Abdul Alzaid
CMSI 402
HW2
B.Johnson

Problem 5.1, Stephens page 116

The two architectures I believe are similar. However, the difference is that the component-based architecture regards pieces of the system as loosely coupled components that provide services for each other. The pieces communicate directly instead of across a network, because they are all contained within the same executable program.

On the other hand service-oriented architecture is similar except the pieces are implemented as services, often running on separate computers communicating across a network, sometimes as found in web services. A service is a self-contained application that runs on its own and provides some services for the clients.

Therefore, the pieces are more separated in a service-oriented architecture.

Problem 5.2, Stephens page 116

Since the application is simple and self-contained, there's no need for remote services or database. Thus, client-server, component-based, and service-oriented architectures are too much for this simple application. Therefore, a monolithic architecture and a data-centric approach would probably work well.

Problem 5.4, Stephens page 116

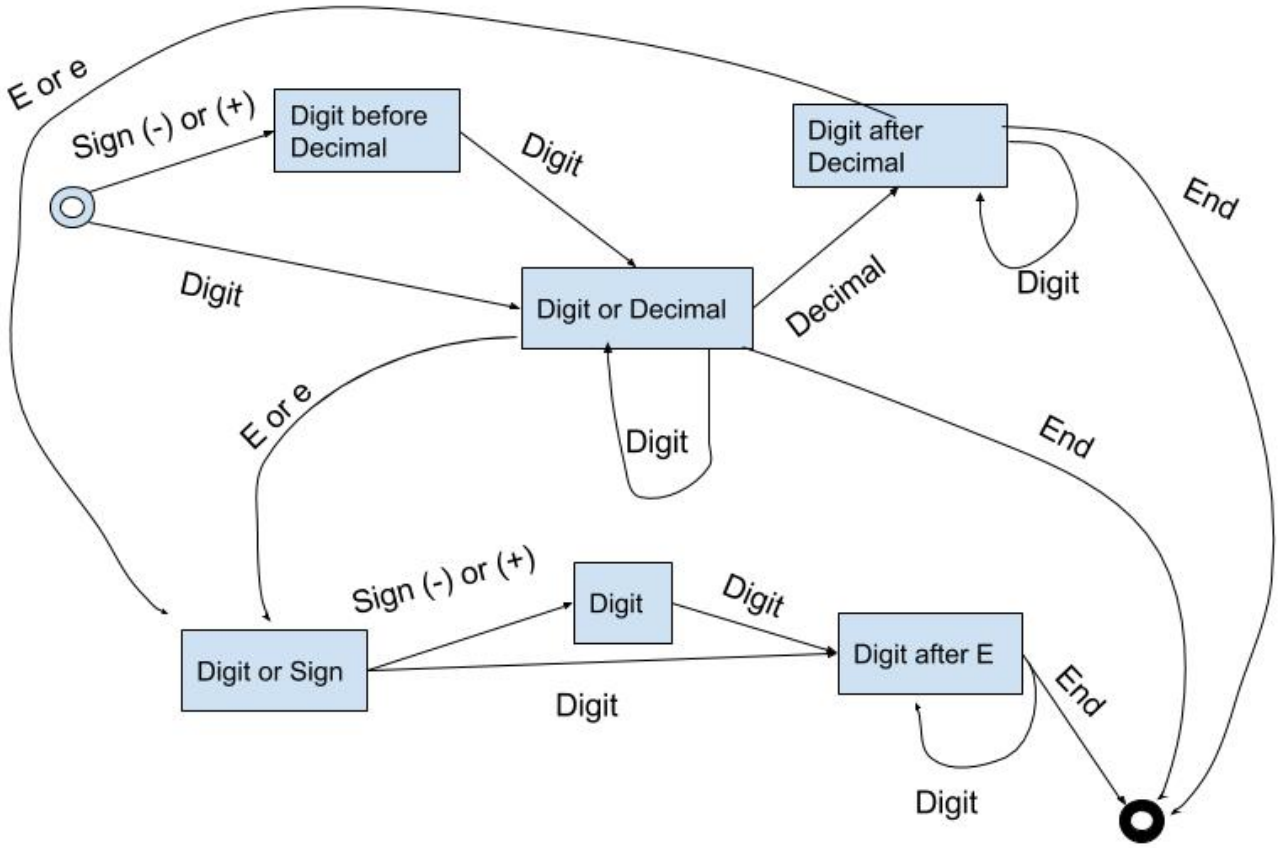
Since the UI is basically the same, the only thing is that the program needs to exchange information with another instance of the program across the Internet, so there's no need for a computer component, thus, the program doesn't need distributed pieces. For those reasons, we could use web services to allow the two programs to communicate over the Internet. That would make the application a monolithic rule-based (data-centric) service-oriented application.

