результат: Список маршрутов корректно от сортируется и выведется на экран.

Шаги выполнения:

1. Запуск программы.
2. В строке для ввода количества маршрутов ввод «3». Нажатие клавиши Enter.
3. Заполнение массива маршрутов данными.
4. Проверка наличия сообщения об успешном добавлении всех маршрутов
5. Проверка вывода отсортированного списка маршрутов.

Тестовый пример: Записи в файл отсортированного списка маршрутов

Тест-требование: тест-требование 3

Описание: Тест проверяет, что после успешной сортировки массива он сохраняется в файл формата json.

Входные параметры: Количество добавляемых маршрутов – «3», номер маршрута 1 – «152T», количество остановок 1– «5», время в пути 1– «145», номер маршрута 2 – «46T», количество остановок 2– «5», время в пути 2– «120», номер маршрута 1 – «11T», количество остановок 1– «10», время в пути 1– «163».

Ожидаемый результат: Список маршрутов сортируется и записывается в файл.

Шаги выполнения:

1. Запуск программы.
2. В строке для ввода количества маршрутов ввод «3». Нажатие клавиши Enter.
3. Заполнение массива маршрутов данными.
4. Проверка наличия сообщения об успешном добавлении всех маршрутов
5. Наличия списка отсортированных маршрутов.
6. Проверка наличия сообщения об успешном сохранении массива в файл.
7. Проверка содержимого файла.