Project 1 - Polymorphism/Inheritance OOP

Ryan A Burkhardt

UMGC

CMSC 335

Professor Ghosh

28May2024

Polymorphism/Inheritance OOP – Approach

For my approach to this project, I started by reading all required documentations in the Project 1 pdf. I then went into [Calculator.net](https://www.calculator.net/area-calculator.html) to pull all required calculations for area and volume. After having all the calculations present, I went top down and started creating my UML diagram. After creating the UML diagram I went into structuring the classes and finding out the hierarchy from shape to dimensional shape to the actual objects, square, triangle etc. After creating one class I realized that mostly all the same dimensional classes were reusable and proceeded to copy and paste the classes while changing the incoming variables and appropriate calculations. After the classes were created, I moved to the testing and evaluation of the code. This is where I hit some errors on how I was using the super classes in creating my functions such as setArea and setVolume. After realizing my error in the super class I just put my formula in and the setArea and setVolume that calculated each at the time they were passed in. This works and is efficient to give proper feedback to the user but could be improved.

**Polymorphism/Inheritance OOP – Assumptions**

The assumptions I have made in this assignment is that the user has basic knowledge of programming, will input accurate values such as non-negative integers, triangle class is a right triangle. With respect to the user having programming knowledge, I presume that they can run a javac command and import all appropriate classes as I am just providing the .java files. It is also assumed with their knowledge that they can run terminal-based applications and input appropriate data as instructed on the terminal. With respect to input data, it is assumed that they know the background of the calculation they are trying to perform. For instance, it is assumed that they know their measurements for each object they will be attempting to pull an area or volume for. If the user has a square, it is assumed that they know they only need one side to calculate the area and not a length and a width. This follows suit for each class created to calculate the area and volume. With the triangle class it is assumed that the user id inputting a right triangle for the area calculation which simply calculates ½(base\*height).

Polymorphism/Inheritance OOP – Not Implemented

In the non-implemented section, I would have liked to have one area where all calculations were present. This could have been a separate class where I defined a simple calculation class where I could have implemented all calculations to use for are and volume. I also would like to add more error handling to the program itself. For example, negative integers and zeroes can currently be inputted into the calculations giving you errors or invalid calculations. This is something that in a full-scale development would have to be addressed and implemented into the code. Another non-implemented method would be for a more user-friendly terminal app. This could have been done by creating a list and having them select from that list like they do but then offering more feedback such as I did in the Torus class where if they do enter incorrect baseline information it tells them what they did wrong and prompts them to correct it. Lastly, I think I would have liked to input the code into the actual class so that the user only sees the class name in the main code. As I did not know exactly what to do for the assignment I chose to do all variable calculations in the main method. This instead would have been a much simpler code such as case 1: circle() and the circle class would have had the code inside for the menu such as enter base enter height and the calculations were done there. This is something I do in python but have not yet started doing in java.

**Polymorphism/Inheritance OOP – User Guide**

The user guide is quite simple for the terminal-based application taking in the assumptions I have listed above. From an IDE you would simply run the main class, and everything would work as is. After you would follow the prompts on the terminal and input a numerical selection from the displayed menu. Next you should again follow the instructions and input the data it is asking for on the terminal until you have concluded the calculations you would like to have made. Lastly to close the application you would select 10 from the main menu or hit n after your calculations are made.

Polymorphism/Inheritance OOP – Lessons Learned

Lessons I learned in this project were that I simply could not create one class and then just implement other classes in the code. For instance, my original plan was to have a shape class then inside that java file have a two-dimensional shape and three-dimensional shape and then also inside that file create all the required objects to calculate the area and volume. I learned that putting them inside the separate java files is when you are creating the objects and using supers and overrides in the separate java classes is when you are creating that implemented override.

**Polymorphism/Inheritance OOP – Possible improvements**

For possible improvements I would start with implementing all error handling inside the program. This is essential and should be improved on if this were to be implemented in the real world. My second improvement would be to use a GUI based application such as JavaFX with text boxes and a calculate button. I feel like this would give the user a much more interactive experience with the application and make it much easier to use.

Polymorphism/Inheritance OOP – UML

A screenshot of a computer

Description automatically generated

Polymorphism/Inheritance OOP – Source Code

References

Last Name, A. B. (Year). Article Title. *Journal Title*, Pages #-#. URL.

Last Name, C. D. (Year). *Book Title (Edition).* Publisher Name. URL.

Last Name, D. E., Last Name, F. G., Last Name, H. I. (Year). *Report Title* (report number). Publisher. URL.

Last Name, J. K. (Year, Month Day). *Article Title/Headline*. Periodical. URL.

Organization Name. (Year, Month Day). *Webpage Title*. URL.