***Документация к проекту «To Do list»***

**Обзор**

* Название проекта: «To Do list»
* Дата: 26.12.2023
* Члены команды:
* Старовойтова Ева - документация
* Дужик Ульяна - написание кода
* Султанов Арсений – Team Lead

**Введение**

To do list — это простой список дел или список задач. Запиши в него все свои важные дела, чтобы не забыть. «Список дел» позволит вам эффективно организовать свое рабочее время. В него можно записать все: от списка покупок до важных деловых встреч.

**Сроки выпуска:**

* Дата анонса: 20.12.2023
* Дата выхода: 26.12.2023

**Репозиторий кода**

* <https://github.com/LikeWolfC/FinalToDo.git>

**Тестирование**

1. Получить правильную реакцию программы на добавление задачи.

* При добавлении задачи программа корректно сохраняет ее в файле «todo.txt».
* Идентификатор задачи ID увеличивается каждый раз при добавлении новой задачи.

Код работает корректно

1. Получить правильную реакцию программы на функцию поиска задач.

* При вводе ID задачи, соответствующая задача корректно находится и отображается.
* При вводе несуществующего ID, программа правильно обрабатывает эту ситуацию.

Код работает корректно

1. Поучить правильную реакцию программы на функцию удаления задачи.

* При выборе задачи, которую нужно удалить, программа срабатывает правильно.
* При отмене удаления, программа работает корректно.

Код работает корректно

1. Получить правильную реакцию программы на функцию обновления задач.

* При вводе нового описания для обновления задачи, программа удачно обновляется в файле «todo.txt».
* При отмене обновления задачи, программа работает корректно.

Код работает корректно

**Основные функции**

* void addtodo()
* Входные параметры: Отсутствуют.
* Выходные параметры: Отсутствуют.
* Описание: Функция позволяет пользователю добавить новую задачу в список. Записывает новую задачу в файл "todo.txt" и увеличивает счетчик ID. Позволяет пользователю добавлять несколько задач подряд.
* void print(todo s)
* Входные параметры: Экземпляр структуры todo (s), представляющий задачу.
* Выходные параметры: Отсутствуют.
* Описание: Функция выводит информацию о задаче (ID и описание) на экран консоли.
* void readData()
* Входные параметры: Отсутствуют.
* Выходные параметры: Отсутствуют.
* Описание: Функция считывает данные из файла "todo.txt" и выводит текущие задачи на экран консоли.
* int searchData()
* Входные параметры: Отсутствуют.
* Выходные параметры: Возвращает ID найденной задачи.
* Описание: Функция позволяет пользователю ввести ID задачи для поиска. Считывает данные из файла "todo.txt", находит соответствующую задачу и выводит ее на экран. Возвращает ID найденной задачи.
* void deleteData()
* Входные параметры: Отсутствуют.
* Выходные параметры: Отсутствуют.
* Описание: Функция позволяет пользователю удалить задачу. Вызывает функцию searchData() для поиска задачи по ID, после чего удаляет найденную задачу из файла "todo.txt".
* void updateData()
* Входные параметры: Отсутствуют.
* Выходные параметры: Отсутствуют.
* Описание: Функция позволяет пользователю обновить описание задачи. Вызывает функцию searchData() для поиска задачи по ID, после чего обновляет описание задачи в файле "todo.txt".
* int main()
* Входные параметры: Отсутствуют.
* Выходные параметры: Возвращает 0 в случае успешного завершения программы.
* Описание: Основная функция программы, в которой осуществляется циклическое отображение меню для пользователя и вызов соответствующих функций в зависимости от выбора пользователя. Использует файлы "todo.txt" и "id.txt" для хранения данных и ID задач.

**Логика программы**

Программа представляет собой консольное приложение для управления списком задач.

* Добавление задачи (addtodo):
* Программа запрашивает у пользователя новую задачу.
* Увеличивает переменную ID, представляющую уникальный идентификатор задачи.
* Записывает новую задачу в файл "todo.txt", включая уникальный идентификатор и саму задачу.
* Спрашивает пользователя, хочет ли он добавить еще задач, и, в случае положительного ответа, рекурсивно вызывает функцию addtodo.
* Чтение данных (readData):
* Программа открывает файл "todo.txt" и считывает данные о задачах.
* Для каждой задачи выводит на экран ее уникальный идентификатор и описание.
* Поиск данных (searchData):
* Программа запрашивает у пользователя уникальный идентификатор задачи для поиска.
* Открывает файл "todo.txt" и ищет задачу с указанным идентификатором.
* Выводит на экран найденную задачу.
* Удаление данных (deleteData):
* Программа вызывает функцию searchData для поиска задачи, которую пользователь хочет удалить.
* Запрашивает подтверждение от пользователя.
* Если подтверждение положительное, создается временный файл "temp.txt", куда записываются все задачи, кроме удаляемой.
* Удаляется исходный файл "todo.txt" и переименовывается временный файл.
* Обновление данных (updateData):
* Программа вызывает функцию searchData для поиска задачи, которую пользователь хочет обновить.
* Запрашивает подтверждение от пользователя.
* Если подтверждение положительное, создается временный файл "temp.txt", в который записываются все задачи, а обновляемая задача заменяется на новую.
* Удаляется исходный файл "todo.txt" и переименовывается временный файл.
* Основной цикл (main):
* Программа входит в бесконечный цикл, предлагая пользователю выбор действия.
* В зависимости от выбора пользователя вызываются соответствующие функции.

**Уровень сложности:** 8/10

**Автор**

Проект «To Do list» разработан Старовойтовой Евой, Дужик Ульяной, Султановым Арсением.