Faculty of Engineering and Technology
Electrical and Computer Engineering Department

Python Project

Name : Osaid Hasan Nur

ID : 1210733

Instructor : Dr. Abdel Salam Sayyad

T.A : Mazen Amria

Section : 5

Abstract

In this Project , we are going to implement a small Command Manual generation system , and we will use it to generate 20 command manual for a 20 command . The Project has 4 main functionalities, The first one is the command manuals generation , the result from this part is a Xml file called Commands.xml that contains the command manuals for 20 chosen command . The second functionality is the command manual verification . In this part it will verify the pre generated manuals , and check if they change after some time period . The third functionality is a search option , the user will have the ability to search for any command from the 20 chosen command and print all information about this command ,also , it will recommend some commands that may share functionality or name with the entered command . The forth and last operation is the command suggestion (recommendation) which explained above . The command generation and verification requires the existence of the xml file , but the search and recommend operations are independent , and the user can use them in any time without limitations .

About the code

My code consist of 4 classes , CommandManual, XmlSerializer CommandManualGeneration , and the main class .

CommandManual class : this class implements the single command manual as a one object , it includes the required information about the command , the only way to contact with class is the setters and getters ,the data inside the class are private .

XmlSerializer : this class contains two static methods , the serializer , and the deserializer , they have opposite functionality , the serializer will take a list of command manuals , and return them into an xml file using build it library called Element Tree and the deserializer will parse the xml file and return it into list of command manuals .

CommandManualGeneration : This class contains all important functions we need , and all are static functions because we don't want to make an object at every operation which will have a bad effect on the memory . The class have the function generate , which will read the commands file , and extract command names from it , then ,it will check for the commands if they actually loaded into the list of commands inside the class , if there is a problem in reading the commands it will show a message for this problem , and it will terminate the execution , if the check operation succeed it will generate the parts of the command manual , at the last of this process , we will have a list of command manuals, the are from the type **CommandManual** , then , we pass this list to the xml serializer to convert the data to xml file called Commands.txt .

The second function is verify , it will generate a test xml file using the function generate , then , we will deserialize the two files , and we will compare the data included in , and we will print the differences to the terminal .

The third function is search , it will take a command , and search for it , if found , it will print it’s data , and if not , it will display a message that the command is not found .

The last

Main class : This class contains the main menu , to let the user selects the operation he wants

The deep explanation is provided with the code

Execution

Display the main menu

And select the option 1 :

```
osaid@Ubuntu:~/Documents/VS Code$ python -
733.py"
* * * * * * * * * * * * * * * * *
*  Choose the option (1-5): *
*  1- generate              *
*  2- verify                *
*  3- search                *
*  4- recommend             *
*  5. Exit                  *
* * * * * * * * * * * * * * * * *
||| choice -> 1
```

The xml file is generated successfully :

commands.txt | OsaidNur_1210733.py | Commands.xml

project2 > Commands.xml

```
1  <Manuals><CommandManual><CommandName>ls</CommandName><CommandDescription>
2      List information about the FILEs (the current directory by default). Sort
3      phabetically if none of -cftuvSUX nor --sort is specified.</CommandDescri
4  commands.txt
5  OsaidNur_1210733.py
6  TEST_.txt
7  </Example><RelatedCommands>else
8  false
9  alsa
10 alsa-info
11 alsactl
12 lspcmcia
13 alsabat-test
14 update-shells
15 lsmod
16 tclsh8.6
17 lspci
18 lsblk
19 lsipc
20 lsmem
```

select the option 2 to verify the generated xml file :

```
  Xml file generated successfully ...
* * * * * * * * * * * * * * * * *
*  Choose the option (1-5): *
*  1- generate              *
*  2- verify                *
*  3- search                *
*  4- recommend             *
*  5. Exit                  *
* * * * * * * * * * * * * * * * *
||| choice -> 2
```

display some changes in the files :

```
The Example for the command ls have been changed...
********** Before **********
>>$ ls
commands.txt
Commands.xml
OsaidNur_1210733.py
TEST_.txt

********** After ***********
>>$ ls
commands.txt
OsaidNur_1210733.py
TEST_.txt
```

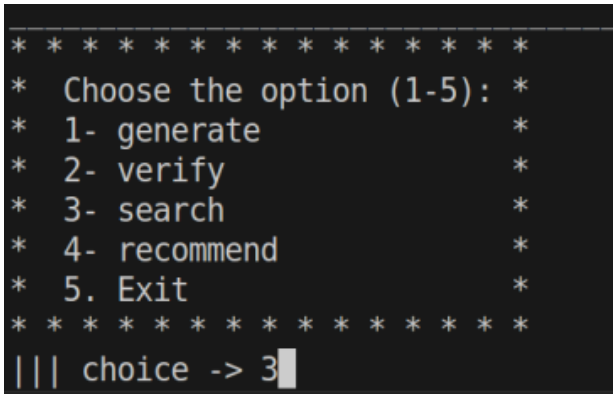
```
The Example for the command date have been changed...
********** Before **********
>>$ date
Wed 31 Jan 2024 11:49:20 PM EET

********** After ***********
>>$ date
Wed 31 Jan 2024 11:46:44 PM EET
```

```
The Example for the command ps have been changed...
********** Before **********
>>$ ps
    PID TTY          TIME CMD
  28430 pts/8    00:00:00 bash
  29616 pts/8    00:00:00 python
  30050 pts/8    00:00:00 ps

********** After ***********
>>$ ps
    PID TTY          TIME CMD
  28430 pts/8    00:00:00 bash
```

select the option 3 to search for a command :



Display the information about the command , and also display the recommended Commands for this command :

```
||| choice -> 3
... You selected search ...
Enter the command to search for : who
////////////////////////////////////////////////////////////////////////
// Command name     : who
// Description      :
    Print information about users who are currently logged in.
// Version          : 8.32
// Example          :
>>$ who
osaid    tty2         2024-01-31 20:40 (tty2)

// Related Commands :
whoopsie
whoopsie-preferences
whoami
who
whoopsie
whoopsie-preferences
whoami
who
////////////////////////////////////////////////////////////////////////
Recommended Commands :
 users
 loggedon
 w
 finger
 getent passwd
```

Select the option 4 for recommend :

```
* * * * * * * * * * * * * * * * *
*  Choose the option (1-5): *
*  1- generate              *
*  2- verify                *
*  3- search                *
*  4- recommend             *
*  5. Exit                  *
* * * * * * * * * * * * * * * * *
||| choice -> 4
```

It will display the recommended Commands for this command :

```
||| choice -> 4
... You selected recommend ...
Enter a word or command name: mv
Recommended Commands :
 move
 rename
 cp
 link
 ln
```

This is another example :

```
||| choice -> 4
... You selected recommend ...
Enter a word or command name: line
Recommended Commands :
 head
 tail
 less
 wc
 rev
 factor
 uniq
```

Exit the program :

```
||| choice -> 5
Exiting...
osaid@Ubuntu:~/Documents/VS Code/project2$
```