

Homework 5

Задание N1

Создайте программу, эмулирующую модель файловой системы, ее иерархическую структуру. Т.е., имеем два типа объектов: каталог (Folder) и файл (File) со следующими свойствами:

- Каталог и файл имеют свойство - имя.
- В каталог можно добавлять другие файлы и каталоги, в произвольном порядке, произвольное количество.
- В файл нельзя добавлять другие каталоги или файлы.
- Файл имеет расширение. Каталог – нет.

Программа принимает на вход строку, представляющую собой путь к файлу/каталогу в некой структуре каталогов (например “ **root**/folder1/file.txt”), и строит дерево соответствующих объектов в памяти.

Пользователь может ввести следующую строку-путь, которая может содержать каталоги из предыдущего пути (например “ **root**/folder2/file.txt”) – в таком случае, к уже существующему пути будут добавлены новые каталоги и файлы.

Пользователь в любой момент может вывести на экран получившуюся структуру каталогов.

Важно! Эта задача не о работе с файловой системой. Не нужно создавать реальные файлы и папки. Только виртуальная структура в памяти!

Пример

Пример:

\$> root/folder1/file.txt - пользователь вводит строку

\$> print - команда вывода на экран

root/
folder1/
file.txt - программа выводит структуру на экран

\$> root/folder2/file.txt - пользователь вводит новую строку

\$> print

root/ - обновленная структура на экран
folder1/
file.txt
folder2/
file.txt

Hints

- Для представления структуры файлов/директорий удобно использовать паттерн проектирования [Composite](#).
- Для создания структуры из строк можно попробовать использовать паттерн [Builder](#).
- Разделяйте ответственность между классами – пусть один класс будет ответственным за создание структуры из строки, другой класс будет выводить структуру на экран а третий обрабатывать ввод пользователя
- Попробуйте использовать TDD на практике. Пишите код от тестов. Пусть вся программа, кроме ввода пользователя, будет покрыта тестами.