Andrea Carver

CMSI 402

B.J. Johnson

10 February 2018

*Homework 2*

*Problem 5.1*

With component-based architecture, the system is envisioned as several components which organize the code into discrete functions. The components can provide services to one another. On the other hand, the ‘components’ in service-oriented architecture are more completely separated- they are completely self-contained. The service-oriented architecture is more optimized for web deployment.

*Problem 5.2*

A component-based architecture would be more appropriate than a services-oriented architecture since the phone application is not connected to any other phones or servers. For the same reasons, a monolithic approach would be wiser than a client/server approach. Additionally, if the computer opponent is implemented as a simple A.I., it could be envisioned as rule-based architecture.

*Problem 5.4*

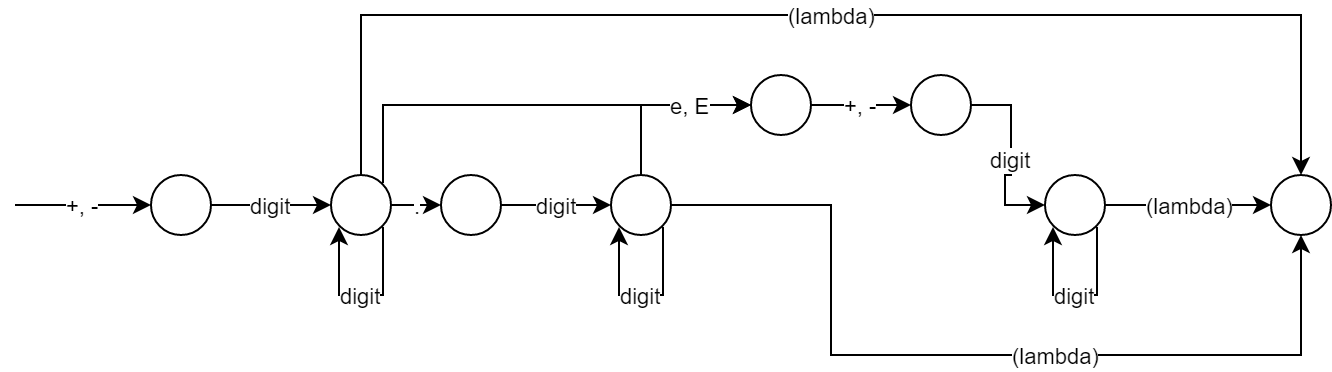
A client/server architecture would be more fitting to this particular application, allowing each of the players to be clients and the game to be stored on the server. A services-oriented architecture would be better than a component-based architecture, since the game is to be deployed on several platforms. If core game features are implemented independently from their target platforms, this could help to minimize time/effort necessary to implement new features or extend to more platforms.

*Problem 5.6*

The ClassyDraw application does not require several different users, simultaneous access, or different views, so a simple text file data storage would be sufficient. Sketches could be stored as a list of the various oval, rectangle, and line objects. Data auditing would be unnecessary as the application is single-user. Moving drawing data into a data warehouse would only be necessary if the developer wanted to train an A.I., similar to Google’s [Quick Draw](https://quickdraw.withgoogle.com/).

