

ТЕСТИРОВАНИЕ

Тест-требования

К разработанному приложению были предъявлены следующие тест требования:

1 Протестировать ввод количества элементов массива.

1.1 Проверить, что при вводе отрицательного/буквенного значения программа выдает уведомление о том, что введено некорректное значение и просит ввести корректное.

1.2 Проверить, что при вводе целого положительного значения программа продолжит свою работу и создаст массив из этого значения.

2 Протестировать заполнение массива.

2.1 Проверить, что при вводе корректных данных программа продолжит работать и запишет все данные в массив.

2.2 Проверить, что при вводе некорректных данных программа попросит повторно ввести значения или оповестит сообщением об ошибке ввода.

3 Протестировать запись данных в файл.

Проверить, что при вводе всех корректных данных они запишутся в текстовый файл.

4 Протестировать сортировку массива.

Проверить, что при вводе всех корректных данных и при их записи в текстовый документ, данные отобразятся в отсортированном порядке.

Тест-план

На основе данных тест-требований был разработан тест-план, состоящий из 4 тестов.

Тестовый пример 1.

Требования: 1.

Описание теста: протестировать функцию ввода длины массива путем введения правильных и неправильных данных.

Входные данные:

1.1 Количество элементов массива: 2.

1.2 Количество элементов массива: qwe.

Ожидаемые выходные данные:

1.1 Продолжение работы программы.

1.2 Вывод сообщения об ошибке, повторный ввод значения.

Сценарий 1.1:

1) Запускаем программу.

2) При запросе введения количества элементов ввести 2.

Сценарий 1.2:

1) Запускаем программу.

2) При запросе введения количества элементов ввести «qwe».

Тестовый пример 2.

Требования: 2.

Описание теста: протестировать заполнение массива после ввода его длины.

Входные данные:

2.1 Количество элементов массива: 2, название: BMW, модель: X6, цена: 2000000, название: KIA, модель: RIO, цена: 890000.

2.2 Количество элементов массива: 1, марка: (пустое значение), модель: RIO, цена: ghb.

Ожидаемые выходные данные:

2.1 Продолжение работы программы.

2.2 Повторный ввод на пустое значение и сообщение о некорректном вводе числа, повторный ввод.

Сценарий 2.1:

- 1) Запускаем программу.
- 2) При запросе введения количества элементов ввести 2.
- 3) При запросе введения название: BMW.
- 4) При запросе введения модель X6.
- 5) При запросе введения цена: 2000000.
- 6) При запросе введения название: KIA.
- 7) При запросе введения модель RIO.
- 8) При запросе введения цена: 890000.

Сценарий 2.2:

- 1) Запускаем программу.
- 2) При запросе введения количества элементов ввести 1
- 3) При запросе введения название: (пустое значение).
- 4) При запросе введения модель RIO.
- 5) При запросе введения цена: ghb.

Тестовый пример 3.

Требования: 3.

Описание теста: протестировать сохранение данных в файл.

Входные данные: Количество элементов массива: 2, название: BMW, модель: X6, цена: 2000000, название: KIA, модель RIO, цена: 890000.

Ожидаемые выходные данные:

Сообщение об успешном сохранении.

Сценарий:

- 1) Запускаем программу.
- 2) При запросе введения количества элементов ввести 2.
- 3) При запросе введения марка: BMW.
- 4) При запросе введения модель X6.
- 5) При запросе введения цена: 2000000.
- 6) При запросе введения марка: KIA.

- 7) При запросе введения модель RIO.
- 8) При запросе введения цена: 890000.
- 9) Открыть созданный файл.

Тестовый пример 4.

Требования: 4.

Описание теста: протестировать сортировку массива.

Входные данные: Количество элементов массива: 2, название: BMW, модель: X6, цена: 2000000, название: KIA, модель RIO, цена: 890000.

Ожидаемые выходные данные:

Сообщение об успешной сортировке.

Сценарий:

- 1) Запускаем программу.
- 2) При запросе введения количества элементов ввести 2.
- 3) При запросе введения марка: BMW.
- 4) При запросе введения модель X6.
- 5) При запросе введения цена: 2000000.
- 6) При запросе введения марка: KIA.
- 7) При запросе введения модель RIO.
- 8) При запросе введения цена: 890000.
- 9) Открыть файл и проверить правильность сортировки.

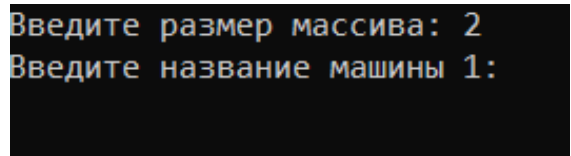
Результаты тестирования.

