Teddy Chu, Justin Kyle Torres, Alejandro Zapata

Homework #2

Problem 5.1: What's the difference between a component-based architecture and a service-oriented architecture?

Component-based architectures focuses the abstraction level of a component that provides functionality or a service to other components. Meanwhile, service-oriented architectures are not as connected, focusing on services that commonly communicate through a network rather than directly. Thus, component-based architectures are commonly more connected than service-oriented ones.

Problem 5.2: Suppose you're building a phone application that lets you play tic-tac-toe against a simple computer opponent. It will display high scores stored on the phone, not in an external database. Which architectures would be most appropriate and why?

Monolithic, rule-based, event-driven architecture is appropriate because it’s a relative simple application and having something like client/server would be overkill. We would also it have be rule-based to respond accordingly to the flow of tic tac toe. Event-driven to respond to user input. We don’t even need it to be data-centric, as there is no external database. We can simply serialize to an encoded file for the save format.

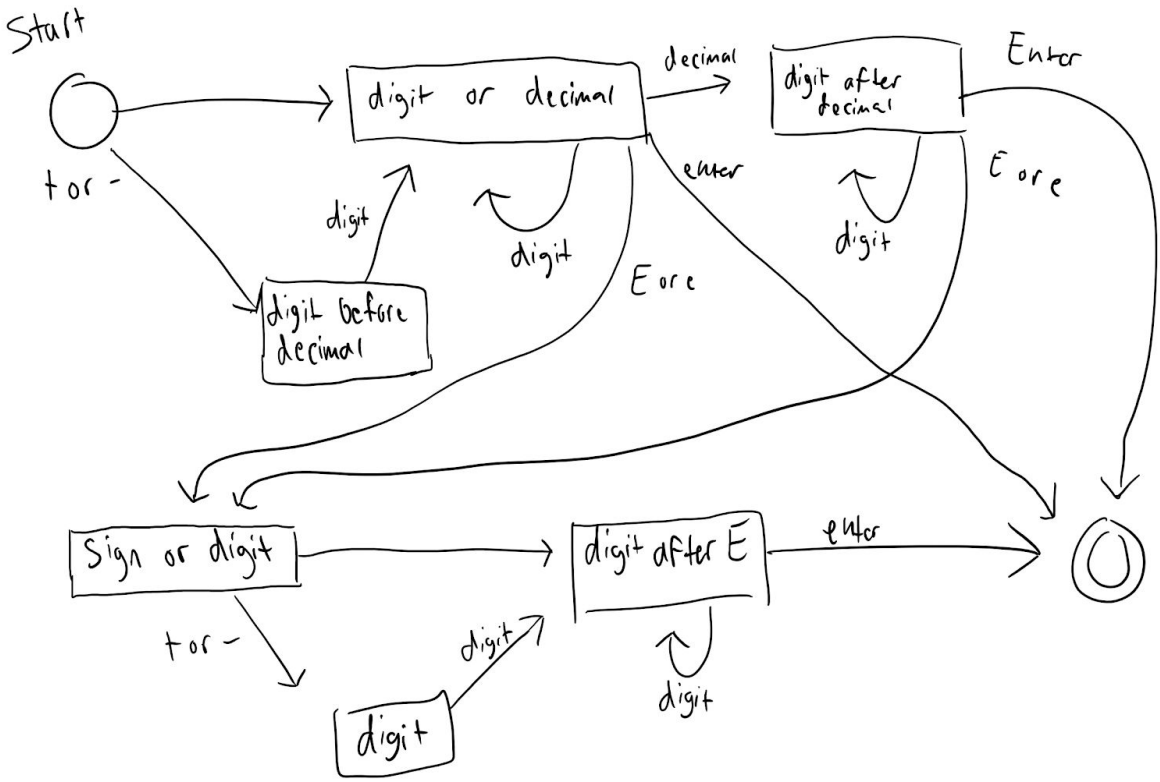
Problem 5.4: Repeat question 3 [after thinking about it; it repeats question 2 for a chess game] assuming the chess program lets two users play against each other over an Internet connection.

To account for the networking, we would have it be service-oriented. It can still be monolithic because of the simple nature of the app, rule-based to account for the game flow and win/lose states of chess.

Problem 5.6: What kind of database structure and maintenance should the `ClassyDraw` application use?

There's no need for a database, since it's save/load functions are simple. It could just serialize to a file for each drawing. Users can manage it like they would normal files on the system. To account for system crashes or failures, there could be a hashed temporary copy of the file that could get restored after startup.

