?

Tτ

6

# Design In-Memory File System

Try to solve the Design In-Memory File System problem.

# We'll cover the following Statement Examples Understand the problem Try it yourself

### **Statement**

Design an in-memory file system. The skeleton for the class FileSystem is provided to you. Simulate the following functions:

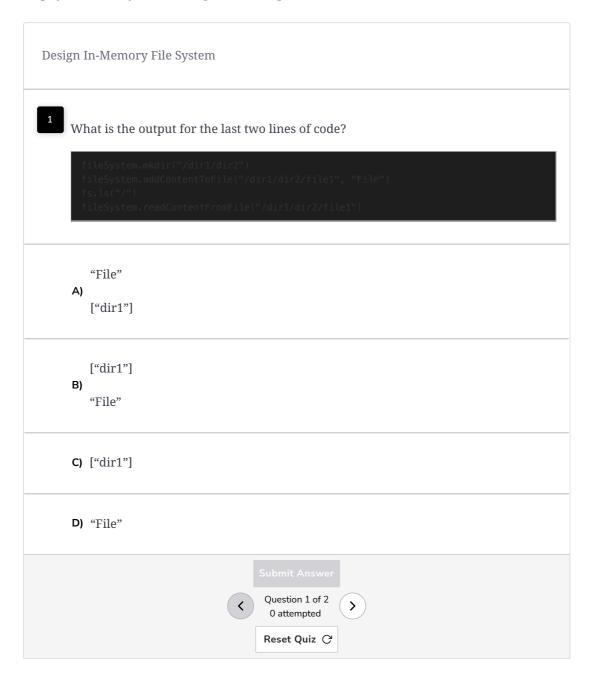
- ls: You're given a string representing a path. If it's a file path, return a list that only contains the file's name. If it's a directory path, return the list of file and directory names in this directory. Your function should return the output (file and directory names together) in lexicographical order.
- mkdir: If you have a directory path that does not exist, make a new directory according to the given path. The function should create all the middle directories in the path if they don't exist. This function's return type is void.
- addContentToFile: You're given a file path and file content in string format. If the file doesn't exist, create that file containing the given content. If the file already exists, append the given content to the original content. This function's return type is void.
- readContentFromFile: Given a file path, return its content in string format.

## **Examples**

Operation	Output	Explanation
FileSystem fs	NULL	The constructor call returns
fs.ls("/")		nothing.  The directory / contains noth-
		ing, so we return an empty list.
fs.mkdir("/dir1/dir2/dir3")	NULL	Create a directory dir1 in /.
		Then, create a directory dir2 in
		the directory dir1. Lastly, cre-
		ate a directory dir3 in the di-
		rectory dir2.
<pre>fs.addContentToFile("/dir1/ dir2/dir3/file1", "File")</pre>	NULL	Create a file file1 with the
		contents File in the directory
		/dir1/dir2/dir3.
fs.ls("/")	["dir1"]	Only directory dir1 exists in
		the <mark>/</mark> directory.
<pre>fs.readContentFromFile("/di</pre>	"File"	Output the file content.
r1/dir2/dir3/file1")		

# Understand the problem

Let's take a moment to make sure you've correctly understood the problem. The quiz below helps you check if you're solving the correct problem:



# Try it yourself

Implement your solution in the following coding playground:

```
👙 Java
                                                                                     ■ C
usercode > FileSystem.java
   1 import java.util.*;
   2 public class FileSystem {
   3
         public FileSystem() {
              // Write your code here
   5
        public List <String> ls(String path) {
   6
   7
             // Write your code here
   8
              List < String > files = new ArrayList < > ();
   9
              return files;
          }
  10
  11
         public void mkdir(String path) {
```

?

Tr

6

```
13
                 // Write your code here
   14
   15
            public void addContentToFile(String filePath, String content) {
   16
            public String readContentFromFile(String filePath) {
   20
   21
                 // Write your code here
                 return "";
   22
   23
            }
   24 }
                                                                         Powered by Al
                                                                                                    Submit
 Test Cases
               Results
   Case 1
               Case 2
                          Case 3
 Input #1
   ["FileSystem","ls","mkdir","addContentToFile","ls","readContentFromFile"]
   [[],["/"],["/a/b/c"],["/a/b/c/d","hello"],["/"],["/a/b/c/d"]]
                                          Design In-Memory File System
                                               ∵Ö Need a Hint?
 ← Back
                                                                                                     Next ->
Same Tree
                                                                                                 Design File System
                                                                                           ✓ Mark as Completed
```