

ТЕСТИРОВАНИЕ

Тест-требования

1. Протестировать поведение программы при вводе размера массива;
2. Протестировать создание элемента в массиве;
3. Протестировать сортировку массива и его сохранения в файл;

Тест-план

Тестовый пример 1

Номер тест требования 1

Описание теста: в данном тесте проверяется механизм создания массива, путем ввода корректного и некорректного размера.

Входные данные:

1. При позитивном – целочисленное значение;
2. При негативном – значение отличное от целочисленного.

Выходные данные:

1. При негативном – предложение о повторном вводе размера массива;
2. При позитивном – продолжение работы программы.

Позитивный сценарий:

1. Запускаем программу;
2. Вводим целочисленное значение для размера массива.

Негативный сценарий:

1. Запускаем программу;
2. Вводим значение, отличное от целочисленного.

Тестовый пример 2

Номер тест требования 2

Описание теста: в данном тесте проверяется механизм создания элемента массива, путем ввода корректных и некорректных значений.

Входные данные:

1. При позитивном – корректные значение;
2. При негативном – некорректные значения.

Выходные данные:

1. При негативном – повторный ввод значений;
2. При позитивном – продолжение работы программы и сохранение данных.

Позитивный сценарий:

1. Запускаем программу;
2. Вводим целочисленное значение для размера массива.
3. Для каждого элемента массива вводим корректные значения: направление поездки, продолжительность, цена.

Негативный сценарий:

1. Запускаем программу;
2. Вводим целочисленное значение для размера массива.
3. Для каждого элемента массива вводим некорректные значения: направление поездки, продолжительность, цена.

Тестовый пример 3

Номер тест требования 3

Описание теста: в данном тесте проверяется сортировка массива по убыванию по сочетанию двух свойств (продолжительность, цена), а также сохранение этого массива в файл.

Входные данные:

1. Массив данных.

Выходные данные:

1. Отсортированный массив данных, сохраненный в файле.

Сценарий:

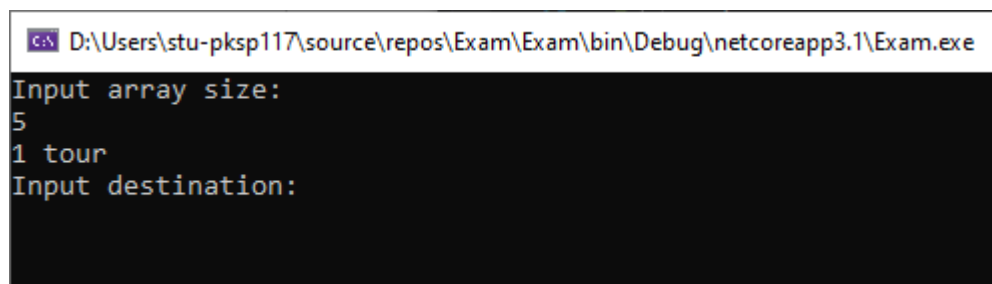
1. Запускаем программу;
2. Создаем массив данных;
3. Заполняем его значениями;
4. Открываем файл и смотрим результаты сортировки и сохранения.

Результаты тестирования

Тестовый пример 1.

Позитивный сценарий:

1. Запускаем программу;
2. Вводим целочисленное значение для размера массива (5);
3. Видим, что программа работает дальше.

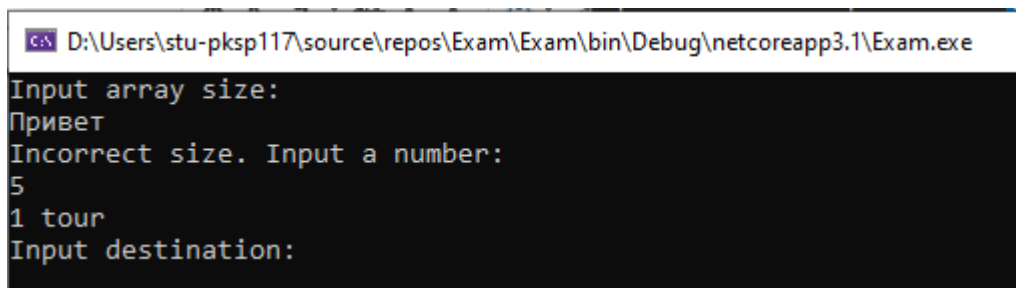


```
D:\Users\stu-pksp117\source\repos\Exam\Exam\bin\Debug\netcoreapp3.1\Exam.exe
Input array size:
5
1 tour
Input destination:
```

Рисунок 1 – Результат позитивного тестирования первого тестового примера

Негативный сценарий:

1. Запускаем программу;
2. Вводим нецелочисленное значение для размера массива (“Привет”);
3. Видим, что программа запрашивает повторный ввод размера массива.



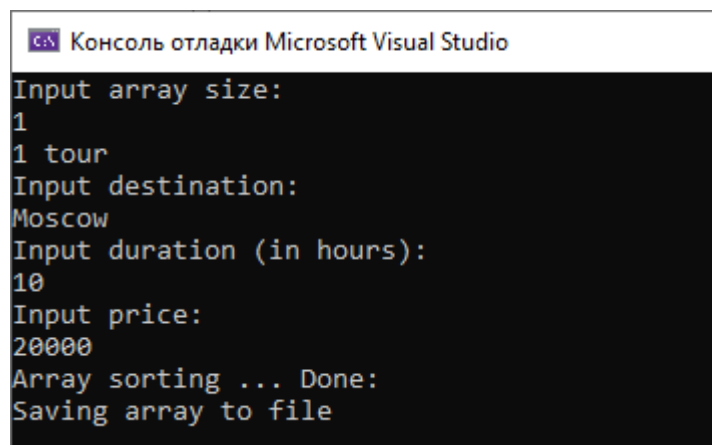
```
D:\Users\stu-pksp117\source\repos\Exam\Exam\bin\Debug\netcoreapp3.1\Exam.exe
Input array size:
Привет
Incorrect size. Input a number:
5
1 tour
Input destination:
```

Рисунок 2 – Результат негативного тестирования первого тестового примера

Тестовый пример 2.

