Тест-требования

В ходе проектирования и разработки приложения были сформированы следующие тест-требования:

1. Проверить корректность введения количества в окне приложения.

Проверить, что данные о количестве людей применяются корректно в приложении, ошибки не выводятся.

1. Проверить функционал добавления информации о людях.

Проверить, что добавление информации о людях работает корректно. Не должно возникать сообщений об ошибках, а поля после добавления автоматически очищаться.

1. Проверить функционал сортировки массива.

Проверить, сначала добавив данные о людях, сохранить в файл массив, потом отсортировать и сохранить в другой файл. Сравнить их, данные должны быть отсортированы по «фамилия»+«имя».

1. Проверить вывод ошибки при ввод текста в поле «возраст» и добавлении человека.

Проверить, что при вводе текста и попытке добавления такого человека – выводится ошибка.

На основе данных тест-требований был составлен тест-план, состоящий из шести тестов. Этот тест-план представлен ниже.

Тест-план

Тестовый пример 1.

Требования 1.

Описание теста: проверить корректность введения количества в окне приложения.

Входные данные: 2.

Ожидаемые выходные данные: отсутствие сообщений об ошибках.

Сценарий:

1. Запустить приложение.
2. Ввести размерность.
3. Нажать «Применить»
4. Проверить, что сообщение об ошибке не выводится.

Тестовый пример 2.

Требования 2.

Описание теста: проверить функционал добавления информации о людях.

Входные данные:

* размерность: 2;
* человек 1: Фонина Анастасия, 19;
* человек 2: Иванов Иван, 12.

Ожидаемые выходные данные: добавление данных и очистка полей ввода.

Сценарий теста:

1. Запустить приложение.
2. Ввести размерность и нажать кнопку «Применить».
3. Добавить данные, путем ввода в соответствующие поля и нажатия на кнопку «Добавить».
4. Проверить что ошибки не выводятся.

Тестовый пример 3.

Требования 3.

Описание теста: проверить функционал сортировки массива.

Входные данные:

* размерность: 2;
* человек 1: Фонина Анастасия, 19;
* человек 2: Иванов Иван, 12;
* файл 2: result2.txt

Ожидаемые выходные данные: два файла, во втором файле сортированный массив.

Сценарий теста:

1. Запустить приложение.
2. Ввести размерность и нажать кнопку «Применить».
3. Добавить данные, путем ввода в соответствующие поля и нажатия на кнопку «Добавить».
4. Нажать кнопку «Сохранить» и сохранить файл.
5. Нажать кнопку «Сортировать».
6. Нажать кнопку «Сохранить» и сохранить второй файл.
7. Сравнить файлы.

Тестовый пример 3.

Требования 3.

Описание теста: проверить вывод ошибки при ввод текста в поле «возраст» и добавлении человека.

Входные данные:

* размерность: 2;
* человек 1: Фонина Анастасия, апр.

Ожидаемые выходные данные: сообщение об ошибке.

Сценарий теста:

1. Запустить приложение.
2. Ввести размерность и нажать кнопку «Применить».
3. Добавить данные, путем ввода в соответствующие поля и нажатия на кнопку «Добавить».
4. Получить сообщение об ошибке.

Результаты тестирования

Тест 1. Проверить корректность введения количества в окне приложения.

Описание теста: проверить, что данные о количестве людей применяются корректно в приложении, ошибки не выводятся.

Входные данные: нет.

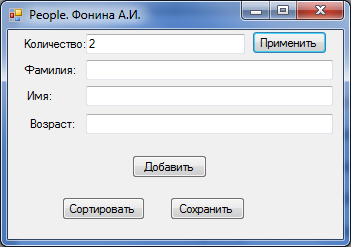


Рисунок 1 – Главная форма

Результаты теста соответствуют ожиданиям.

Тест 2. Проверить функционал добавления информации о человеке.

Входные данные:

* размерность: 2;
* человек 1: Фонина Анастасия, 19;
* человек 2: Иванов Иван, 12.

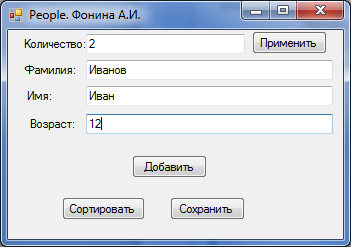


Рисунок 2 – Главная форма

Результат соответствует требованиям.

