К программе представлены следующие требования по проверке:

1. Проверка ввода количества учебных дисциплин

1.1 При вводе некорректного количества, пользователю показывается ошибка

* 1. При вводе корректного количества, пользователь переходит к следующему шагу

1. Проверка ввода характеристик объекта

2.1 При вводе некорректного значения поля «семестр» пользователю показывается ошибка

1. Проверка сортировки массива

3.1 При вводе корректных значений учебных дисциплин производится сортировка массива согласно заданному условию.

1. Проверка записи результата в файл

4.1 При вводе корректных значений учебных дисциплин результат сортировки выводится в файл output.txt.

Тест-план

Тестовый пример 1.

Номер тест требования 1.1

Описание теста: в данном тесте проверяется некорректный ввод количества учебных дисциплин.

Входные данные:

1. Количество учебных дисциплин – -2

Ожидаемые выходные данные: при вводе некорректного количества учебных дисциплин, пользователю будет показана ошибка.

Сценарий:

1. Запустить приложение
2. Ввести количество учебных дисциплин.
3. Получить соответствующее сообщение об ошибке

Тестовый пример 2.

Номер тест требования 1.2

Описание теста: в данном тесте проверяется корректный ввод количества учебных дисциплин

Входные данные:

1. Количество учебных дисциплин - 3

Ожидаемые выходные данные: при вводе количества учебных дисциплин, пользователь переходит на следующий шаг (ввод характеристик)

Сценарий:

1. Запустить приложение
2. Ввести количество учебных дисциплин.
3. Перейти на следующий шаг (ввод характеристик дисциплин)

Тестовый пример 3.

Номер тест требования 2.1

Описание теста: в данном тесте проверяется некорректный ввод значения поля «семестр».

Входные данные:

1. Количество учебных дисциплин – -2
2. Наименование учебной дисциплины – математика
3. Семестр - abc

Ожидаемые выходные данные: при вводе некорректного значения поля «семестр», пользователю будет показана ошибка.

Сценарий:

1. Запустить приложение
2. Ввести количество учебных дисциплин.
3. Ввести наименование учебной дисциплины
4. Ввести значение поля «семестр»
5. Получить соответствующее сообщение об ошибке

Тестовый пример 4.

Номер тест требования 3.1

Описание теста: в данном тесте проверяется сортировка массива по условию: возрастание по сочетанию двух свойств в порядке убывания приоритета: «форма аттестации» и «название»

Входные данные:

1. Количество учебных дисциплин – 3
2. Дисциплины:

* Математика, 3, итоговая
* Русский, 2, итоговая
* Английский, 3, итоговая

Ожидаемые выходные данные: сортировка массива происходит согласно условию.

Сценарий:

1. Запустить приложение
2. Ввести количество учебных дисциплин.
3. Поэтапно ввести дисциплины.
4. Посмотреть результат сортировки и сравнить его с заданным условием.

Тестовый пример 5.

Номер тест требования 4.1

Описание теста: в данном тесте проверяется отображение результата сортировки в файле output.txt.

Входные данные:

1. Количество учебных дисциплин – 3
2. Дисциплины:

* Математика, 3, итоговая
* Русский, 2, итоговая
* Английский, 3, итоговая

Ожидаемые выходные данные: результат сортировки отображается в файле без ошибок.

Сценарий:

1. Запустить приложение
2. Ввести количество учебных дисциплин.
3. Поэтапно ввести дисциплины.
4. Посмотреть результат работы программы в файле output.txt в корне проекта.