**Тестирование документации по игре «Тетрис»**

Данная работа посвящена созданию игры «Тетрис». Созданная программа предназначена для собственно игры в «Тетрис», сохранения текущего состояния игры и ведения счёта игроков.

Цель работы: создание игры «Тетрис»

Условия:  
При разработки программного продукта необходимо было выполнить ряд обязательных требований, описанных ниже:

1. Разрабатываемое приложение должно представлять собой Windows приложение.

2. Приложение в обязательном порядке независимо от предметной области, указанной в задании, должно выполнять следующие операции:

- при первом запуске программы создать список в коде программы и отобразить его в DataGridView.

- реализовать добавление в список нового объекта, удаление объекта из списка, редактирование объекта в списке;

- вывести на форму сведения об объектах списка, удовлетворяющих введенному пользователем критерию;

- сохранить список объектов в указанном пользователем файле (используя сериализацию и стандартное диалоговое окно). Для сохранения файла приложение должно создать подкаталог в текущей папке приложения, если подкаталог не существует. Имя подкаталога и файла выбирает пользователь.

- создать копию файла, используя метод копирования.

- используя меню или панель инструментов, вывести информацию о пути к программе, а также файлах и вложенных папках текущего каталога.

- сохранить в реестре имя подкаталога и файла.

- в подкаталоге Manual с помощью Блокнота создать текстовый файл с инструкцией к программе. Для просмотра файла использовать меню или панель инструментов, по которому запустить Блокнот и загрузить инструкцию.

- при повторном запуске программы прочитать из реестра запомненные имена, создать список путем чтения (десериализации) данных из файла. Отобразить список в DataGridView. При повторном запуске программы список загружается из файла, создание списка в коде программы не использовать.

3. Список должен быть реализован в виде коллекции, например, динамического массива. Для сохранения файла использовать стандартное диалоговое окно.

4. Программа не должна завершаться аварийно: сообщения о некорректном вводе данных, противоречивых или недопустимых значениях данных, при отсутствии данных по функциональному запросу пользователя и других нештатных ситуациях отображать в окнах сообщений.

5. Программа должна быть читабельной и содержать полезные комментарии.

В ходе работы были учтены и описаны данные условия.

**Описание работы**

Работа выполнена в файле Word, количество страниц 15.  
Тестируемая документация включает в себя следующие разделы:  
• Введение

• Актуальность

• Условие задачи

• Описание предметной области

• Описание класса предметной области

• Описание интерфейса пользователя на контрольном примере

• Таблица лидеров

• Форма загрузки сохранений

• Заключение

• Список литературы

**Критерии качества тестируемой документации:**

1. Структура документации.  
   Данный параметр важен, так как сложно ориентироваться в документациях, в которых все свалено в кучу. Данную документацию можно характеризоваться наличием структурированных материалов, в которых удобно адаптироваться.   
   Также документация дополнена информируемыми фотографиями, которые помогут разобраться в интерфейсе разрабатываемого приложения.  
   ТЗ в документации было описано в полном объеме, оно дает представление о том, как будет выглядеть будущая программа.
2. Однозначность   
   Терминология была описана в корректной форме, понятной каждому участнику проекта. Но стоит обратить внимание на важность вынесения терминов в отдельный раздел. Поэтому стоит дополнить документацию разделом «Терминология»
3. Непротиворечивость  
   В документации отсутствуют конфликтные требования
4. Актуальность  
   Документация и реально разработанный программный продукт соответствуют друг другу в один и тот же момент времени.
5. Тестируемость  
   Проверка-тестирование производится после создания функционала.
6. Соответствие стандартам  
   Тестируемая документация должна быть выполнена в соответствии со стандартами (ГОСТ) и ТЗ.

**Описание и обоснование метода тестирования документации**

Для данной работы был выбран метод рецензирования. Цель рецензирования тестовой документации состоит в том, чтобы обеспечить ее качество, полноту, соответствие требованиям и эффективность в контексте тестирования программного обеспечения.

Этот процесс включает в себя анализ документации с точки зрения различных аспектов, таких как правильность, полнота, читаемость, соответствие стандартам, корректность данных и т. д. В связи с этим данная методика тестирования оказалась наиболее эффективной, так как благодаря ей я смогла проанализировать документацию и определить степень ее качества.

**Список несоответствий в документации критериям качества**

1. Полнота

• Курсовая работа раскрыта полноценно, можно с легкостью разобраться в сути работе, а также прикрепленных инструкциях к проекту.

2. Однозначность

• Рекомендация: добавить разъяснение техническим терминам

3. Непротиворечивость

• В данном работе все элементы проекта не противоречат друг другу.

4. Актуальность

• Разработанный проект по сей день является актуальным, но нужно удостовериться работает ли он с новой версией ПО. В случае возникновения конфликта присутствует рекомендации к его устранению.

5. Структурированность

• Текст работы структурирован, содержание проработано хорошо. Оглавление соответствует подразделам.

6. Тестируемость

• Отсутствуют замеры совершения шагов в игре. Рекомендация – провести замеры.

7. Соответствие стандартам:

• Текст работы не содержит ссылки на используемую литературу согласно ГОСТ.

• Работа полностью соответствует ТЗ

• Рекомендация: включить в работу ссылки используемой литературы согласно ГОСТ.