

# GeoDataConverter

## План реализации WPF-приложения GeoDataConverter с использованием FDO

### 1. Подготовительный этап

- **Анализ требований:**
  - Изучить функциональные и технические требования.
  - Определить особенности маппинга атрибутов и конвертации форматов данных (SHP, SDF, SQLite) с использованием FDO.
- **Составление технической документации:**
  - Описать архитектуру, сценарии использования (use-case) и схемы данных.
  - Определить точки интеграции FDO для чтения и записи геоданных.
- **Настройка среды разработки:**
  - Установить Visual Studio 2022.
  - Создать WPF-проект на .NET 8.
  - Подключить необходимые пакеты через NuGet или добавить ссылки на DLL FDO.

### 2. Проектирование архитектуры и структуры приложения

- **Выбор архитектурного шаблона:**
  - Применить паттерн MVVM для разделения бизнес-логики, представления и управления пользовательским интерфейсом.
- **Структура проекта:**

- **Models:** Классы для представления геоданных, схем атрибутов и других моделей.
- **Views:** XAML-файлы и связанные с ними ресурсы (UserControl, окна).
- **ViewModels:** Логика связывания модели и представления.
- **Services:** Реализация работы с файлами, интеграция с FDO, конвертация и валидация данных.
- **Utilities:** Вспомогательные классы (логирование, конвертеры типов и т.д.).
- **Определение зависимостей:**
  - Выделить компоненты FDO, которые будут использоваться для импорта и экспорта данных.

## 3. Реализация основных компонентов

### 3.1 Модуль загрузки и анализа исходных данных

- Реализовать сервисы для импорта данных из файлов форматов SHP, SDF и SQLite с использованием FDO.
- Разработать парсеры для автоматического извлечения атрибутов, типов данных и геометрии (поддерживаются точки, линии и полигоны).

### 3.2 Модуль маппинга атрибутов

- **Создание модели маппинга:**
  - Разработать классы, отражающие соответствие атрибутов исходной схемы и целевой схемы.
- **Реализация интерфейса ручного маппинга:**
  - Обеспечить возможность пользователю вручную задавать соответствие атрибутов.
  - Добавить возможность сохранения шаблонов маппинга для повторного использования.

### 3.3 Модуль преобразования данных и валидации

- **Преобразование типов данных:**

- Реализовать логику преобразования (например, строка → DateTime) в соответствии с настройками маппинга.

- **Валидация данных:**

- Проверить корректность данных (например, формат даты, соответствие типов).
- Вывести пользователю все обнаруженные ошибки.
- Прервать процесс конвертации при наличии ошибок до их устранения.

### 3.4 Модуль экспорта данных

- Реализовать генерацию выходного файла с выбранным форматом (SHP, SDF или SQLite) с новой схемой данных с помощью FDO.
- Провести тестирование экспорта, чтобы убедиться, что файл соответствует заданной схеме.

## 4. Разработка пользовательского интерфейса

- **Основное окно приложения:**

- Обеспечить выбор исходного файла, указание целевого формата и доступ к настройкам маппинга.

- **Диалоги и окна:**

- Диалог выбора файла для импорта.
- Окно настройки маппинга атрибутов с возможностью ручного редактирования и сохранения шаблонов.
- Окно отображения ошибок валидации с подробными рекомендациями.

- **Предварительный просмотр:**

- Реализовать функционал для предварительного просмотра исходных данных и результатов преобразования.

## 5. Интеграция библиотеки FDO

- **Анализ возможностей FDO:**
  - Изучить документацию FDO для определения подходящих API для чтения и записи геоданных.
- **Настройка и подключение FDO:**
  - Подключить FDO через NuGet (если доступно) или добавить ссылки на необходимые DLL.
- **Инкапсуляция работы с FDO:**
  - Создать отдельный сервис в слое Services для оборачивания вызовов FDO API:
    - Метод для открытия файла и извлечения схемы.
    - Метод для экспорта данных в выбранный формат.
- **Тестирование интеграции:**
  - Провести тестирование работы с FDO на тестовых наборах данных, убедившись, что данные корректно читаются и записываются.

## 6. Логирование и обработка ошибок

- **Логирование:**
  - Реализовать систему логирования (например, с использованием NLog или Serilog) для отслеживания действий приложения и ошибок.
- **Обработка исключений:**
  - Настроить централизованную обработку ошибок:
    - При возникновении исключений (например, при работе с FDO) выводить подробную информацию пользователю.
    - Прекращать процесс конвертации до устранения обнаруженных ошибок.

## 7. Тестирование и отладка

- **Модульное тестирование:**
  - Написать юнит-тесты для основных компонентов (парсеры, маппинг, преобразование данных) с использованием MSTest, NUnit или xUnit.
- **Интеграционное тестирование:**
  - Проверить корректное взаимодействие между модулями (например, обмен данными между сервисами FDO и модулем экспорта).
- **Пользовательское тестирование:**
  - Провести тестирование с участием конечных пользователей для оценки удобства интерфейса и выявления ошибок.

## 8. Документация и релиз

- **Техническая документация:**
  - Описать архитектуру приложения, принципы работы модулей, особенности интеграции с FDO и схемы данных.
- **Пользовательская документация:**
  - Подготовить руководство пользователя, описывающее установку, настройку маппинга, процедуру валидации и экспорта данных.
- **Финальное тестирование и релиз:**
  - Провести финальное тестирование, устранить выявленные баги и подготовить релизную версию приложения.

## 9. Поддержка и дальнейшее развитие

- **Сбор обратной связи:**
  - Организовать сбор отзывов от пользователей для определения направлений дальнейшего улучшения функционала.
- **Расширение функционала:**
  - Планировать добавление поддержки новых форматов, дополнительных типов геометрии и пространственных операций.
- **Обновления и поддержка:**

- Разработать стратегию регулярных обновлений для повышения стабильности, производительности и функциональности приложения.

### Функциональные требования

### Use Cases

### Техническая часть реализации