Assignment 3

(See Blackboard for Due Date)

Overview

After playing around with the game ***Zork*** a little bit more (or really any of the classic "interactive fiction games"), draw a **"High level" UML *Class (aka, Static Model) Diagram*** that models the game's key elements and their relationships. (Rooms, Items, creatures, objects, etc.).

For your convinience (the URL's of online versions of Zork and HHGG are below.)

- Zork
  - https://www.pcjs.org/software/pcx86/game/infocom/zork1/
    - This runs in an emulator for the IBM PCjr. So, it's a 1980s game… running on emulated 1980s hardware… in JavaScript… in a browser!!
  - https://www.ifiction.org/games/play.php?cat=2&game=3&mode=html
    - This version has a strange web interface where you must type into a HTML field below the text screen.
- Hitchhikers Guide to the Galaxy (HHGG) -- This is one of my Favorites.
  - https://www.bbc.co.uk/programmes/articles/1g84m0sXpnNCv84GpN2PLZG/the-game-30th-anniversary-edition
  - Note that this one is rather difficult if you are not experienced with these types of games! While rated "Standard Level", I have seen ***modern*** reviews from people going back to play this claiming that it is "*impossible"* to play without cheating and looking up solutions (hint, it's definitely not). *However, there are puzzles where reading the books will help, but there are others where it makes the puzzle more difficult due to preconceived notions leading you astray. Given that Douglas Adams was the story writer and Steve Meretzky was the main programmer, I believe this was totally by design! You must think creatively, and the game often tries to mislead you!*

Instructions

1. Play around with Zork or HHGG, until you get a general sense of the objects in the environment. Think about how various game elements could be represented as ***classes*** and what their responsibilities and interactions might be.

2. After familiarizing yourself with the game, using a drawing tool like one of those listed on the last page of this document, draw a UML diagram for the game's main elements, including what pieces of data (fields) might be relevant and what methods might be needed.

3. Again, this is a *general diagram*, ***so you do not need to include every specific item*** like "mailbox" or "trophy case", but instead, each of those items might be an instance "fixed container" object, for example. Other types of objects may be items you can pick up (relocatable objects) that can store other relocatable items, etc.

4. While this is NOT about diagraming every possible item/location/etc., your diagram should include at least one representative "concrete class" for any abstract classes or interfaces in the game.

   (For example, if in your design, you would have many "concrete classes" extending an abstract "Item" class, then you should show at least one of those concrete classes, but you **do not** need to include ***every possible*** class that would extend "Item".)

