

Zadanie 1.

Utwórz klasę FileCommander przechowującą ścieżkę do katalogu w postaci prywatnego pola typu Path. Napisz konstruktor, który ustawia tę ścieżkę w katalogu domowym.

W klasie zdefiniuj metody publiczne:

- pwd, zwracającą napis zawierający aktualną ścieżkę,
- cd, przyjmującą ścieżkę i zmieniającą tę ścieżkę względem dotychczasowej.

Zadanie 2.

W klasie FileCommander napisz metodę ls, która zwraca listę napisów zawierających nazwy plików i katalogów w aktualnej ścieżce. Następnie, krok po kroku, zmodyfikuj tę metodę tak, aby:

1. wyniki zostały posortowane alfabetycznie,
2. nazwy katalogów były otoczone znakami [],
3. wyniki zostały posortowane tak, aby katalogi były przed plikami
4. wyniki zostały posortowane tak, aby katalogi były przed plikami, a dodatkowo obie grupy były posortowane alfabetycznie.

Zadanie 3.

W klasie FileCommander napisz metodę find, która przyjmie argument napisowy. Metoda powinna zwrócić listę napisów zawierających pełne ścieżki wszystkich plików, które znajdują się w aktualnej ścieżce i zawierają w nazwie podany argument.

Zadanie 4.

Napisz klasę FileCommanderCLI, która przyjmie jako parametry konstruktora strumień wejściowy i strumień wyjściowy. Konstruktor powinien utworzyć obiekt klasy FileCommander i przechowywać go w prywatnym polu klasy. Klasa powinna posiadać dwie metody:

- publiczną, bezargumentową eventLoop, w której ze strumienia wejściowego pobierane są linie napisów. Metoda ta powinna wywoływać kolejną metodę:
- prywatną, runCommand, przyjmującą pobraną linię, interpretującą ją jako wywołanie jednej z funkcjonalności klasy FileCommander, tj., pwd, cd <argument>, ls, find

<argument>. Metoda ta powinna wywołać odpowiednią metodę obiektu FileCommander i wyświetlić listę wynikową na strumieniu wejściowym. Do rozróżnienia między poleceniami użyj instrukcji switch.

Utwórz obiekt klasy FileCommanderCLI przekazując mu jako parametry standardowe strumienie wejścia i wyjścia.

Zadanie 5.

Zmodyfikuj wywołanie funkcjonalności "ls" tak, aby można było go wywołać z opcją "--color". Katalogi powinny wówczas nie być otoczone nawiasami ale mieć kolor niebieski. W tym celu zmodyfikuj metodę FileCommander::ls, aby przyjmowała obiekt funkcyjny Function<String, String>. Napisz dwie metody, które będzie można przekazać takim parametrem: jedną dodającą nawiasy, a drugą zmieniającą kolor. Skorzystaj z dołączonej klasy ConsoleColors.

Zadanie 6.

Zmodyfikuj wywołanie funkcjonalności "ls" tak, aby można było go wywołać z opcją "--filter=", gdzie po znaku równości może pojawić się dowolny ciąg znaków. Metoda FileCommander::ls powinna przyjmować ten napis jako parametr i zwracać wyłącznie pliki i katalogi zawierające ten napis.

Jeżeli opcje filter i color zostaną użyte jednocześnie, podciąg z filtra powinien być w nazwach plików i katalogów wyróżniony kolorem czerwonym.