H/W Assignment #2

Student(s): James Yen

Class: Senior Project Laboratory

Problem 5.1

They are similar in that both contain multiple components that do one thing very well, but they are different in that services stand alone and can be queried over a network, whilst components usually interface directly with each other within one single program.

Problem 5.2

I believe a component-based architecture would since it would be easy to think of the program as consisting of different components. One function acts as the board and sets it up, another function makes choices for the AI, a third function checks game logic and makes sure no rules are broken and so on and so forth. Since no network is needed and the program resides on a single phone, component-based seems to be the way to go. One could argue that a monolithic architecture would be superior, but dividing up the application into components is more intuitive.

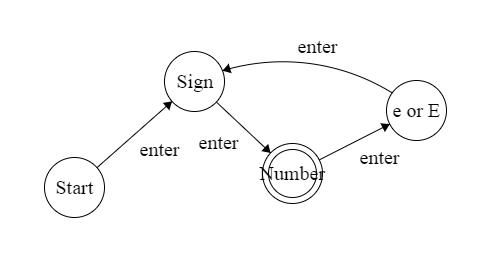
Problem 5.4

Client/server is the way to go in this case. Service-oriented might be applicable, however, service-oriented architecture doesn’t allow for the synchronization of a single timeline in a chess game to take place. Service-oriented architectures would apply when multiple different “asynchronous” clients would query an app and receive their needed data. For example, an app that performs miscellaneous metric conversions and language translation services would fit nicely with a service-oriented architecture. One service translates, another converts weights, and so on and so forth. In our case, we want one player to make a move, then the second player, and so on and so forth, we are keeping both players in sync. We therefore need two clients, but one server, that acts as the master timeline for the game. I argue, therefore, for a client/server architecture.

Problem 5.6

I think a simple on-disk file database would do, there’s no need for a relational database. As long the application stores temporary drafts of work on the disk while the app is open, and continues to store backups even after the app is closed, that should be all that’s necessary.

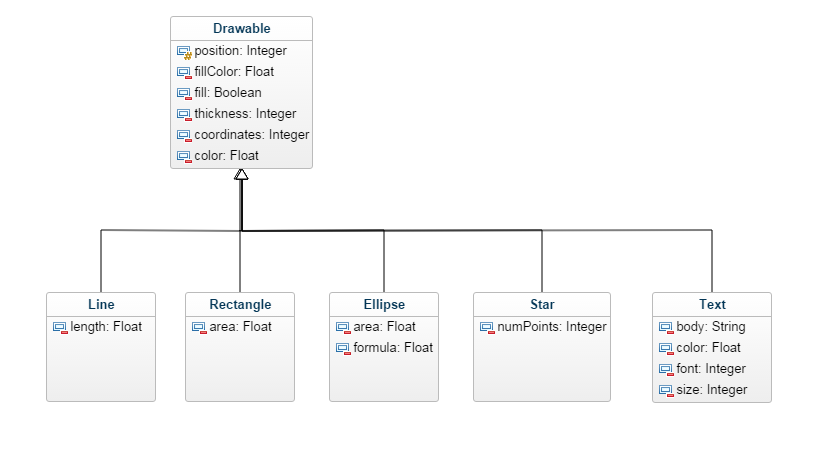
Problem 5.8



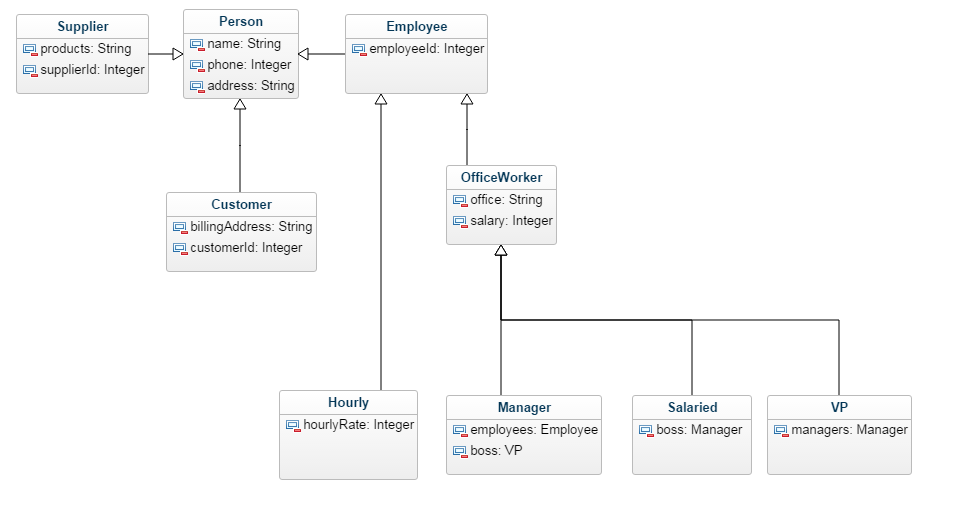
Problem 6.1

All classes share a position property. I can always move the Line, the Rectangle, the Ellipse, the Star, and the Text. All classes have an edit property. I can always move one or both ends of the line to wherever I want. For the Rectangle and Ellipse, I can move and expand any point, but the generic shape remains immutable. The same goes for the Star, except the Star cannot be mutated, but resized only. I can always edit the content of the Text, as well. The Text has a content property that none of the geometric classes have. The content itself has several properties, such as font, color, and size. All classes have a color property, except for the Text class. The color for the Text class is within its content property.

Problem 6.2



Problem 6.3



Problem 6.6

Do later…