

РАЗРАБОТКА АРХИТЕКТУРЫ ПРИЛОЖЕНИЯ ДЛЯ ИНТЕРВАЛЬНОГО ЗАПОМИНАНИЯ

Плетнёв М.Э., студент

Гилёв А.Ю., старший преподаватель

Бирский Филиал УУНиТ, г. Бирск, Россия

Аннотация. В статье рассматривается проектирование структуры учебного приложения в соответствии с принципами объектно-ориентированного проектирования. Приведены основные классы приложения и диаграмма классов.

Ключевые слова: Архитектура программного обеспечения, объектно-ориентированное проектирование, принципы SOLID, паттерны проектирования.

Введение

В процессе «» изучения дисциплины «Проектирование информационных систем» была поставлена задача о проектировании простой учебной программы с использованием современных подходов к разработке сложного программного обеспечения.

Приложение Memorize Cards (Запоминайте карточки) предназначено для обучения иностранным языкам и заучивания слов с использованием карточек. Метод интервального запоминания основан на том факте, что человек забывает новую информацию в первое время значительно интенсивнее, чем в более позднее. Зависимость скорости забывания от времени экспоненциальная. Поэтому метод предлагает обновлять информацию периодически, причём эти периоды постепенно увеличиваются.

Основные классы приложения

Класс Card

Этот класс представляет собой модель одной карточки слова. Он содержит следующие свойства и методы:

- TimeInterv – временной интервал;
- CreatDate – дата создания;
- Word1, Word2 – два слова, которые могут быть связаны между собой как перевод или синонимы;
- Hint – подсказка к слову;
- StatusWord – статус слова;
- Метод ChangeStatusCard - изменяет статус карточки.

Класс FileCard

Класс FileCard отвечает за работу с файлами, связанными с карточкой.

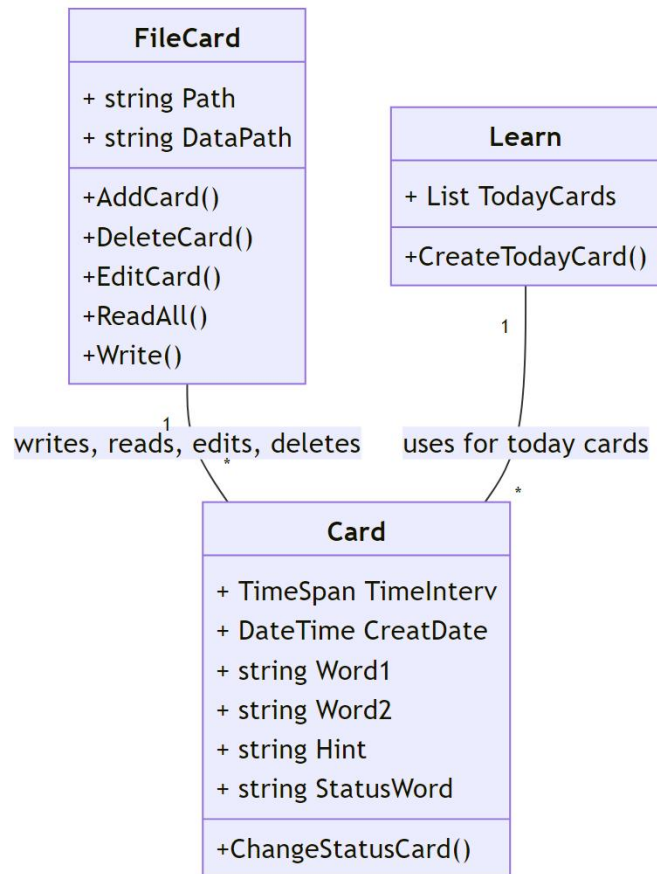
Основные методы этого класса включают:

- Добавление новой карточки (AddCard)
- Удаление существующей карточки (DeleteCard)
- Редактирование карточки (EditCard)
- Чтение всех карточек из файла (ReadAll)
- Запись карточки в файл (Write)

Класс Learn

Этот класс предназначен для управления процессом обучения. В нем содержится поле TodayCards, представляющее собой список карточек, которые нужно изучить сегодня. Также имеется метод CreateTodayCard, предназначенный для формирования списка карточек на текущий день. Реализация метода отсутствует.

Диаграмма классов



Заключение

Описанные классы представляют собой каркас системы для изучения слов с помощью карточек. По нашему мнению, архитектурно классы разделены достаточно хорошо: Card хранит информацию о конкретной карточке, FileCard управляет операциями ввода-вывода, а Learn формирует ежедневные задачи. Теперь требуется дальнейшая работа над реализацией каждого компонента. Нами планируется:

- Реализовать методы чтения и записи файлов, используя форматы JSON или XML.

- Определить возможные статусы карточек и реализовать логическую обработку изменений статуса в методе `ChangeStatusCard`.
- Реализовать механизм фильтрации карточек для формирования ежедневного плана обучения в классе `Learn`.
- Добавить проверки ошибок и обработки исключительных ситуаций во всех методах, работающих с файлами.
- Рассмотреть возможность использования базы данных вместо простого хранения данных в файлах для повышения производительности и удобства работы с большими объемами данных.

Литература

1. Арора Г., Чилберто Д. Паттерны проектирования для C# и платформы .NET Core. — СПб.: Питер, 2021. — 352 с.
2. Вайсфельд М. Объектно-ориентированное мышление. — СПб.: Питер, 2014. — 304 с.
3. Гамма Э., Хелм Р., Джонсон Р., Влиссидес Д. Приемы объектно-ориентированного проектирования. паттерны проектирования. — СПб.: Питер, 2018. — 368 с.
4. Мартин Р. Чистая архитектура. Искусство разработки программного обеспечения. — СПб.: Питер, 2018. — 352 с.
5. Тепляков С. Паттерны проектирования на платформе .NET. — СПб.: Питер, 2015. — 320 с.