Homework 2

- Eileen Choe
- CMSI 402

Problem 5.1, Stephens page 116

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

A component-based architecture considers the pieces of the system as loosely coupled components that provide services for each other.

A service-oriented architecture is similar except the pieces are implemented as services, often running on separate computers communicating across a network. The two are similar, but the pieces are more separated in a service-oriented architecture.

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?

The most appropriate architcture for this application will be a monolithic architecture that follows a data-centric approach. Since this is a simple application the more complicated architectures such as client-server, multitier, component-based, etc. are not necessary. Tic tac toe is a structured game, thus rule-based techniques are appropriate for modeling the possible moves in the game state. The user-interface will be event-driven, because the game state should update based on user events.

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.

The user interface will remain the same as in question 3, however the chess program in which two players play against each other over an Internet connection needs to exchange this information over the Internet, using web services. Additionally, the opposing player replaces the computer simulated opponent.

Problem 5.6, Stephens page 116

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

Instead of a database, the application can use a file system managed by the Operating system to store drawings as files. Files can be added, deleted, and modified.

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 1017). Allow both E and e for the exponent symbol. [Jeez, is this like Dr. Dorin's DFAs, or *what*???]

(IMAGE)

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 all represent things that are drawn using the application, and share properties such as color and its location on the drawing canvas, as stored in `(x,y)` coordinates.

The Text class will need additional properties such as font, font type (i.e. bold, italic, etc). The Rectangle, Ellipse, and Star classes may need additional properties such as border, border-width, and border-color.

Rectangle, Ellipse, and Star can be filled so they may share a fill color property. The classes that are line-based (text, line)may share properties such as the line thickness.

Shared properties should be implemented at a higher level of abstraction, vs. non-shared properties should be stored in the classes themselves.

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.)

(ANSWER BELOW)

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.)

(ANSWER BELOW)

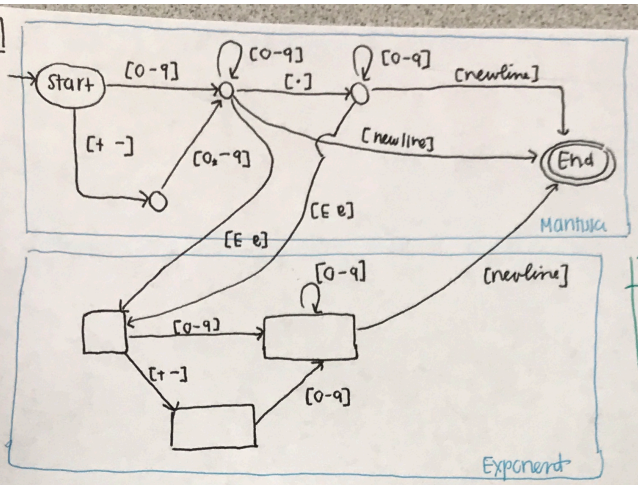
Problem 6.6, Stephens page 139

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

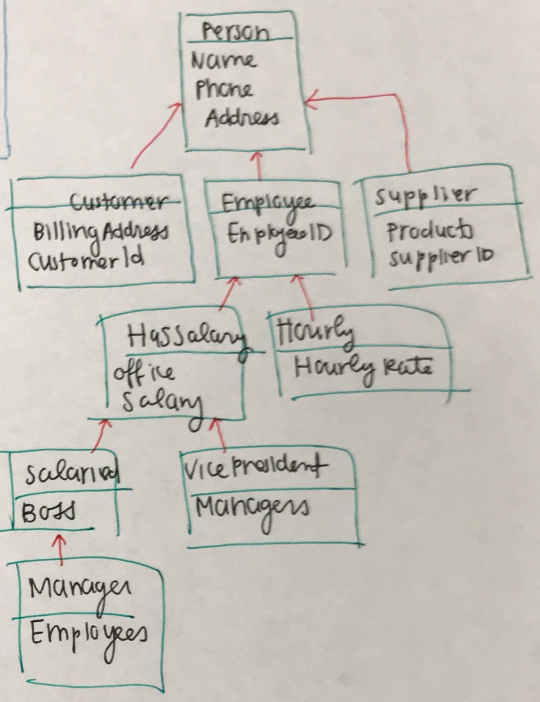
You can consider the Salaried, Manage, and VicePresident positions under the general "Employee" class. Then the `salaried` class can be modified to include properties such as Office, Salary, Boss, and Employees in order to capture the differences between Employees. For example The `boss` property will be `null` for VicePresidents who don't report to anyone.