Тест 3. Проверить функционал сортировки массива.

Входные данные:

* размерность: 2;
* человек 1: Фонина Анастасия, 19;
* человек 2: Иванов Иван, 12;
* файл 2: result2.txt

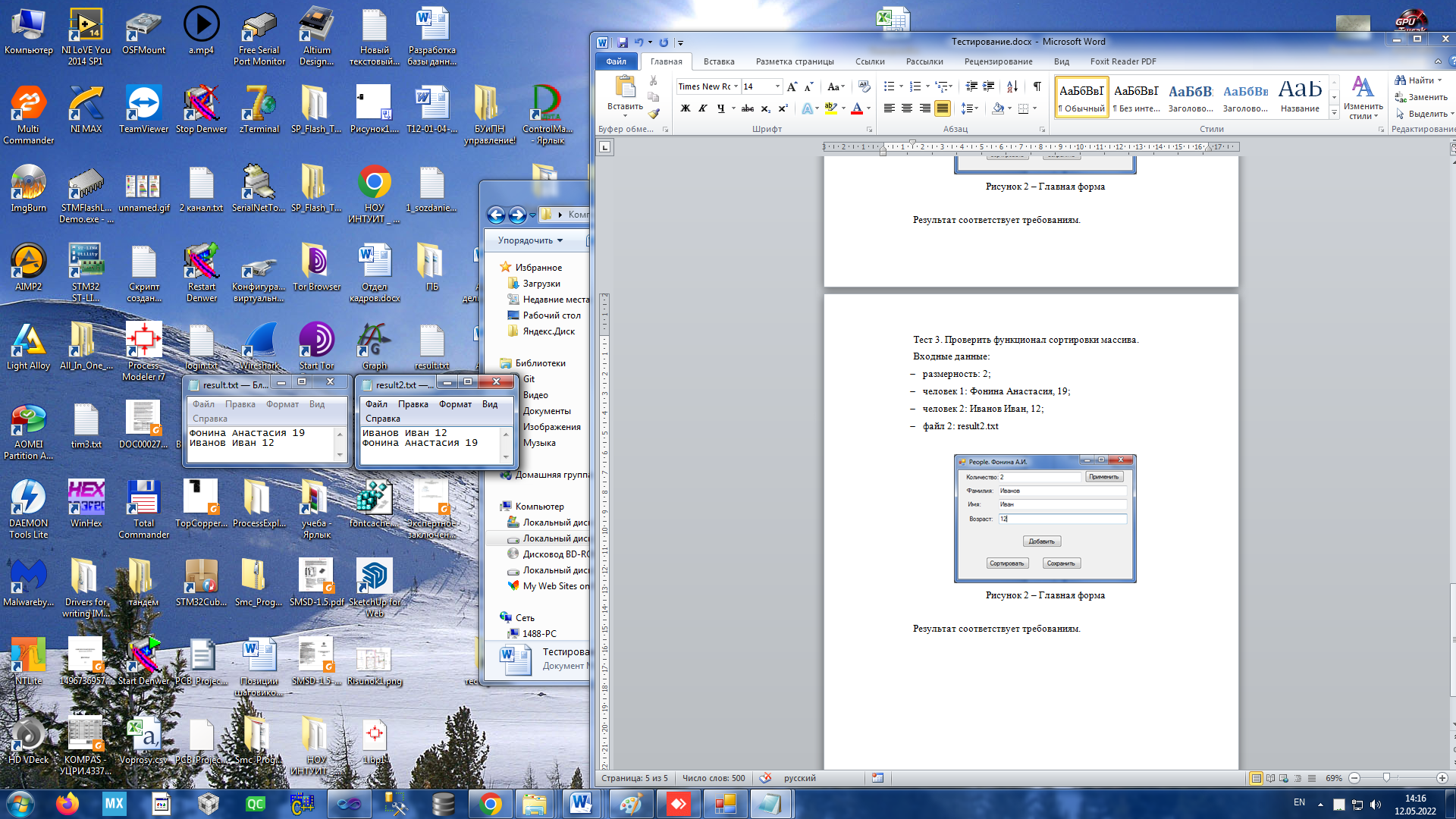


Рисунок 2 – Выходные файлы

Результат соответствует требованиям.

