Hans Alan Whiburn Haugen

Find my Programming-I repository and Compulsory-III branch here: https://github.com/alanhaugen/Programming-I/tree/Compulsory3

Implementation UML diagrams for the classes you have used

I have made three classes.

- Folder
- File
- Info

The relationship looks like this: (See figure 1)

Info contains name, size and date information.

A brief paragraph describing where you feel you are with programming, what do you feel most comfortable doing from what you've learned this year, and what you feel you need to improve on the most from what you've learned this year

I think I am very good at the basics. I have learnt new things about UML.

I have also played a bit with flex and bison for making calculators which does PEMDAS correctly. I want to look into recreating the Silicon Graphics program fsn, as seen in Jurrassic Park.

I feel I need to learn more about UML and try to understand how such diagrams can be useful for programmers.

I tried using nomnoml to create the UML, but I still know so little about UML I am not sure how useful the diagram ends up being https://nomnoml.com/.

Documentation

I have used Doxygen to generate pretty documentation (XML).

I have made the documentation available on my website here: https://alanhaugen.github.io/docs.

Note the program implements a typical unix file system. The following commands are available:

- **touch** (to create files)
- **mkdir** (to create directories)
- **cd** (to change directory)
- **ls** (to list your files and directories)

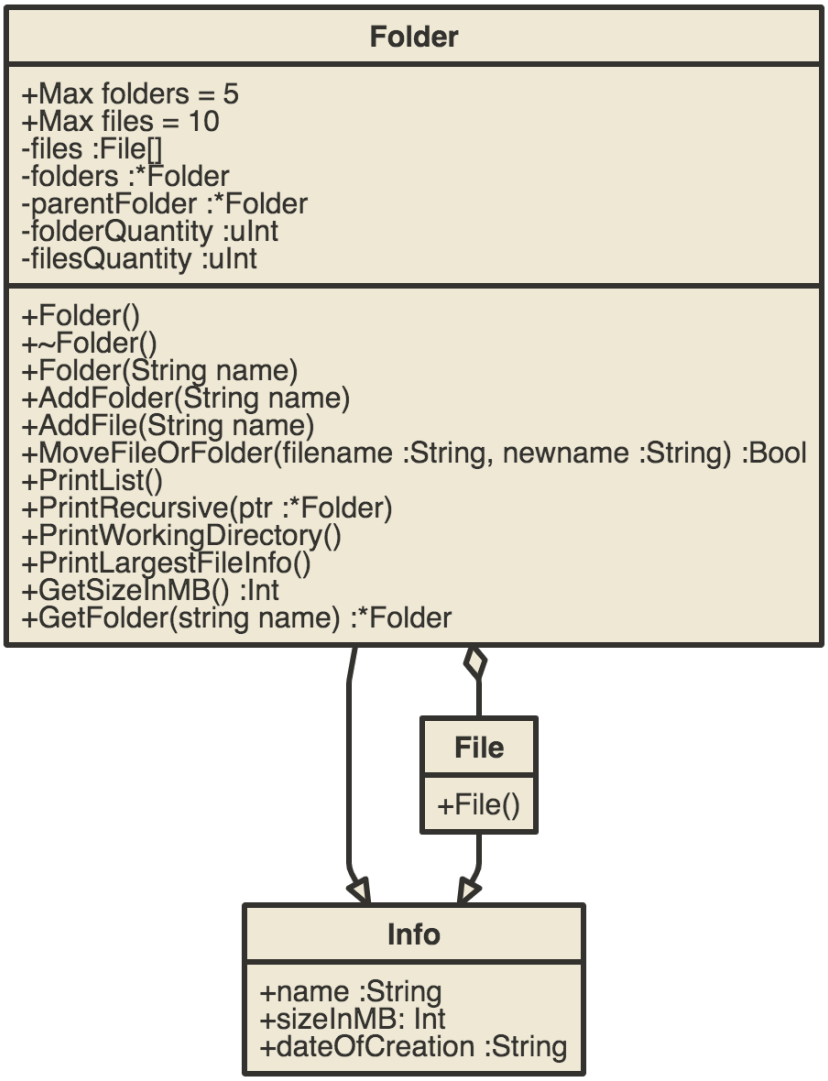
Folder
+Max folders = 5
+Max files = 10
-files :File[]
-folders :*Folder
-parentFolder :*Folder
-folderQuantity :uInt
-filesQuantity :uInt
+Folder()
+~Folder()
+Folder(String name)
+AddFolder(String name)
+AddFile(String name)
+MoveFileOrFolder(filename :String, newname :String) :Bool
+PrintList()
+PrintRecursive(ptr :*Folder)
+PrintWorkingDirectory()
+PrintLargestFileInfo()
+GetSizeInMB() :Int
+GetFolder(string name) :*Folder
File
+File()
Info
+name :String
+sizeInMB: Int
+dateOfCreation :String

Figure 1: UML

- **mv** (to change the name of a file or directory)
- **pwd** (to print where you are in the file system)

In addition:

- **sort** (to find the biggest file)
- **quit** (to close the program)
- **help** (to print instructions on how to use the program)

To create a folder, type:

- mkdir
- folder

To list the contents of the folder named folder, type:

- cd
- folder
- ls

I personally really enjoy the arguments found by googling "Getters and setters are evil' '. I like to use public as often as possible to keep encapsulation and OOP valid in my program and only use setters OR getters (and private variables) when it improves a computer program.