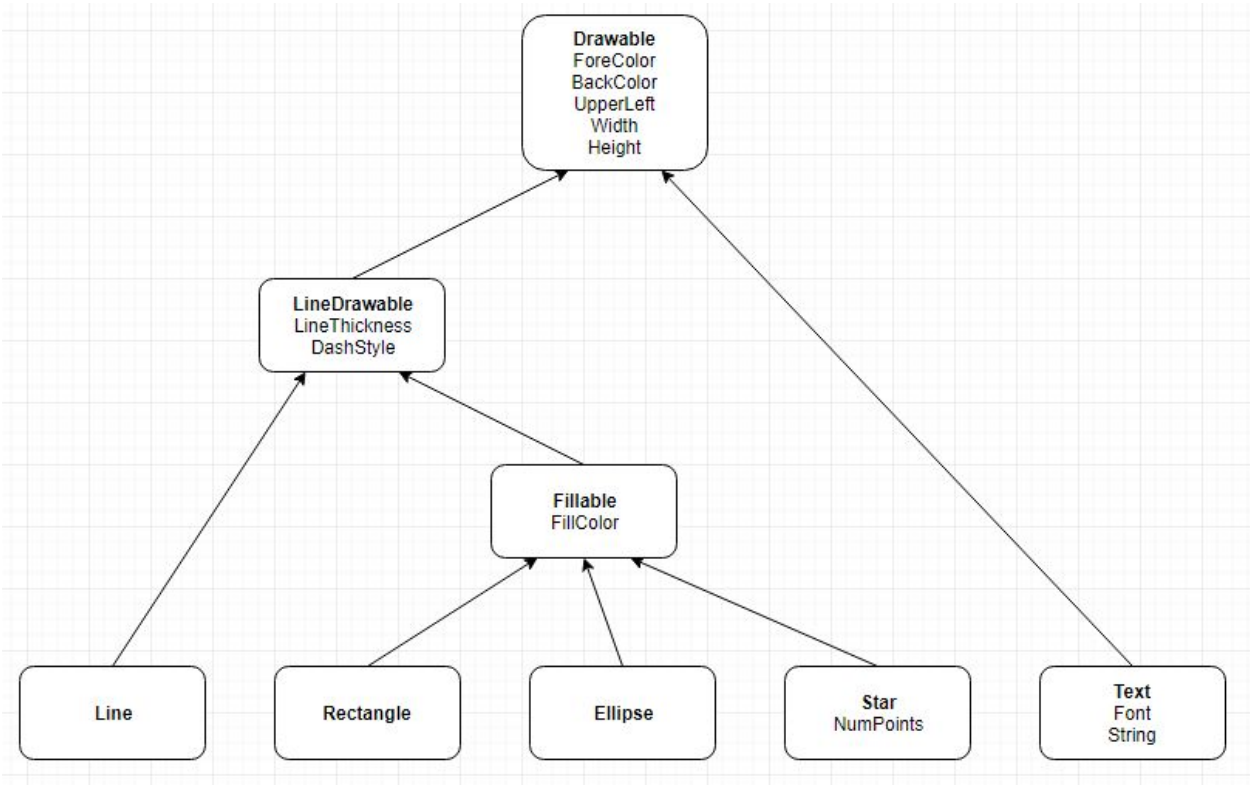
Problem 5.8: Draw a state machine diagram to let a program read floating point numbers in scientific notation as in +37 or -12.3e+17 (which means -12.3 x 10^{17}). Allow both E and e for the exponent symbol. [Jeez, is this like Dr. Dorin's DFAs, or *what*???]



