Project 1

Shapes

Samuel Scalf

CMSC 335 · Dr. Osama Morad

University of Maryland Global Campus

Table of Contents

User’s Guide....3
System Requirements....3
Compiling the Program....3
Windows 10....3
Linux....3
Using the Program....3
Documentation....4
Testing....4
Test Case 1: Invalid Menu Selection....4
Test Case 2: Exit Program Option....5
Test Case 3: Shape – Invalid Input....5
Test Case 4: 2D Shape – Valid Input....6
Test Case 5: 3D Shape – Valid Input....6
Lessons Learned....7
Appendix A: UML Class Diagram....8

User’s Guide

System Requirements

Space: 19.0 kB

Software: Java Development Kit (JDK) version 8 update 261

Compiling the Program

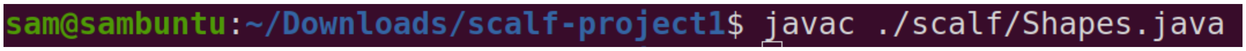
Before compiling the program, the downloaded zip file needs to be uncompressed. The means of uncompressing, or extracting, a zip file is not covered here. Open the directory containing the extracted files, if not already there. Open the “src” folder. Compilation of the program requires the JDK version 8. The procedures required for this are not covered in this guide, but can be found on Oracle’s website.

Windows 10

Click File > Open Command Prompt > Open Command Prompt. Type “javac .\scalf\Shapes.java” without the quotes and press the “Enter” key.

Linux

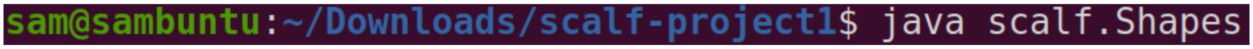
Open a terminal in the location of the files. Type “javac ./scalf/Shapes.java” without the quotes and press the “Enter” key.



```
sam@sambuntu:~/Downloads/scalf-project1$ javac ./scalf/Shapes.java
```

Using the Program

To run the program, from the command prompt or terminal (see Compilation steps), type “java scalf.Shapes” without the quotes and press the “Enter” key. The program is command linedriven, so all user input will be in the command prompt or terminal. To respond to questions, type your answer and press the “Enter” key.



```
sam@sambuntu:~/Downloads/scalf-project1$ java scalf.Shapes
```

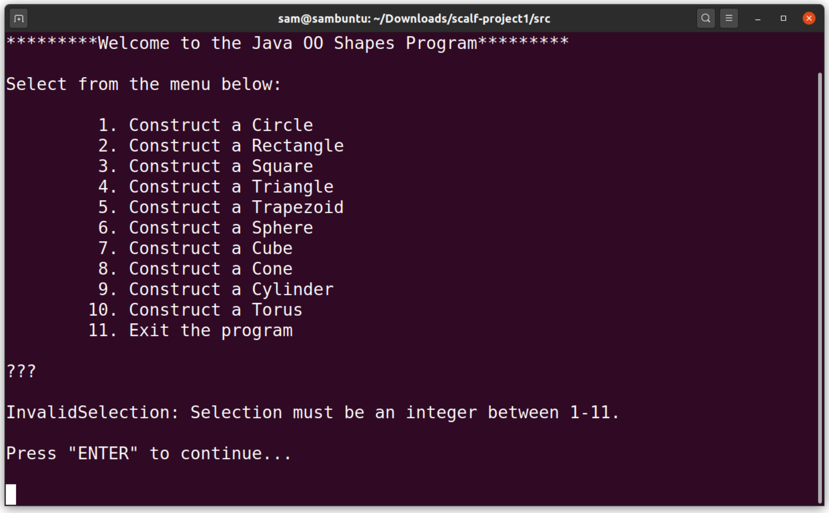
Documentation

Additional information about the packages and classes can be found by opening “index.html” from the downloaded “doc” folder in a web browser. The website was created using javadoc, so it has the same format as the Oracle Java 8 API.

