

# 1. Позитивное тестирование

---

## 1.1 Тестирование ввода количества автомобилей

---

Тест-кейс TC-POS-01: Ввод корректного количества автомобилей

Предусловие: Программа запущена

Шаги:

При запросе количества автомобилей ввести "3"

Нажать Enter

Ожидаемый результат: Программа переходит к запросу данных об автомобилях

Фактический результат: Программа перешла к запросу данных об автомобилях

Статус: Пройден

## 1.2 Тестирование ввода данных автомобиля

---

Тест-кейс TC-POS-02: Ввод корректных данных автомобиля

Предусловие: Введено корректное количество автомобилей

Шаги:

Для модели ввести "Toyota Camry"

Для цены ввести "25000"

Для мощности ввести "180"

Ожидаемый результат: Данные принимаются, программа запрашивает следующий автомобиль

Фактический результат: Данные принимаются и программа запрашивает следующий автомобиль

Статус: Пройден

## 1.3 Тестирование сортировки

---

Тест-кейс TC-POS-03: Проверка правильности сортировки

Предусловие: Введено 3 автомобиля с разными параметрами

Шаги:

Запустить сортировку

Проверить порядок вывода

Ожидаемый результат: Автомобили отсортированы сначала по убыванию мощности, затем по возрастанию цены

Фактический результат: Автомобили отсортированы успешно

Статус: Пройден

## 1.4 Тестирование сохранения в файл

---

Тест-кейс TC-POS-04: Корректное сохранение в файл

Предусловие: Данные введены и отсортированы

Шаги:

Ввести путь для сохранения "cars.txt"

Проверить содержимое файла

Ожидаемый результат: Файл создан, содержит корректные данные в правильном порядке

Фактический результат: Файл создан и содержит корректные данные в правильном порядке

Статус: Пройден

# 2. Негативное тестирование

---

## 2.1 Тестирование неверного ввода количества

---

Тест-кейс TC-NEG-01: Ввод отрицательного числа автомобилей

Предусловие: Программа запущена

Шаги:

При запросе количества ввести "-2"

Нажать Enter

Ожидаемый результат: Программа сообщает об ошибке и запрашивает повторный ввод

Фактический результат: Программа сообщает об ошибке и запрашивает повторный вво

Статус: Пройден

## 2.2 Тестирование нечислового ввода цены

---

Тест-кейс TC-NEG-02: Ввод текста вместо цены

Предусловие: В процессе ввода данных автомобиля

Шаги:

В поле цены ввести "двадцать тысяч"

Нажать Enter

Ожидаемый результат: Программа сообщает об ошибке и запрашивает повторный ввод

Фактический результат: Программа сообщает об ошибке и запрашивает повторный ввод

Статус: Пройден

## 2.3 Тестирование нулевой мощности

---

Тест-кейс TC-NEG-03: Ввод нулевого значения мощности

Предусловие: В процессе ввода данных автомобиля

Шаги:

В поле мощности ввести "0"

Нажать Enter

Ожидаемый результат: Программа сообщает об ошибке и запрашивает повторный ввод

Фактический результат: Программа сообщает об ошибке и запрашивает повторный ввод

Статус: Пройден

## 2.4 Тестирование неверного пути сохранения

---

Тест-кейс TC-NEG-04: Попытка сохранения в несуществующую директорию

Предусловие: Данные готовы к сохранению

Шаги:

Ввести путь "/несуществующая\_папка/cars.txt"

Нажать Enter

Ожидаемый результат: Программа выводит сообщение об ошибке записи

Фактический результат: Программа выводит сообщение об ошибке записи

Статус: Пройден

# 3. Интеграционное тестирование

---

## 3.1 Полный сценарий работы

---

Тест-кейс TC-INT-01: Полный цикл от ввода до сохранения

Предусловие: Программа запущена

Шаги:

Ввести количество автомобилей: 2

Ввести данные первого авто: "BMW X5", 50000, 300

Ввести данные второго авто: "Audi A4", 35000, 250

Запустить сортировку

Сохранить в "test\_output.txt"

Ожидаемый результат: Все этапы выполнены без ошибок, файл содержит корректные отсортированные данные

Фактический результат: Все этапы выполнены без ошибок, файл содержит корректные отсортированные данные

Статус: Пройден