(ANSWER BELOW)

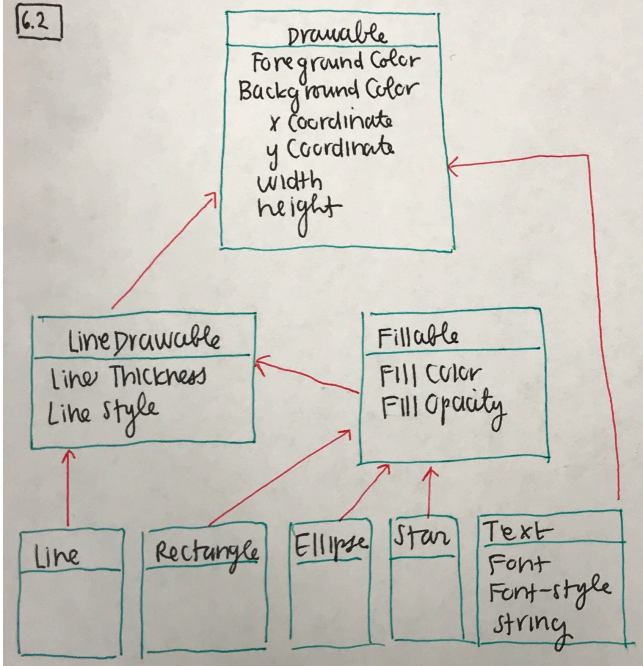
5.8


Start
[0-9]
[0-9]
[.]
[0-9]
[newline]
[+ -]
[0-9]
[newline]
End
[E e]
[E e]
Mantissa
[0-9]
[newline]
[0-9]
[+ -]
[0-9]
Exponent

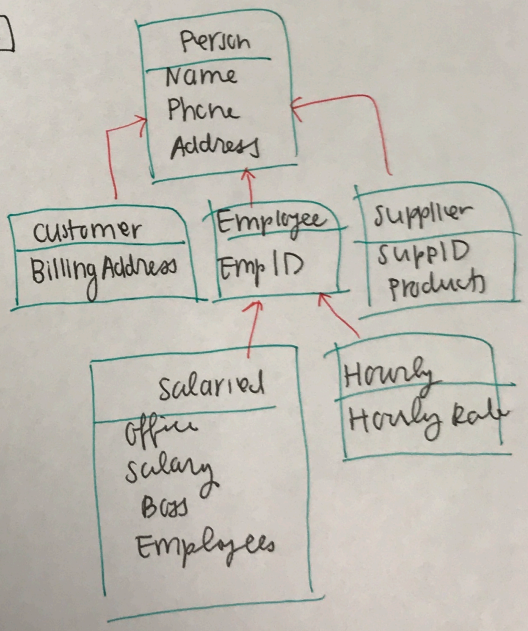
6.3


Person
Name
Phone
Address
Customer
Billing Address
Customer Id
Employee
Employee ID
Supplier
Products
supplier ID
HasSalary
office
Salary
Hourly
Hourly Rate
Salaried
Boss
Vice President
Managers
Manager
Employees

6.2


Drawable
Foreground Color
Background Color
x Coordinate
y Coordinate
Width
height
LineDrawable
Line Thickness
Line Style
Fillable
Fill Color
Fill Opacity
Line
Rectangle
Ellipse
Star
Text
Font
Font-style
string

6.6


Person
Name
Phone
Address
Customer
Billing Address
Employee
Empl ID
Supplier
SupplD
Product
Salaried
Office
Salary
Boss
Employees
Hourly
Hourly Rate