Testing

Test Case 1: Invalid Menu Selection

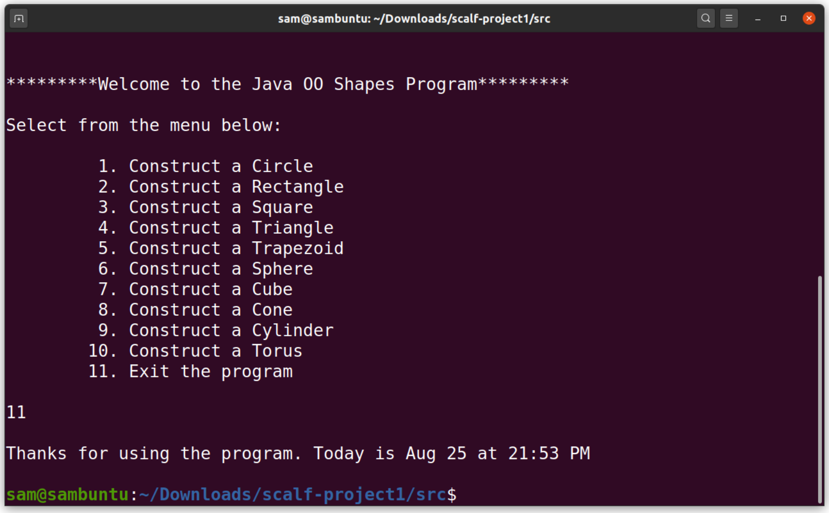
Any value that is not an integer value listed on the menu should return a notice that the input must be between 1 and 11, or whatever is the last number on the menu.



```
*********Welcome to the Java OO Shapes Program*********

Select from the menu below:

         1. Construct a Circle
         2. Construct a Rectangle
         3. Construct a Square
         4. Construct a Triangle
         5. Construct a Trapezoid
         6. Construct a Sphere
         7. Construct a Cube
         8. Construct a Cone
         9. Construct a Cylinder
        10. Construct a Torus
        11. Exit the program

???

InvalidSelection: Selection must be an integer between 1-11.

Press "ENTER" to continue...

```

Test Case 2: Exit Program Option

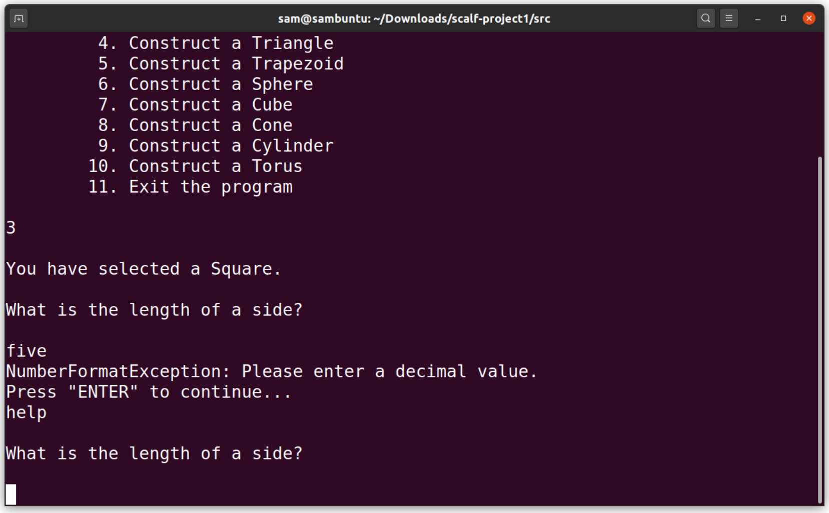
When the last menu option is entered, the program should display a message with a "thank you" for the user and the current date and time.



```
*********Welcome to the Java OO Shapes Program*********

Select from the menu below:

         1. Construct a Circle
         2. Construct a Rectangle
         3. Construct a Square
         4. Construct a Triangle
         5. Construct a Trapezoid
         6. Construct a Sphere
         7. Construct a Cube
         8. Construct a Cone
         9. Construct a Cylinder
        10. Construct a Torus
        11. Exit the program

11

Thanks for using the program. Today is Aug 25 at 21:53 PM

sam@sambuntu:~/Downloads/scalf-project1/src$
```

