

Stwórz aplikację konsolową, która będzie emulowała terminal. Aplikacja powinna pozwalać na interaktywne wpisywanie komend. Komendy, które należy zaemulować to:

- “ls” – wyświetlanie listy plików w określonym folderze
- “cd” – przejście do wskazanego katalogu
- “pwd” – wypisanie bieżącej lokalizacji
- “mkdir” – utworzenie w bieżącej lokalizacji katalogu o określonej przez użytkownika nazwie
- “cat” – wypisanie zawartości podanego pliku
- “grep” – filtrowanie zawartości podanego pliku i wypisanie tych wierszy które posiadają wpisany w pierwszym parametrze wzorec

Do tego celu należy:

Wykorzystać klasę Dictionary, w której należy składować obiekty komend (1 pkt). Każda komenda powinna być zapisana jako osobna klasa. Do celu obsługi komand należy wykorzystać mechanizm atrybutów C#. Klasy w projekcie posiadające implementację komendy należy oznaczyć atrybutem CommandAttribute przy czym w parametrze atrybutu powinien znajdować się ciąg znaków odpowiadający za wywołanie komendy (np. “ls”) (6 pkt). Podczas uruchamiania program powinien przeszukać Assembly w poszukiwaniu klas oznaczonych atrybutem i dla każdej klasy, którą znajdzie – utworzyć wpis w Dictionary taki, że ciągowi znaków odpowiadający za wywołanie komendy powinien przypisać instancję klasy komendy (należy tu stworzyć instancję klasy za pomocą mechanizmu refleksji) (8 pkt). Metodę, klasy którą należy wywołać w trakcie uruchomienia komendy przez użytkownika należy oznaczyć atrybutem ExecuteCommandAttribute (4 pkt). Zadbaj by każda komenda posiadała przynajmniej trzy opcje deklarowane przez przełączniki. Każdemu przełącznikowi powinien odpowiadać jeden property. W zależności czy przełącznik przyjmuje dodatkowo parametr ciągu znakowy czy nie powinien być deklarowany odpowiednio przez atrybut CommandOptionParameterlessAttribute albo CommandOptionWithParameterAttribute. W pierwszym wypadku property przełącznika powinien zwracać wartość bool (czy został wykorzystany przez użytkownika czy nie) w drugim wartość ciągu znakową (z parametrem przełącznika – argumentem podanym tuż za opcją z poziomu terminalu) (8 pkt). Dodatkowo nieużyte argumenty komendy wpisz na listę stringów przechowywaną przez pole oznaczone atrybutem OtherCommandArgumentsAttribute (5 pkt).