Тест 4. Проверить вывод ошибки при ввод текста в поле «возраст» и добавлении человека.

Входные данные:

* размерность: 2;
* человек: Фонина Анастасия, апр;

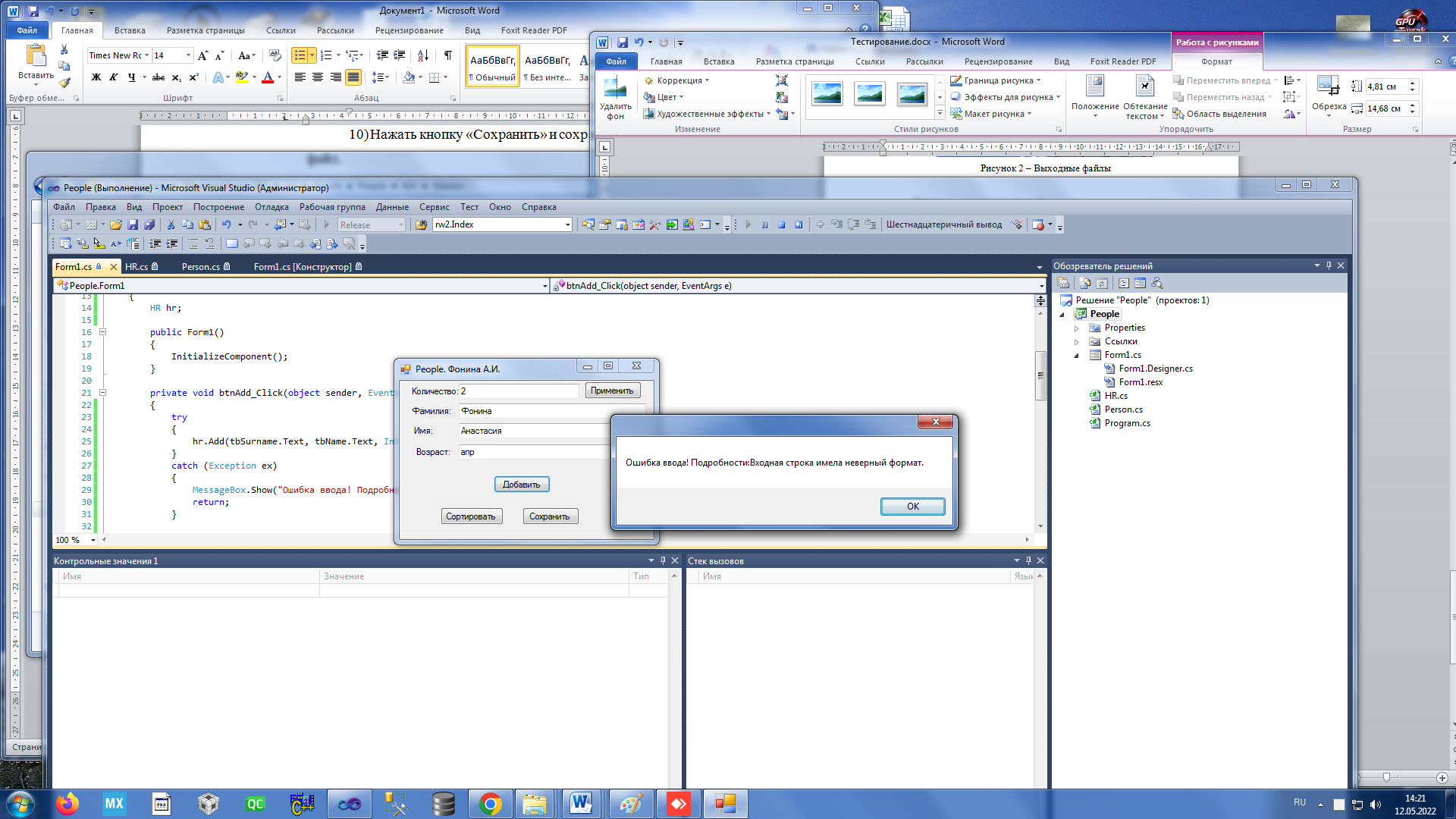


Рисунок 2 – Главная форма

Результат соответствует требованиям.