Emre Karataş

22001641

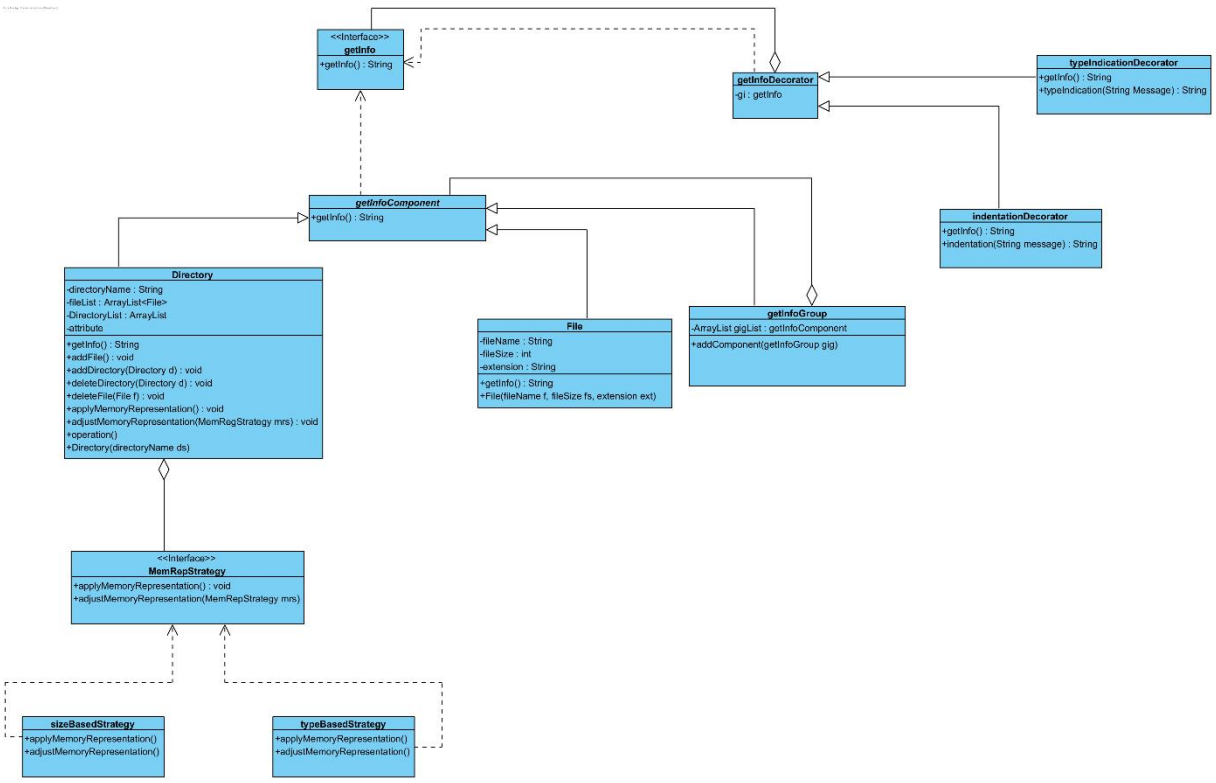
CS 319-03

CS 319

DESIGN PATTERNS HOMEWORK

1. Class Diagram:

Class diagram of the homework can be seen below. Note that main.java is not included in the diagram, since it is a test code.



Design Patterns:

For three different parts of the homework, the different design pattern is applied.

- Part I: Composite Design Pattern
- Part II: Decorator Design Pattern
- Part III: Strategy Design Pattern

Composite Design Pattern:

In part I, composite design pattern is used to apply getInfo() method on different objects. Since the system are made up from directories and files, and both of these objects have getInfo() method, we put them some kind of a container “getInfoGroup” to do the getInfo operation. We added them on the component so that depending on the type, necessary getInfo () method is called.

Decorator Design Pattern:

Decorator design pattern is used in Part II. Since we want to manipulate the output coming from getInfo() method, and that manipulation should work in different and/or joined ways, necessary adjustments are needed. We created a new interface and that indention () and typeIndication () can work together. Note that by interface, we can work these two methods independently also.

Strategy Design Patern:

Strategy design pattern is applied in Part III. We want to see different outputs in the runtime in Part III. To do that, we created a new interface MemRepStrategy which aggreates from Directory (it uses Directory objects). That interface implements two methods, as mentioned in the question: sizeBasedStrategy and typeBasedStrategy. Applying which method should work is chosen in the runtime, therefore to see the output coming from Directory object, we are applying Strategy Design Pattern.