*Problem 5.8*

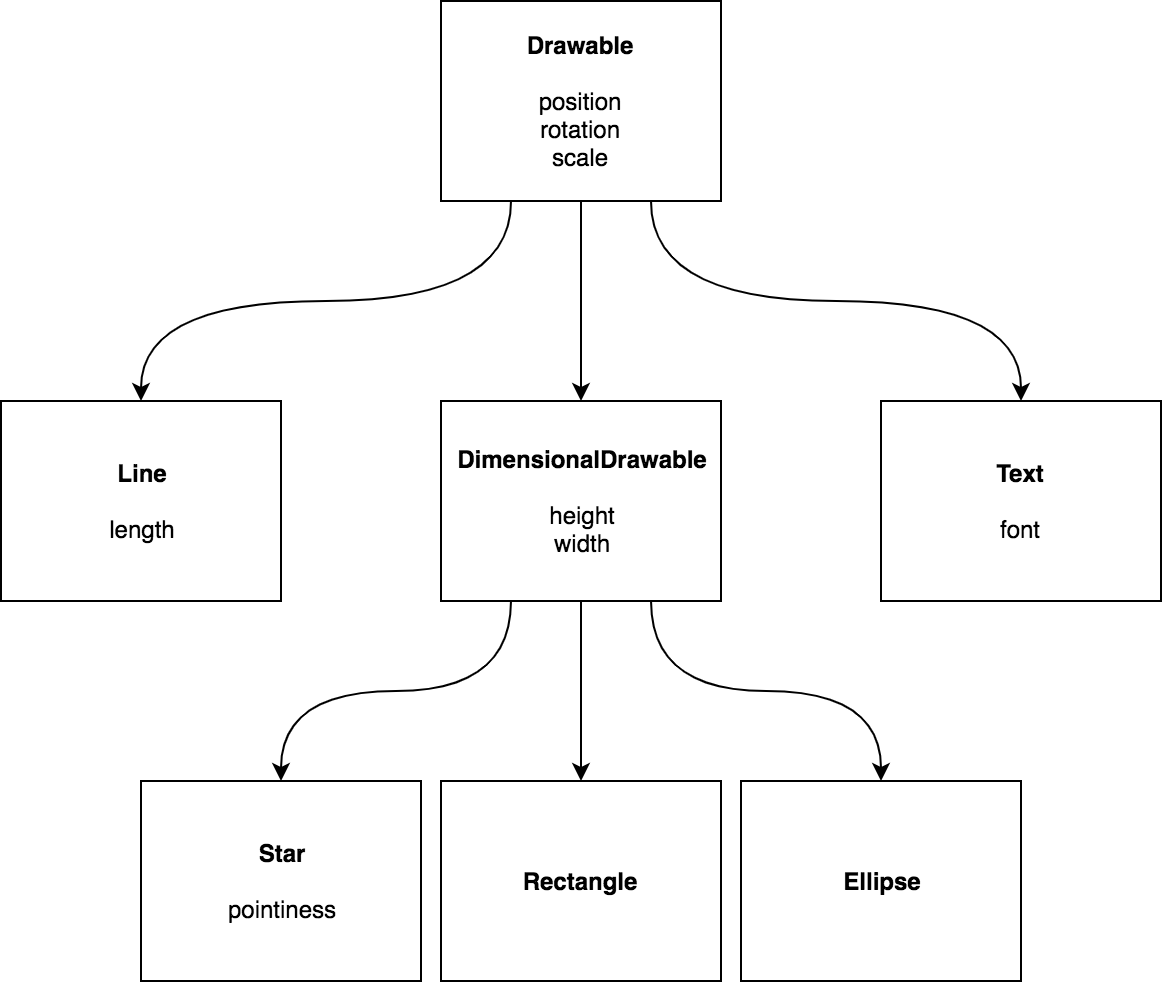
Please interpret “a, b” to mean “a” or “b”. Please interpret “(lambda)” as the empty string.



