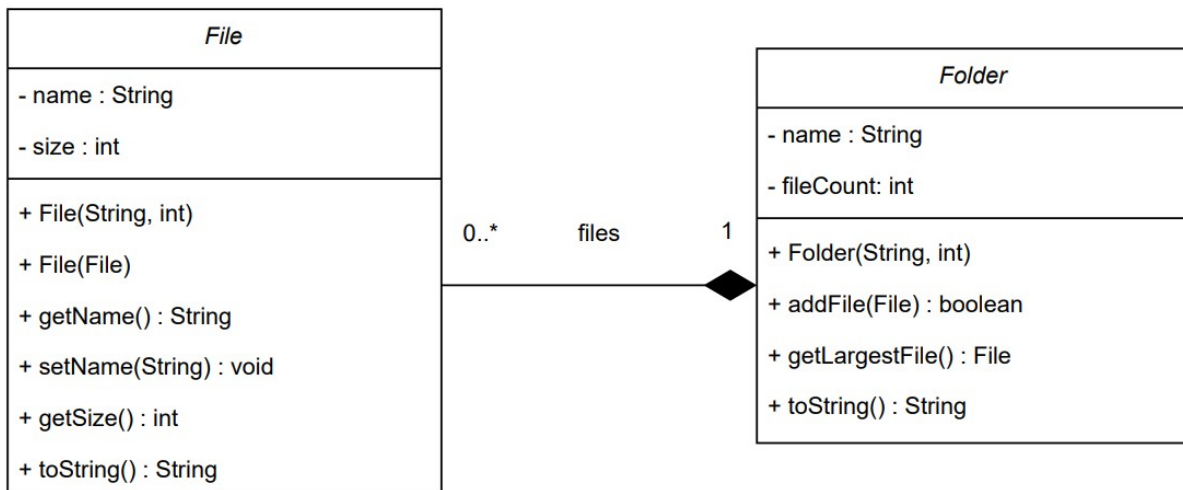


We want to write a program that manages simple files and folders using the following UML diagram:



Class **File**:

- Instance Attributes:
 - **name**: the name of the file
 - **size**: the size of the file in bytes
- Methods:
 - **File(name:String, size:int)**: constructor
 - **File(File f)**: copy constructor
 - **getName()**: returns the value of **name**
 - **setName(name:String)**: assigns the value of **name**
 - **getSize()**: returns the value of **size**
 - **toString()**: returns a string representation of the file in the following format:
 - **File: name, size: sizeB**

Class Folder:

- Instance Attributes:
 - *name*: name of the folder
 - *files*: array of File objects
 - *fileCount*: number of File objects in *files*

- Methods:
 - *Folder(name:String, size:int)*: constructor that receives the size of *files* array
 - *addFile(f:File)*: adds *f* to the first available space of *files* array if there's space and there isn't a file in the folder with the same name and returns true. Otherwise, returns false.
 - *getLargestFile()*: returns the first largest file in size in *files* if possible, or null otherwise
 - *toString()*: returns a string representation of the folder in the following format:
 - **Folder name (fileCount):**
 - **File: name, size: sizeB**
 - **File: name, size: sizeB**
 - ...
 - -----

Exercise 1: Write classes **File & Folder**

Exercise 2: Write class **FolderTest** that has a main method to test the functionalities of classes **File & Folder**:

- Create a folder of size 10
- Create and add three files to the folder (see sample run)
- Print the folder
- Modify the name of the largest file in the folder
- Print the folder

Sample run:

```
Folder src (3):  
File: Cart.java, size: 675B  
File: Item.java, size: 428B  
File: ContactTest.java, size: 703B
```

```
-----  
Folder src (3):  
File: Cart.java, size: 675B  
File: Item.java, size: 428B  
File: CartTest.java, size: 703B  
-----
```