**Тест требования**

Позитивное тестирование

**Тест Create** Auto**Test**

Входные данные:

* Auto [] \_Autos, заполнен данными;
* Auto Auto .

Ожидаемые выходные данные:

* Auto не равен null;
* Auto .Length = 4.

Требования: объект Auto должен быть успешно создан, и количество элементов в нём должно быть равно 4.

**Тест Sort**Auto**Test**

Входные данные:

* Auto [] unsortedArray, заполнен данными;
* Auto [] sortedArray, заполнен данными, отсортирован по возрастанию;
* Auto Auto .

Ожидаемые выходные данные:

* Auto отсортирован и равен sortedArray.

Требования: объект Auto должен быть успешно создан и корректно отсортирован.

**Тест Save**Auto**Test**

Входные данные:

* Auto [] \_Autos, заполнен данными.
* Auto Auto .

Ожидаемые выходные данные:

* в директории тестового проекта существует файл Autos.txt.

Требования: объект Auto должен быть успешно создан, корректно отсортирован и сохранён в файл.

Негативное тестирование

**Тестовый пример 4**

Входные данные:

* int Legth

Ожидаемые выходные данные:

* программа завершает свою работу с сообщением о вводе некорректных данных.

Требования: программа должна завершить свою работу с сообщением о вводе некорректных данных.

**Тест план**

Позитивное тестирование

**Тест CreateAutoTest**

Описание: в этом тесте проверяется корректность создания объекта класса Auto с помощью массива объектов класса Driver.

Входные данные:

* Auto[] \_Autos, заполнен данными;
* Auto Auto.

Ожидаемые выходные данные:

* Auto не равен null;
* Auto.Length = 4.

Сценарий теста:

* создание объекта класса Auto и его заполнение с помощью готовых данные;
* проверка корректности создания объекта.

**Тест Sort** Auto

Описание: в этом тесте проверяется корректность работы метода сортировки в классе Auto.

Входные данные:

* Auto[] unsortedArray, заполнен данными;
* Auto[] sortedArray, заполнен данными, отсортирован по возрастанию;
* Auto Auto.

Ожидаемые выходные данные:

* Auto отсортирован и равен sortedArray.

Сценарий теста:

* создание объекта класса Auto и его заполнение с помощью готовых данные;
* сортировка объекта;
* проверка объекта на корректность сортировки.

**Тест Save**Auto**Test**

Описание: в этом тесте проверяется корректность работы метода сохранения в файл в классе Auto.

Входные данные:

* Auto[] \_ Autos, заполнен данными.
* Auto Auto.

Ожидаемые выходные данные:

* в директории тестового проекта существует файл Autos.txt.

Сценарий теста:

* создание объекта класса Auto и его заполнение с помощью готовых данные;
* сортировка объекта;
* сохранение объекта в файл;
* проверка файла на существование.

Негативное тестирование

**Тестовый пример 4**

Описание: в этом тесте проверяется поведение программы на ввод некорректных данных.

Входные данные:

* int Legth

Ожидаемые выходные данные:

* программа завершает свою работу с сообщением о вводе некорректных данных.

Сценарий теста:

* запуск программы;
* ввод в числовое поле не числа.