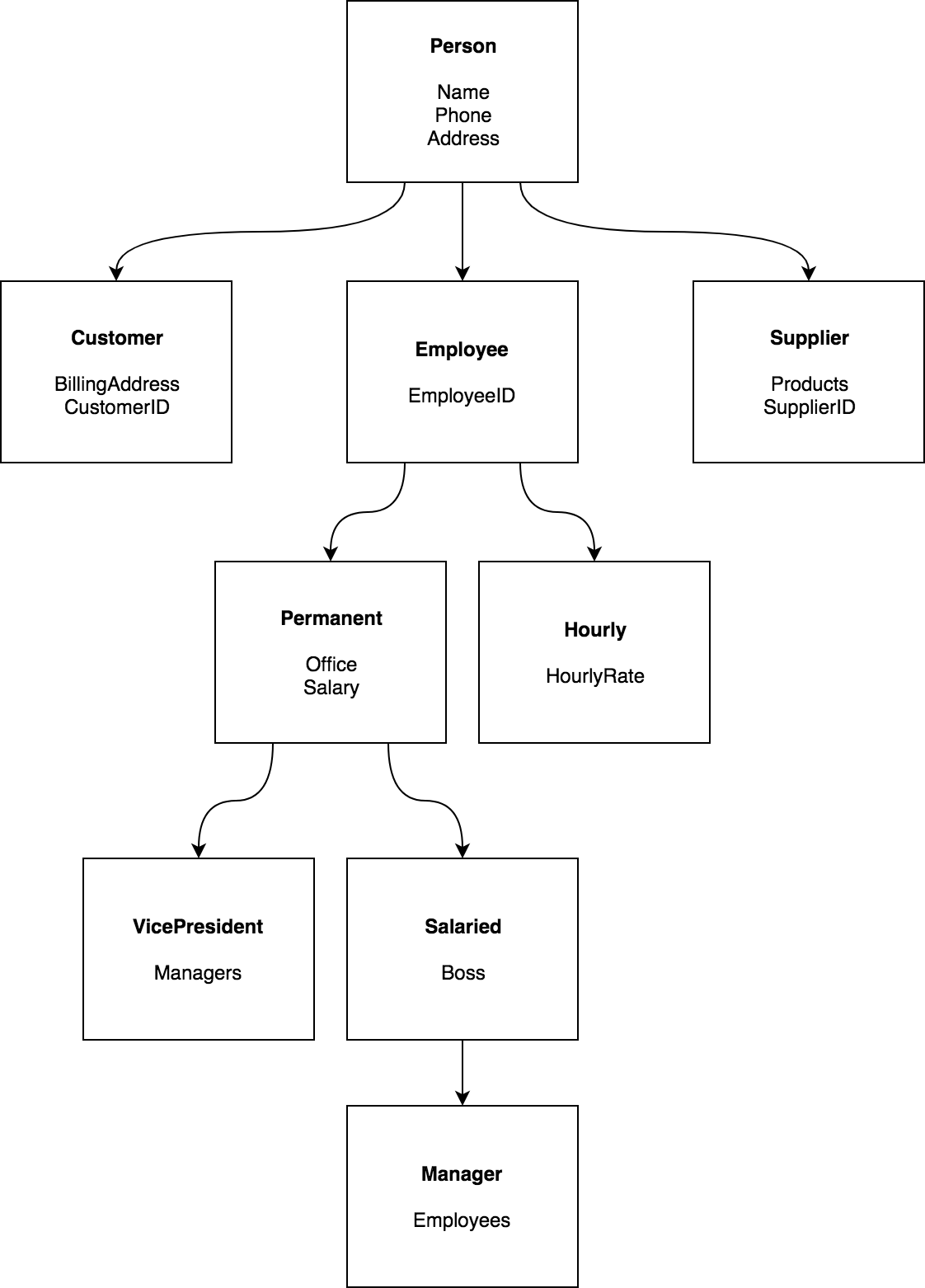
*Problem 6.1*

Line, Rectangle, Ellipse, Star and Text each share properties such as position, rotation, scale, and potentially fill color, outline color, and outline thickness properties. The individual properties for each class could include length for Line, height and width for Rectangle, height and width for Ellipse, height and width and pointiness for Star, and font for Text. Only the Rectangle, Ellipse, and Star class share the height and width properties. The properties shared amongst all classes should be included in the Drawable class, while properties shared by Rectangle, Ellipse, and Star should be included in a DimensionalDrawable class that is a subclass of Drawable and a superclass of Rectangle, Ellipse, and Star.

*Problem 6.2*



*Problem 6.3*



*Problem 6.6*

