Техническое задание (ТЗ) для разработки проекта по управлению данными продуктов

---

1. Общие сведения

Проект представляет собой консольное приложение для управления данными о продуктах. Приложение позволяет генерировать случайные данные о продуктах, сохранять их в текстовый файл, выводить данные на экран, а также искать и удалять записи по их идентификатору.

2. Цель проекта

Создать приложение, которое будет:

- Генерировать случайные записи продуктов.
- Сохранять данные в текстовый файл.
- Предоставлять интерфейс для вывода, поиска и удаления данных.

3. Требования к функционалу

3.1 Классы

- **Класс Node_data**:
  - **Поля:**
    - id — уникальный идентификатор продукта.
    - products_name — название продукта.
    - category_id — идентификатор категории продукта.
    - countId — статическая переменная, отвечающая за генерацию уникальных идентификаторов.
    - products_names — статический вектор, содержащий названия категорий продуктов.
    - product_map — статическая карта, связывающая названия категорий с их идентификаторами.
  - **Методы:**
    - Конструкторы:
      - Node_data(const string& products_name) — конструктор, создающий объект продукта на основе названия категории.
      - Node_data(int id, const string& name) — конструктор, создающий объект продукта с указанным идентификатором и названием категории.

        - get_id() — метод для получения идентификатора продукта.
        - NodeString() — метод для получения строки, описывающей продукт, с форматированием.
        - generate_products() — статический метод для генерации случайного продукта.
        - find_id(string product_id) — статический метод для поиска продукта по его идентификатору.
        - showInterface() — метод для вывода интерфейса таблицы с данными продуктов.
- **Класс Data_base**:
    - **Поля:**
        - path — путь к файлу для хранения данных.
    - **Методы:**
        - Конструкторы:
            - Data_base(string path) — конструктор, задающий путь к файлу.
        - addLine(const string& line) — метод для добавления строки в файл.
        - removeLines(const vector<int>& ids) — метод для удаления строк из файла по списку идентификаторов.
        - showDoc() — метод для отображения содержимого файла.
        - splitStr(string str, char del) — метод для разделения строки по заданному разделителю.

3.2 Основные функции

- Генерация случайных записей продуктов с использованием метода generate_products() класса Node_data.
- Сохранение данных о продуктах в текстовый файл через метод addLine() класса Data_base.
- Вывод данных на экран с использованием форматированного метода NodeString() и интерфейса showInterface() класса Node_data.
- Поиск продукта по идентификатору с использованием метода find_id() класса Node_data.
- Удаление записей из файла по списку идентификаторов через метод removeLines() класса Data_base.

4. Требования к интерфейсу

- Приложение должно быть консольным.
- Интерфейс должен быть простым и интуитивно понятным.
- Вывод данных о продуктах должен быть в виде таблицы с колонками: product_id, product_name, category_id.

5. Требования к технологии

- Язык программирования: C++.
- Использование стандартных библиотек C++.
- Поддержка компиляции и выполнения на ОС Windows (использование #include <Windows.h>).

6. Порядок работы программы

1. Программа запускается и выводит заголовок таблицы с использованием метода showInterface().
2. Генерируются 10 случайных продуктов и сохраняются в файл.
3. Программа выводит содержимое файла на экран.
4. Производится поиск и вывод продуктов по их идентификаторам.
5. (Опционально) Удаление строк по идентификаторам