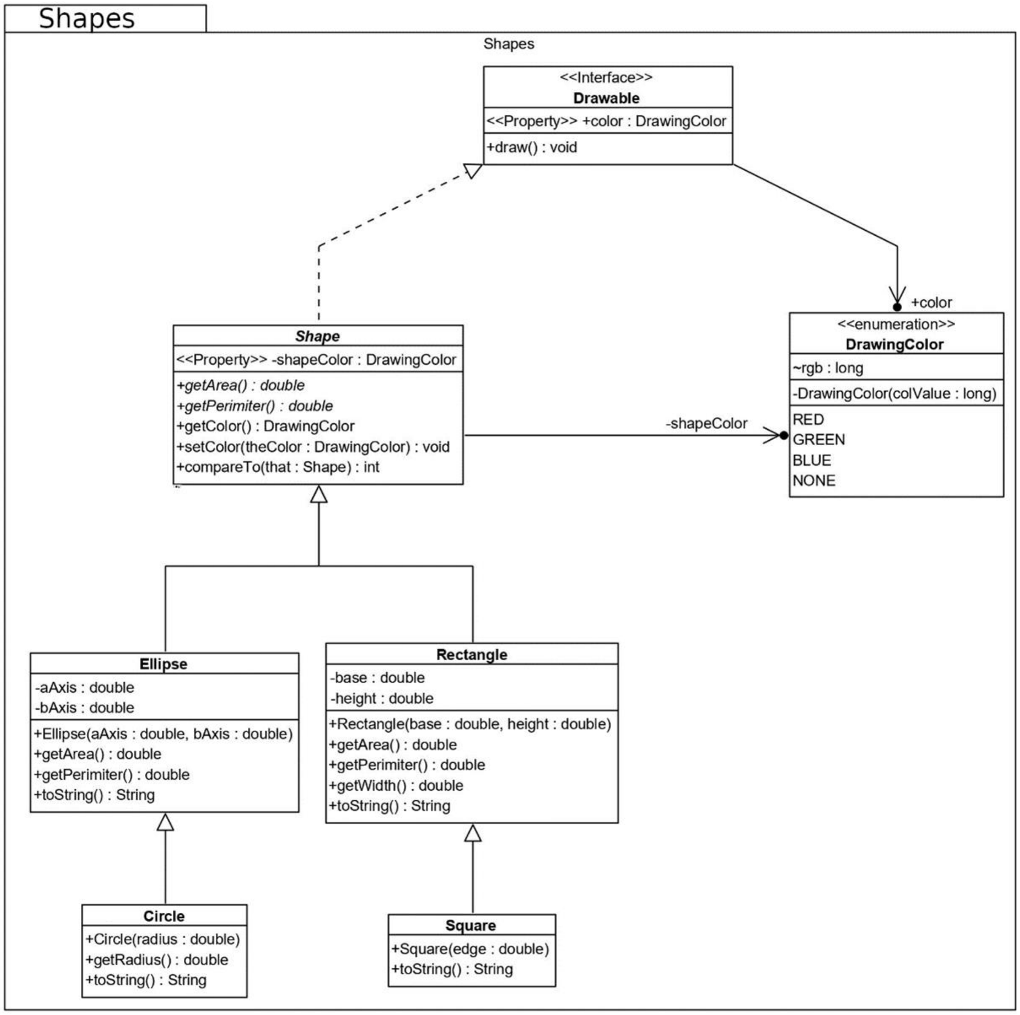
Submission

You only need to submit one diagram as either a PNG or JPG with sufficient resolution to be clearly legible. Your name should appear (somewhere) on the image as well.

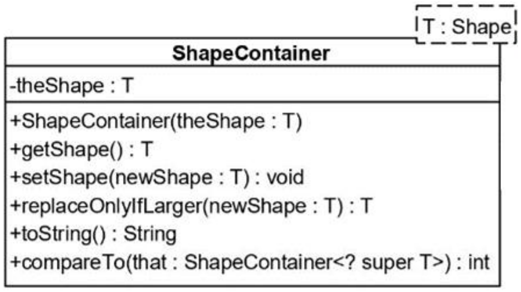
An example UML diagram is provided below, and a list of diagramming tools is located on the last page of this document.

An Example UML Diagram

(but NOT an example of a “Good Design”, just the general diagram concepts)



An Example of Templatized Types “Generics”


T : Shape
ShapeContainer
-theShape : T
+ShapeContainer(theShape : T)
+getShape() : T
+setShape(newShape : T) : void
+replaceOnlyIfLarger(newShape : T) : T
+toString() : String
+compareTo(that : ShapeContainer<? super T>) : int

References for UML Class Diagrams.

- A Microsoft reference

- https://holub.com/uml/ (See Section: **"Static-Model (Class) Diagram"**)

- https://www.uml-diagrams.org/class-diagrams-overview.html (See: "**Diagram of Implementation Classes"**)

- https://loufranco.com/blog/assets/cheatsheet.pdf

- A few other are listed in Blackboard as well.

Diagramming Tools.

While there are MANY (and understatement) UML tools, many free ones are terrible and most professional systems can cost many thousands of dollars!

However, here is a **SMALL** selection of useful tools, **MY RECOMMENDATION IS "**PlantUML**"**, while "Violet" is super simple, it is a bit lacking in functionality; "Dia" is a bit "clunky" but has good UML semantic awareness.

- PlantUML (Free)
  - This one has recently become my favorite!
  - PlantUML has no GUI but rather uses simple text descriptions to draw the diagrams automatically, handling all the layout, etc. While I know that NON-GUI interfaces can be intimidating for some, PlantUML can be MUCH easier to use than many other "Drawing Tools.
  - Plugins exist for many IDE's

- Dia: General Purpose Diagramming Tool (Free)

- Lots of other choices:

  - Violet: Super Simple Diagramming Tool
    - (Free)
    - https://horstmann.com/violet/
    - https://horstmann.com/violet/violet-0.16c.jar

  - Lucid Charts
    - (Free/Commercial)
    - https://www.lucidchart.com/pages/

  - yED
    - (Free/Commercial)
    - https://www.yworks.com/
    - https://www.yworks.com/yed-live/

  - Visual-Paradigm
    - (Commercial with trial Version. Only included here for those that want to explore it)
    - https://www.visual-paradigm.com

  - Of course, there are many others: Draw.io, etc.