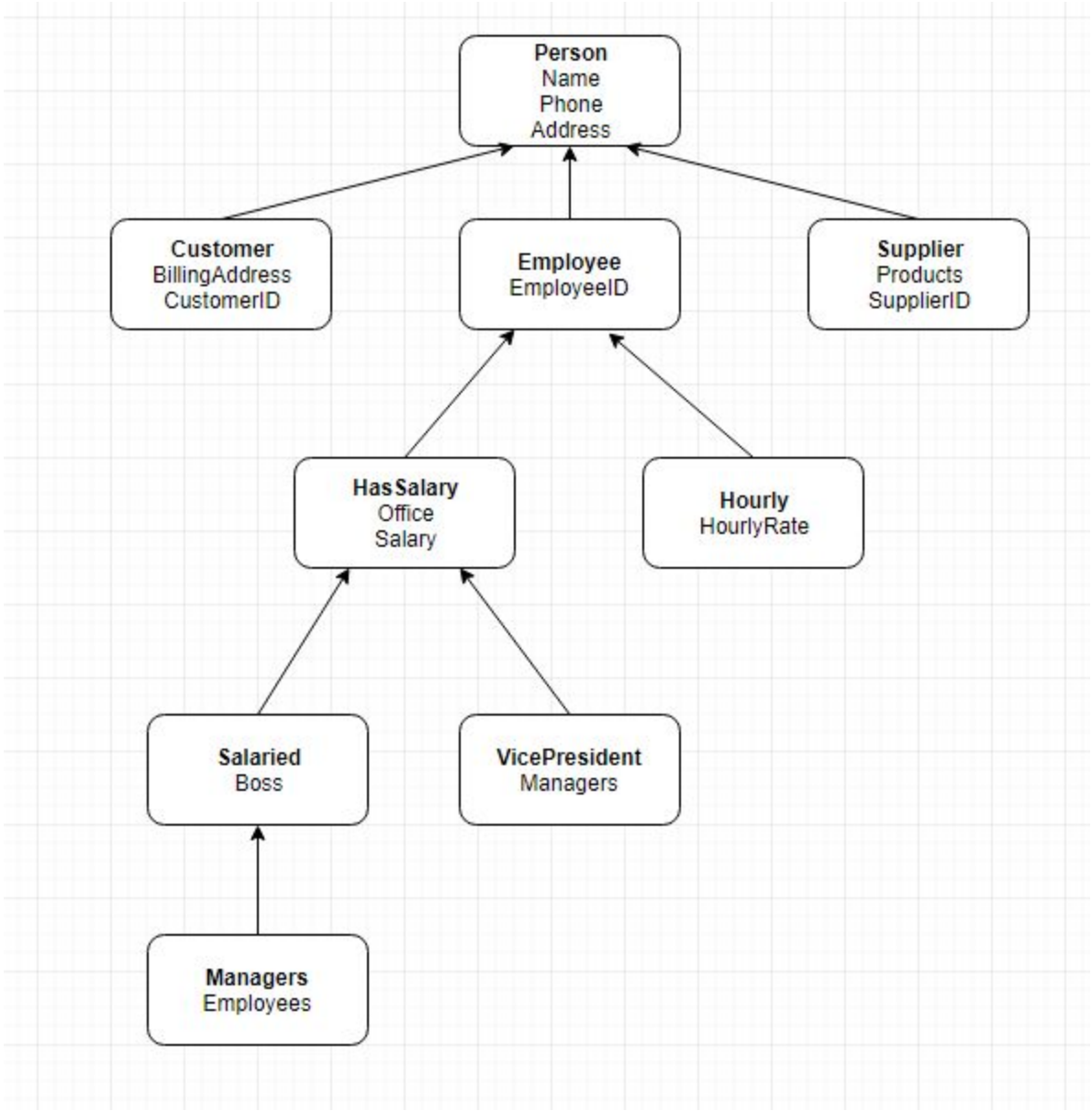
Problem 6.1, Stephens page 138: Consider the ClassyDraw classes Line, Rectangle, Ellipse, Star, and Text. What properties do these classes all share? What properties do they not share? Are there any properties shared by some classes and not others? Where should the shared and nonshared properties be implemented?

These classes share the necessary properties to represent a drawing. They would share a position for their anchor (x, y) or you can just default the anchor for all them to be top left, and have the position describe where the top left of the drawing is. You can also include colors such as the foreground and background colors. There are differences between shapes that might make it so that these shapes won't share the information needed to draw the shapes between the classes. The shared properties should be implemented in some parent drawing class, and the nonshared properties should be implemented in their respective class.

Problem 6.2:


Drawable
ForeColor
BackColor
UpperLeft
Width
Height
LineDrawable
LineThickness
DashStyle
Fillable
FillColor
Line
Rectangle
Ellipse
Star
NumPoints
Text
Font
String

Problem 6.3:


Person
Name
Phone
Address
Customer
BillingAddress
CustomerID
Employee
EmployeeID
Supplier
Products
SupplierID
HasSalary
Office
Salary
Hourly
HourlyRate
Salaried
Boss
VicePresident
Managers
Managers
Employees

Problem 6.6:

You can combine most of the classes down the deepest subtree into the “Salaried” class and leave the “Boss” blank if manager and “Employees” blank if bottom-level employees.