

SimpleShell

Alapvető instrukciók

A SimpleShell segítségével végezhetünk alapvető parancssoros műveleteket, a cmd.exe-hez hasonlóan.

Kibontás után a SimpleShell Windows platformokon, angol nyelven futtatható a `..\simpleshell\out\artifacts\simpleshell_jar\simpleshell.bat` címen.

Használata: bemenetként a felhasználó egyidejűleg egy sort adhat meg, amelyet a SimpleShell megpróbál utasításként értelmezni. Ha a bemenet megfelel a megadott formátumnak és a parancs nem ütközik egyéb hibába, a parancs végbemegy és erről értesíti a felhasználót is a parancssoron keresztül.

A bevitt parancsot a program az első space karakter által bontja szét. Első space előtti rész a parancsot választja ki, az utána lévő szöveg pedig a parancsnak átadandó paraméterként értelmezi. A paraméterek támogatják az idézőjel használatát, így pl. fájlnev megadásakor használhatunk space-t a parancs felbontás összezavarása nélkül. Nem minden esetben van szükség paraméterre.

Formátum: [parancs] [paraméter]

Például: `echo "ez egy teszt"`

Parancsok

- **help**: megjelenít egy angol nyelvű, segítséget tartalmazó szövegrészt.
- **echo [szövegrész]**: kiírja a paraméterként megadott szövegrészt (visszhang).
- **cd**: megjeleníti a jelenleg kiválasztott mappát.
- **cd [könyvtár]**: a program megkísérli az újonnan megadott mappába való belépést.
- **dir**: megjeleníti a jelenleg kiválasztott mappában található összes fájlt és mappát.
- **dir [könyvtár]**: megjeleníti a megadott mappában található összes fájlt és mappát.
- **cmd**: megnyitja a cmd.exe-t.
- **cmd [paraméterek]**: megnyitja a cmd.exe-t és az összes paramétert átadja neki.
- **perfmon**: megnyitja a Windows erőforrás-figyelő programját.

- **viewfile [útvonal]:** ha a megadott útvonal egy szöveges file-ra mutat, megjeleníti a tartalmát.
- **createfile [fájlnév]:** a SimpleShell főkönyvtárával megegyező mappába helyez ez új file-t, a megadott névvel ellátva - ezután soronként kéri be a file-ba szánt szöveget.
- **deletefile [útvonal]:** az útvonalon található file-t véglegesen eltávolítja.
- **date:** megjeleníti a jelenlegi rendszer által nyilvántartott, pontos időt.
- **extract_quotation [szövegrész]:** felismeri és kiírja a szövegrészben talált első idézetet (első két idézőjel közötti szöveg).
- **exit:** kilépés a SimpleShell-ből.

Fejlesztői leírás:

Main.java

1. **public static void main:** innen kezdődően fut le az alkalmazás, egy gyors operációs rendszer ellenőrzés után egy while loop kéri a parancs bevitelét. Ezt csak az exit paranccsal lehet megszakítani.
2. **public static String extractQuotation:** String-et kér be, majd ha a string tartalmaz idézetet (két idézőjel közötti szövegrész), visszatér azzal.

DirectoryHandler.java

1. **Változó:** File currentdir, File[] filesincurrentdir. Mind a kettő azt a célt szolgálja, hogy a program pontosan tudja, hogy mely mappában tartózkodik és azok milyen elemeket tartalmaznak.
2. **public DirectoryHandler:** Konstruktor metódus, amely beállítja az alapértelmezett mappát.
3. **Public String getCurrentDirectory:** Visszatér a jelenlegi mappa abszolút útvonalával.
4. **public void SetCurrentDirectory:** Bekéri egy mappa útvonalát, majd validitás után bele lép abba a mappába.
5. **private void fillFilesInCurrentDirectory:** Feltölti a filesincurrentdir változót a jelenleg kiválasztott mappában található file-okkal.
6. **private File[] filesOfDirectory:** Visszatér egy megadott mappában található file-okból készült tömbbel.
7. **public void outputStringVector:** Megjeleníti egy szöveglista (vector) összes elemét.
8. **public void outputFilesInCurrentDirectory:** Megjeleníti a jelenleg kiválasztott mappában található elemet.
9. **public void outputFilesInCustomDirectory:** Megjeleníti a megadott mappában található elemet.
10. **private void outputCustomDirectoryInfo:** outputFilesInCurrentDirectory és outputFilesInCustomDirectory-t összefűző, File tömböt megjelenítő metódus.

FileHandler.java

1. **Változói:** File currentfile. Segítségével nyomon követi a program, hogy mely file-al dolgozik
2. **public FileHandler:** Konstruktor metódus, amely beállítja az alapértelmezett file-t.
3. **public File getCurrentFile:** Egyszerűen visszatér a currentfile-al.
4. **public void setCurrentFile:** Bekéri egy file útvonalát, majd validitás után kiválasztja.
5. **public String getCurrentFileContents:** A jelenleg kiválasztott file szöveges tartalmát jeleníti meg.
6. **public void viewFile:** Bekér egy Main-ből származó inputString-et és egy fileHandler-t. Megkeresi az inputString-ben található útvonalat, majd megjeleníti az ott talált file szöveges tartalmát.
7. **public void createNewFile:** Bekér egy Main-ből származó inputString-et, amit az új file nevéként értelmez. Ezután belép a program egy "input" módba, ami addig kéri az új file-ba szánt sorokat, amíg a felhasználó EOF-ot ad. EOF megadása után mentésre kerül a file.
8. **public void deleteFile:** Bekér egy Main-ből származó inputString-et és egy fileHandler-t. Megkeresi az inputString-ben található útvonalat, majd véglegesen eltávolítja a címen található file-t.

Interpreter.java

1. **Változói:** DirectoryHandler directoryHandler és FileHandler fileHandler. Ezek a parancsok értelmezését segítik elő.
2. **public Interpreter:** Konstruktor metódus, inicializálja a változókat.
3. **public void outputCurrentDirectory:** Rövidítő metódus a jelenlegi mappa tartalmának megjelenítéséhez.
4. **public void outputHelpSheet:** Segítő szövegrész megjelenítése.
5. **public void interpret:** Bekér egy sornyi szöveget, validálja, majd végrehajtja az abban leírt utasítást. Szóközönként különíti el a parancs részeit, az első szóköz előtti részlet adja meg, hogy milyen funkciót hajtson végre.

ShellTest.java

1. **Test_Main_ExtractQuotation:** A Main osztály extractQuotation metódussal kezelt Stringeket teszteli.
2. **Test_DirectoryHandler:** DirectoryHandler osztály tesztelhető funkcióit futtatja és teszteli.
3. **Test_FileHandler:** FileHandler osztály funkcióit teszteli. [hiányos]
4. **Test_Interpreter:** Az Interpreter parancs felismerő funkcióit futtatja.