Jay Patel

Homework 2

Problem 5.1, Stephens page 116

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

A component based architecture regards pieces of the system as looselt coupled components that provide servies for each other while a service oriented architecture is implemented as services on seperate computers communicating across a network.

Problem 5.2, Stephens page 116

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?

It would be easiest with a monolithic rule based application

Problem 5.4, Stephens page 116

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.

It would make the chess game an monolithic rule based service oriented application

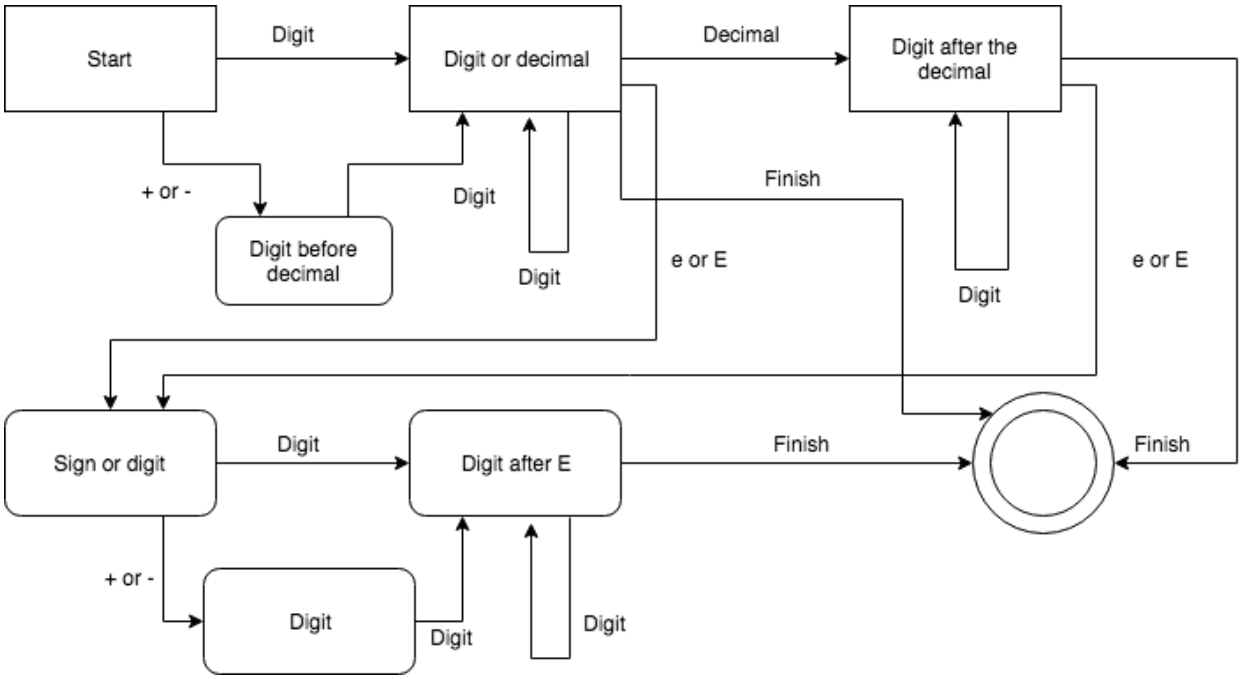
Problem 5.6, Stephens page 116

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

In terms of database structure none really and for maintence probably audit trails.

Problem 5.8, Stephens page 116

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?

They all share color, width, height.

