

[Отрабатываемый материал](#)

[Цель](#)

[Задание](#)

[Функциональные требования](#)

[Не функциональные требования](#)

[Глоссарий](#)

[Семантика команд](#)

[Test cases](#)

[Definition of done](#)

## Отрабатываемый материал

SOLID, поведенческие, структурные, порождающие паттерны

## Цель

Проверить освоение студентом принципов SOLID, паттернов проектирования.

## Задание

Разработать приложение для взаимодействия и управления файловой системой.

## Функциональные требования

- Навигация по дереву файловой системы (относительные и абсолютные пути)
- Просмотр содержимого каталога в консоли
- Просмотр содержимого файлов в консоли
- Перемещение файлов
- Копирование файлов
- Удаление файлов
- Переименование файлов
- Консольный механизм взаимодействия с приложением
- Реализация операций для локальной файловой системы

## Не функциональные требования

- Система должна поддерживать взаимодействие посредством консольных команд, имеющих флаги.
- Логика работы системы не должна быть привязана к обработке консольных команд.
- Система должна поддерживать возможность расширения параметров консольных команд.
- Обработка команд не должна быть привязана к консоли.
- Система не должна быть завязана на локальную файловую систему.

- Вывод содержимого каталога должен быть параметризован глубиной выборки (значение по умолчанию - 1)
- Вывод системного каталога должен быть в виде дерева.
- Параметры выводимого дерева (символы обозначающие файл, папку, символы используемые для отступов должны быть программно параметризуемыми).
- Логика вывода содержимого каталога не должна быть завязана на консоль.
- Логика вывода содержимого файла не должна быть завязана на консоль.
- Система должна адекватно обрабатывать случаи коллизий имён.
- Система должна уметь переключаться между файловыми системами (например смена диска C, на диск D).
- После вывода результата на консольный интерфейс, программа должна ожидать ввод следующей команды.
- Для реализации системы нельзя использовать какие-либо сторонние библиотеки

## Глоссарий

- Относительный путь - путь от текущего положения, выбранного в системе
- Абсолютный путь - путь от положения, в которое изначально было сделано подключение

## Семантика команд

- connect [Address] [-m Mode]  
Address - абсолютный путь в подключаемой файловой системе  
Mode - режим файловой системы (требуется реализовать только локальную ФС, значение `local` )
- disconnect  
Отключается от файловой системы
- tree goto [Path]  
**Path** - относительный или абсолютный путь до каталога в файловой системе
- tree list {-d **Depth**}  
**Depth** - параметр, определяющий глубину выборки, должен объявляться флагом `-d`
- file show [Path] {-m **Mode**}  
**Path** - относительный или абсолютный путь до файла  
**Mode** - режим вывода файла (требуется реализовать только консольный, значение `console` )
- file move [SourcePath] [DestinationPath]  
**SourcePath** - относительный или абсолютный путь до перемещаемого файла  
**DestinationPath** - относительный или абсолютный путь до директории, куда файл должен быть перемещён
- file copy [SourcePath] [DestinationPath]  
**SourcePath** - относительный или абсолютный путь до копируемого файла  
**DestinationPath** - относительный или абсолютный путь до директории, куда файл должен быть скопирован
- file delete [Path]  
**Path** - относительный или абсолютный путь до удаляемого файла
- file rename [Path] [Name]  
**Path** - относительный или абсолютный путь до изменяемого файла  
**Name** - новое имя файла

## Test cases

- Протестировать парсер команд: обработка консольных команд с аргументами должна создавать команду корректного типа с корректными аргументами

## Definition of done

- Реализованы все функциональные требования
- Реализация соответствует всем нефункциональным требованиям
- Реализация не нарушает принципы SOLID, следует основным принципам ООП
- Реализован консольный интерфейс работы с приложением