CS 319

Object Oriented Programming Languages

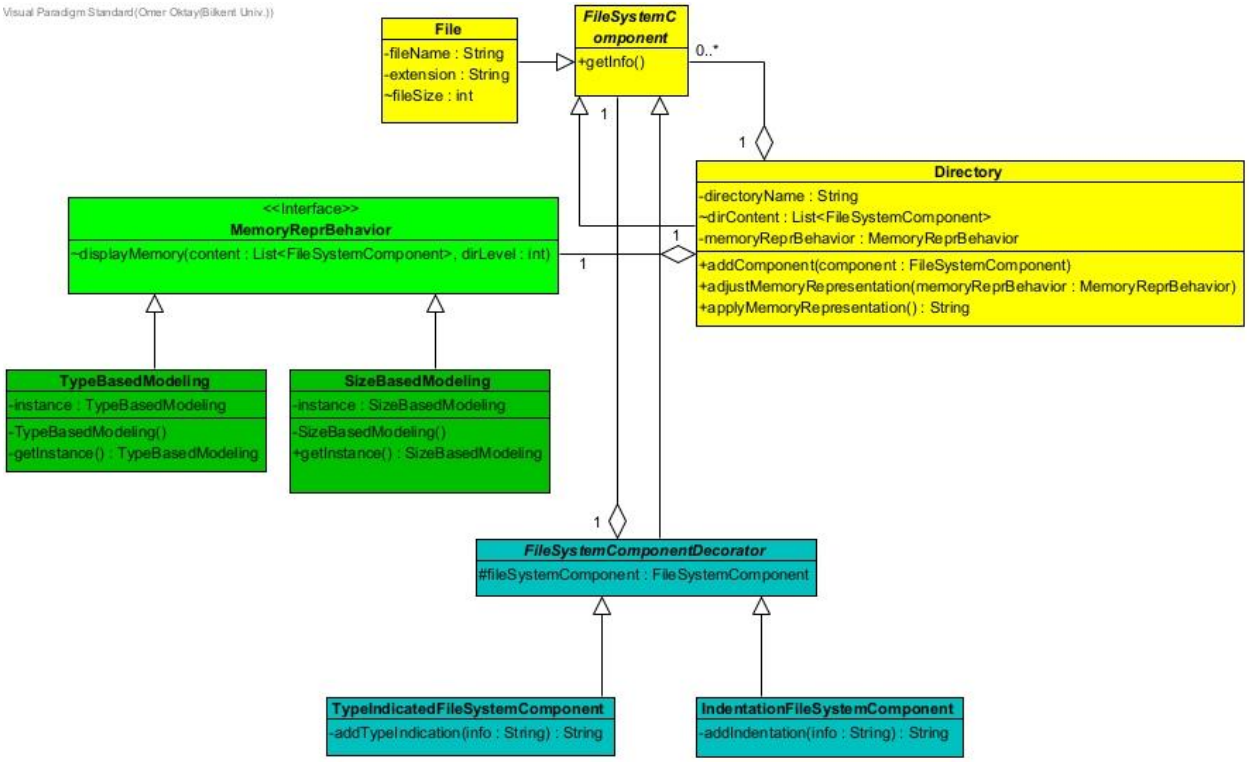
Design Patterns Take Home Exam

Section 2

Ömer Oktay Gültekin

21901413

2022-2023 / Fall Semester



Note: Constructors and overridden methods are not shown as in the tutorial.

Part 1:

To implement a file system, a Composite design pattern should be used. The yellow classes use the Composite pattern since a directory can “compose” of files or subdirectories. The Composite pattern was implemented by inheriting the File and Directory classes from FileSystemComponent class and aggregating the Directory from the FileSystemComponent. Therefore, instead of storing references of both File and Directory classes separately in the Directory Class, the Directory class directly stores the list of the FileSystemComponent, which is the advantage of using the Composite pattern.

Part 2:

The Decorator design pattern is used for “decorating” the output of the parent class method, getInfo() method of the FileSystemComponent, statically. The classes with blue

color show the Decorator design pattern. The Directory class represents concrete components, whereas TheIndicatedFileSystemComponent and IndentationFileSystemComponent classes represent concrete decorators of the Decorator design pattern. According to GoF, Decorator and Composite design patterns are often used together by making them extend from a common parent class [1]. Therefore, the class diagram correctly represents the combination of the Decorator and Composite design patterns. When decorators are used, the client needs to create a decorator and give the components that will be decorated statically in the client code. Thus, a component does not hold any reference to the decorators, but the decorator will hold the reference of the component, which is the fileSystemComponent attribute of the abstract decorator. The Decorator design pattern allows more than one decorator usage at any time.

Part 3:

The Strategy design pattern is used for choosing a strategy at runtime by calling the two methods, adjustMemoryRepresentation() and applyMemoryRepresentation(). The green classes indicate the usage of the Strategy in the class diagram. For implementation, a MemoryReprBehavior field is stored in the Directory class, and the previously mentioned two method helps to select the correct strategy for the need. The strategy classes override the same method for the strategy design pattern to work properly. Unlike Decorator, the Strategy design pattern only allows one strategy to be selected at any time. Also, the concrete strategies, TypeBasedModeling and SizeBasedModeling, (dark green classes), use Singleton design pattern since their instances do not require to have an identity, i.e., they will not be changed with the methods that they have. Therefore, it is logical to only create one instance of them by making the constructors private and providing a method to get the instance for the user.

References:

[1] Reshma, ReshmaReshma 4166 bronze badges, Nikolas CharalambidisNikolas Charalambidis 37.6k1313 gold badges9797 silver badges167167 bronze badges, and ChristopheChristophe 64.8k66 gold badges7272 silver badges129129 bronze badges, “Decorator design pattern with composite design pattern,” *Stack Overflow*, 01-Nov-1968. [Online]. Available: https://stackoverflow.com/questions/69286561/decorator-design-pattern-with-composite-design-pattern. [Accessed: 09-Dec-2022].