Star has number of points

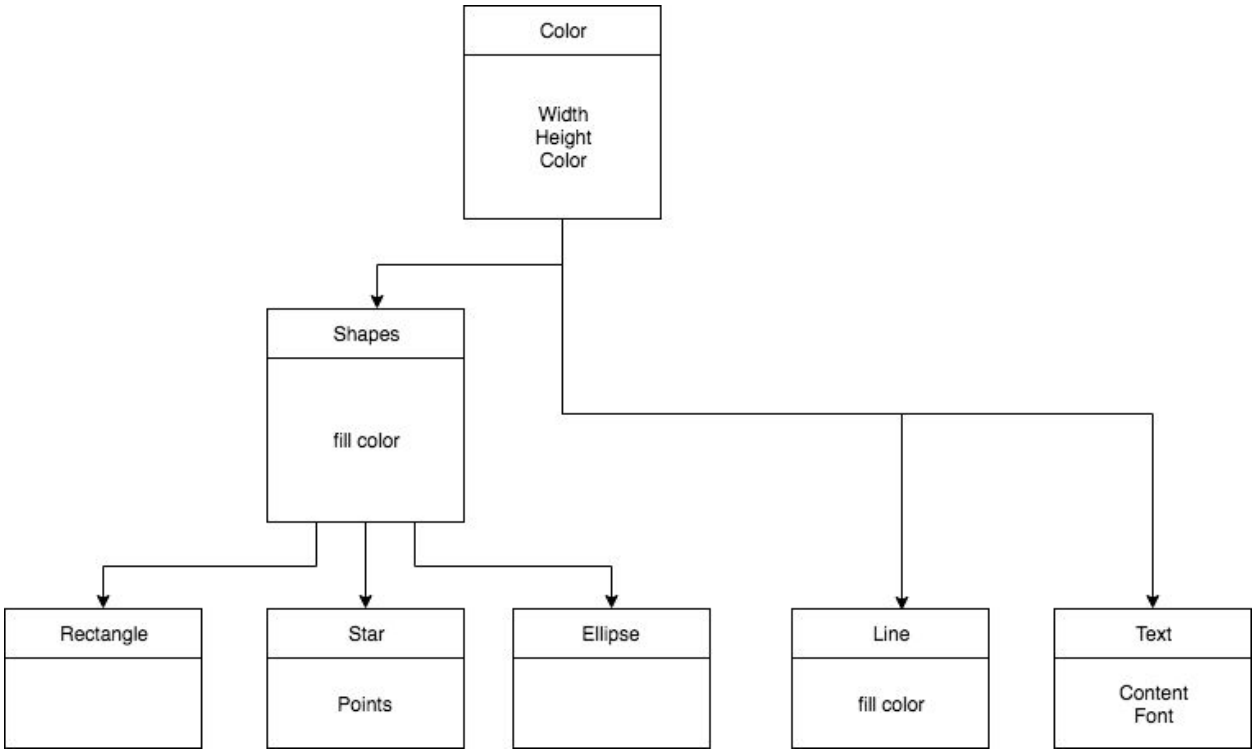
Text has font type, content

Rectangle, Ellipse and Star has fill color

All the properties they share should be in their parent classes and nonshared should be implemented in their respective classes.

Problem 6.2, Stephens page 138

Draw an inheritance diagram showing the properties you identified for Exercise 1. (Create parent classes as needed, and don't forget the `Drawable` class at the top.)



Problem 6.3, Stephens page 139

The following list gives the properties of several business-oriented classes.

- Customer — Name, Phone, Address, BillingAddress, CustomerID
- Hourly — Name, Phone, Address, EmployeeID, HourlyRate
- Manager — Name, Phone, Address, EmployeeID, Office, Salary, Boss, Employees
- Salaried — Name, Phone, Address, EmployeeID, Office, Salary, Boss
- Supplier — Name, Phone, Address, Products, SupplierID

- VicePresident — Name, Phone, Address, EmployeeID, Office, Salary, Managers

Assuming a `Supplier` is someone who supplies products for your business, draw an inheritance diagram showing the relationships among these classes. (Hint: Add extra classes if necessary.)

Problem 6.6, Stephens page 139

Suppose your company has many managerial types such as department manager, project manager, and division manager. You also have multiple levels of vice president, some of whom report to other manager types. How could you combine the `Salaried`, `Manager`, and `VicePresident` types you used in Exercise 3? Draw the new inheritance hierarchy.