Позитивный сценарий:

1. Запускаем программу;
2. Вводим целочисленное значение для размера массива (1);
3. Вводим направление поездки (Moscow);
4. Вводим продолжительность поездки (10);
5. Вводим стоимость поездки (20000);
6. Видим, что программа работает дальше.



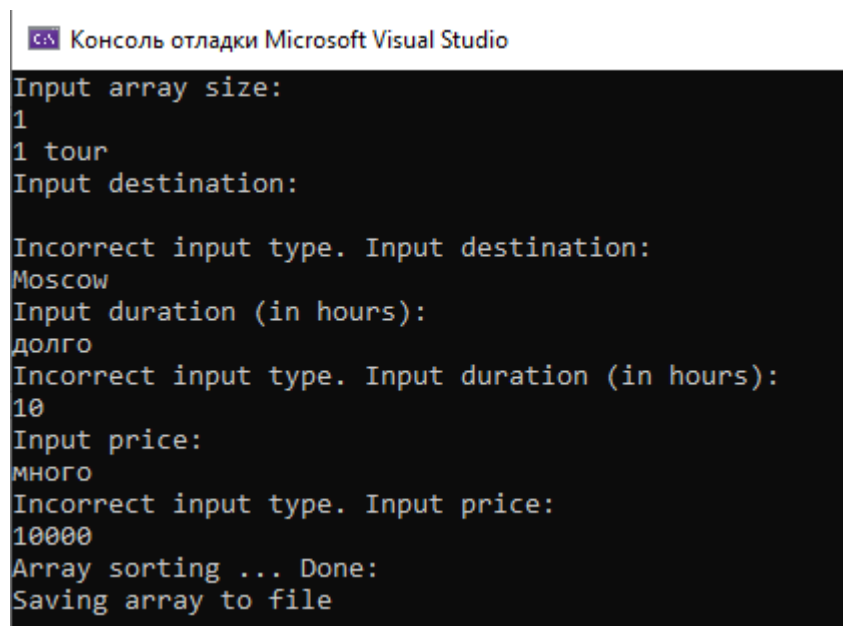
```
Консоль отладки Microsoft Visual Studio
Input array size:
1
1 tour
Input destination:
Moscow
Input duration (in hours):
10
Input price:
20000
Array sorting ... Done:
Saving array to file
```

Рисунок 3 – Результат позитивного тестирования второго тестового примера

Негативный сценарий:

1. Запускаем программу;
2. Вводим целочисленное значение для размера массива (1);
3. Вводим некорректное направление поездки ();

4. Видим, что программа запрашивает повторный ввод направления поездки.
5. Вводим корректное направление поездки (Moscow);
4. Вводим продолжительность поездки (долго);
5. Видим, что программа запрашивает повторный ввод продолжительности поездки.
6. Вводим корректную продолжительность поездки (10);
7. Вводим стоимость поездки (много);
8. Видим, что программа запрашивает повторный ввод стоимости поездки.
9. Вводим корректную стоимость поездки;
10. Видим, что программа работает дальше;



```
Консоль отладки Microsoft Visual Studio
Input array size:
1
1 tour
Input destination:

Incorrect input type. Input destination:
Moscow
Input duration (in hours):
долго
Incorrect input type. Input duration (in hours):
10
Input price:
много
Incorrect input type. Input price:
10000
Array sorting ... Done:
Saving array to file
```

Рисунок 4 – Результат негативного тестирования второго тестового примера

Тестовый пример 3.

Сценарий:

1. Запускаем программу;

2. Создаем массив с размером 4 и заполняем его следующими значениями:

а. Первый тур. Направление поездки – А. Продолжительность – 10. Стоимость 500.

б. Второй тур. Направление поездки – В. Продолжительность – 10. Стоимость 1000.

с. Третий тур. Направление поездки – С. Продолжительность – 15. Стоимость 200.

д. Четвертый тур. Направление поездки – D. Продолжительность – 5. Стоимость 2000.

3. Открываем файл и смотрим сортировку. Результат сортировки по убыванию по двум полям должен быть так: третий тур, второй тур, первый тур, четвертый тур.

```
Консоль отладки Microsoft Visual Studio
Input array size:
4
1 tour
Input destination:
A
Input duration (in hours):
10
Input price:
500
2 tour
Input destination:
B
Input duration (in hours):
10
Input price:
1000
3 tour
Input destination:
C
Input duration (in hours):
15
Input price:
200
4 tour
Input destination:
D
Input duration (in hours):
5
Input price:
2000
Array sorting ... Done:
Saving array to file
```

Рисунок 5 – Ввод тестовых данных для тестового примера 3

tours.txt	Exam.csproj	Tour.cs	TourSystem.cs
1	Destination: C duration (in hours): 15 price: 200		
2	Destination: B duration (in hours): 10 price: 1000		
3	Destination: A duration (in hours): 10 price: 500		
4	Destination: D duration (in hours): 5 price: 2000		
5			

Рисунок 6 – Результаты тестирования для тестового примера 3

4. Видим, что сохранение массива в файл и сортировка прошли правильно и успешно.