Problem 5.6, Stephens page 116

For the DB structure the ClassyDraw application can store each drawing in a separate file, so it doesn't need much of a database. Managing files can be done using Operating System tools that are out there.
For the maintenance the program could create a temporary file while the user is editing a drawing. Then if something goes wrong and the app crashes, it could ask the user if it should restore the temporary file the next time the application starts.

Problem 5.8, Stephens page 116

Aoh Dr. Dorin you know I hated that class!! But let's show Mr. Johnson some diagrams and hopefully they won't disappoint.

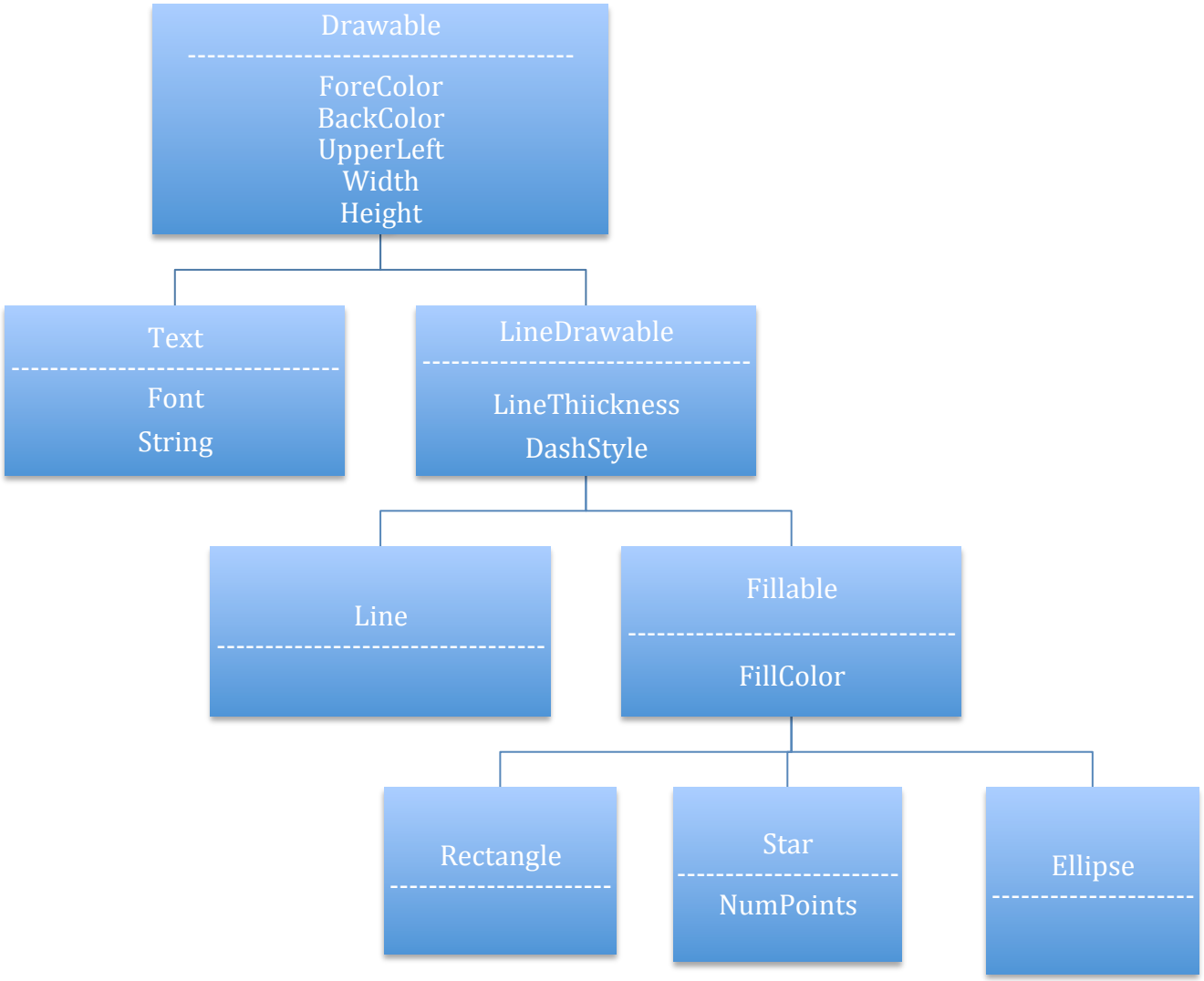


Problem 6.1, Stephens page 138

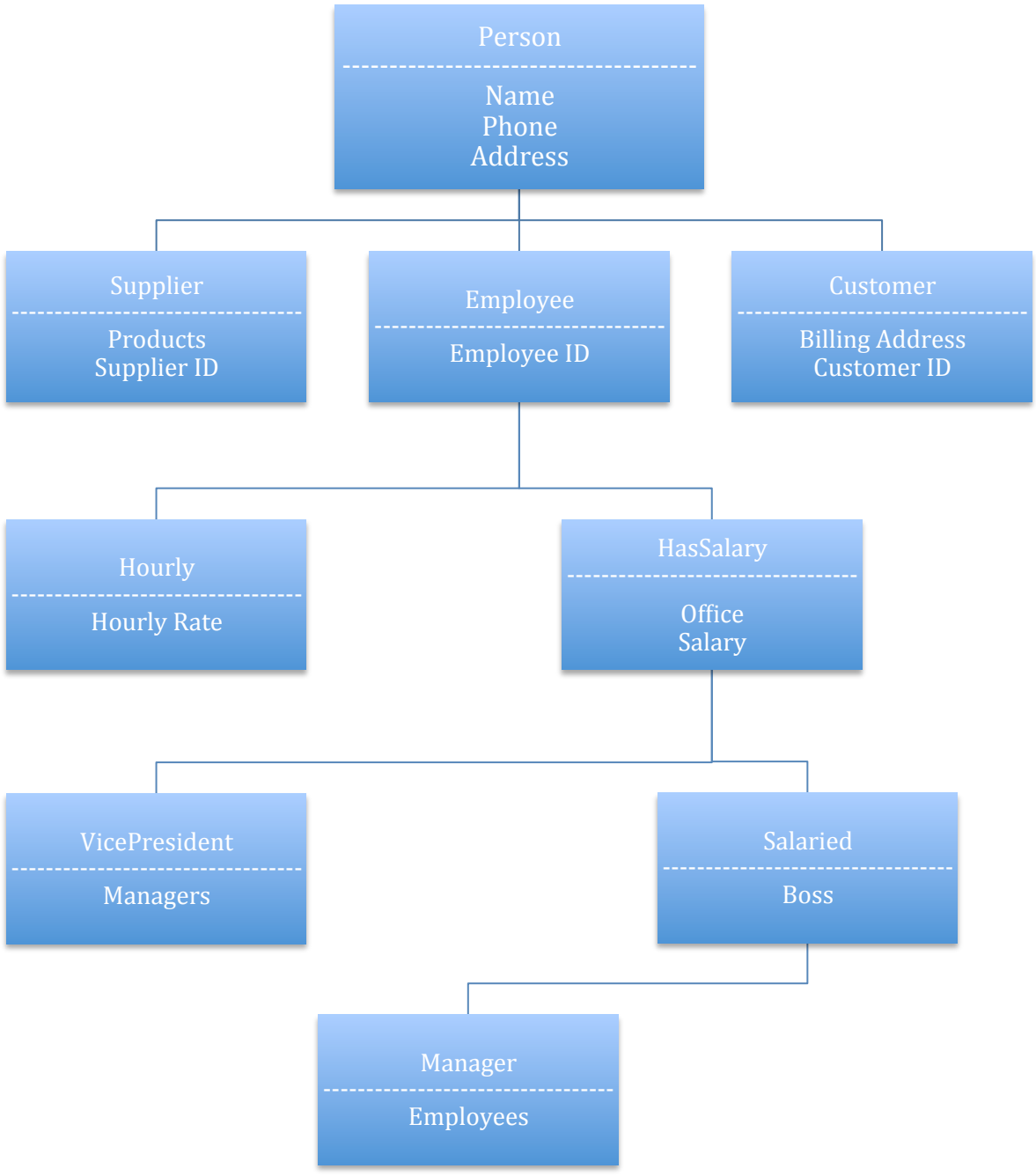
The foreground color and background color are properties needed for those classes to draw.
The upper-left corner, width, and height are needed for those classes to define the position.
The Text class needs font information and the string to draw.
The Star class needs to know how many points to give the star.
Some properties can be shared by some classes and not others. For example, the Rectangle, Ellipse, and Star can be filled, so they need a fill color.
Finally, we will need properties such as line thickness and dash style for the classes that draw line (Line, Rectangle, Ellipse , and Star.)

PROPERTY	USED BY
UpperLeft	All
BackGroundColor	All
ForeGroundColor	All
Width	All
Height	All
String	Text
Font	Text
NumPoints	Star
FillColor	Rectangle , Ellipse , Star
DashStyle	Rectangle , Ellipse , Star, Line
LineThickness	Star, Line, Rectangle , Ellipse

Problem 6.2, Stephens page 138



Problem 6.3, Stephens page 139



Problem 6.6, Stephens page 139

(1) We could give the Salaried class the properties Office, Salary, Boss, and Employees.
(2) The Boss property would not be filled in for top-level vice presidents who don't report to anyone.
(3) The Employees property would be empty for bottom-level employees who are not managers of anything.