Сценарий 1.1:

- 1) Запускаем программу.
- 2) При запросе введения количества элементов ввести 2.

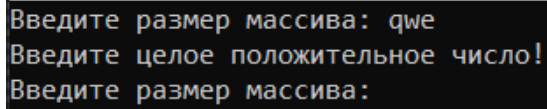
Сценарий 1.2:

- 1) Запускаем программу.
- 2) При запросе введения количества элементов ввести «qwe».



```
Введите размер массива: 2
Введите название машины 1:
```

Рисунок 1 – Результат 1.1



```
Введите размер массива: qwe
Введите целое положительное число!
Введите размер массива:
```

Рисунок 2 – Результат 1.2

Вывод: результат соответствует тест-плану и тест-требованиям.

Тестовый пример 2.

Сценарий 2.1:

- 1) Запускаем программу.
- 2) При запросе введения количества элементов ввести 2.
- 3) При запросе введения название: BMW.
- 4) При запросе введения модель X6.
- 5) При запросе введения цена: 2000000.
- 6) При запросе введения название: KIA.
- 7) При запросе введения модель RIO.
- 8) При запросе введения цена: 890000.

Сценарий 2.2:

- 1) Запускаем программу.
- 2) При запросе введения количества элементов ввести 1
- 3) При запросе введения название: (пустое значение), затем при повторном значении «KIO».
- 4) При запросе введения модель RIO.
- 5) При запросе введения цена: ghb, затем при повторном вводе 890000.

```
Введите размер массива: 2
Введите название машины 1: BMW
Введите марку машины 1: X6
Введите цену машины 1: 2000000
Введите название машины 2: KIA
Введите марку машины 2: RIO
Введите цену машины 2: 890000
Отсортировано!
Сохранение прошло успешно!
```

Рисунок 3 – Результат 2.1

```
Введите размер массива: 1
Введите название машины 1:
Введите название машины 1: KIA
Введите марку машины 1: RIO
Введите цену машины 1: qwe
Введите целое число!
Введите цену машины 1: 890000
Отсортировано!
Сохранение прошло успешно!
```

Рисунок 4 – Результат 2.2

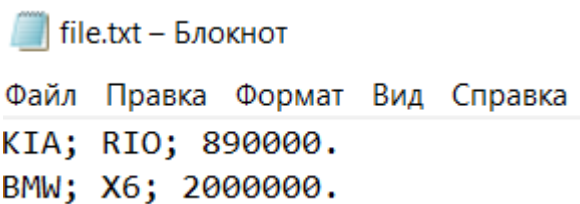
Тестовый пример 3.

Сценарий:

- 1) Запускаем программу.
- 2) При запросе введения количества элементов ввести 2.
- 3) При запросе введения марка: BMW.
- 4) При запросе введения модель X6.
- 5) При запросе введения цена: 2000000.
- 6) При запросе введения марка: KIA.
- 7) При запросе введения модель RIO.
- 8) При запросе введения цена: 890000.
- 9) Открыть созданный файл.

```
Введите размер массива: 2
Введите название машины 1: BMW
Введите марку машины 1: X6
Введите цену машины 1: 2000000
Введите название машины 2: KIA
Введите марку машины 2: RIO
Введите цену машины 2: 890000
Отсортировано!
Сохранение прошло успешно!
```

Рисунок 5 – Результат тестирования



file.txt – Блокнот

Файл Правка Формат Вид Справка

KIA; RIO; 890000.
BMW; X6; 2000000.

Рисунок 6 – Результат сохранения в файл

Тестовый пример 4.

Сценарий:

- 1) Запускаем программу.
- 2) При запросе введения количества элементов ввести 2.
- 3) При запросе введения марка: BMW.
- 4) При запросе введения модель X6.
- 5) При запросе введения цена: 2000000.
- 6) При запросе введения марка: KIA.
- 7) При запросе введения модель RIO.
- 8) При запросе введения цена: 890000.
- 9) Открыть файл и проверить правильность сортировки.

```
Введите размер массива: 2
Введите название машины 1: BMW
Введите марку машины 1: X6
Введите цену машины 1: 2000000
Введите название машины 2: KIA
Введите марку машины 2: RIO
Введите цену машины 2: 890000
Отсортировано!
Сохранение прошло успешно!
```

Рисунок 7 – Результат тестирования



file.txt – Блокнот

Файл Правка Формат Вид Справка

KIA; RIO; 890000.

BMW; X6; 2000000.

Рисунок 8 – Результат сортировки