Test Case 3: Shape – Invalid Input

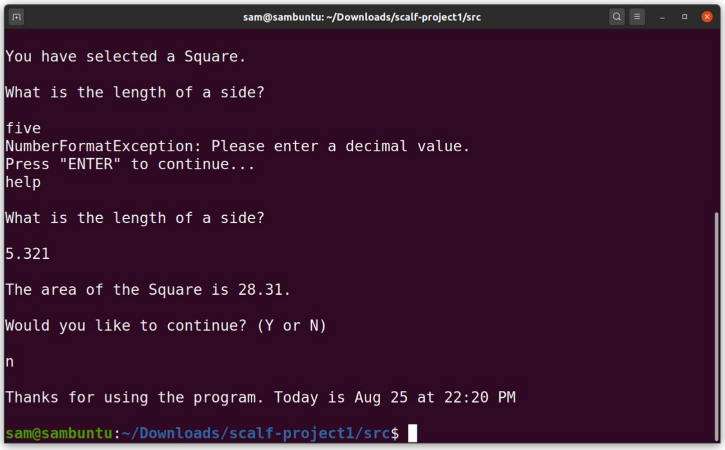
Regardless of the Shape selected, the same method for input validation is called so the same message notifying the user that only decimal values are accepted should be displayed. The program will also prompt the user to press "Enter" to continue. If anything is input at this point, it is ignored and effectively discarded. The program should then repeat the prompt for which the user had provided invalid input.



```
         4. Construct a Triangle
         5. Construct a Trapezoid
         6. Construct a Sphere
         7. Construct a Cube
         8. Construct a Cone
         9. Construct a Cylinder
        10. Construct a Torus
        11. Exit the program

3

You have selected a Square.

What is the length of a side?

five
NumberFormatException: Please enter a decimal value.
Press "ENTER" to continue...
help

What is the length of a side?

```

Test Case 4: 2D Shape – Valid Input

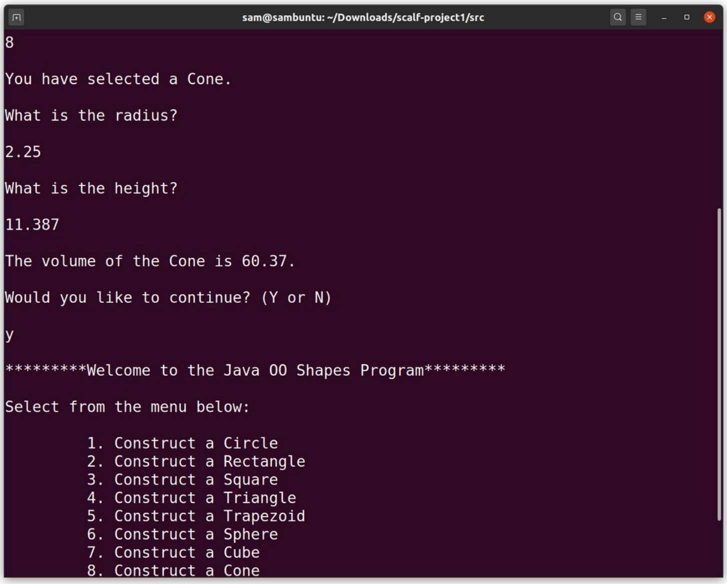
Regardless if the user had originally provided invalid input (see Test Case 3) or not, when the user provides valid input, the program should calculate the area of the two-dimensional shape rounded to two decimal places. The program will then ask the user whether to continue: a “No” or “N” is considered to be a wish to exit, while a “Yes” or “Y” should display the menu again. The negative option was selected here.



```
You have selected a Square.

What is the length of a side?

five
NumberFormatException: Please enter a decimal value.
Press "ENTER" to continue...
help

What is the length of a side?

5.321

The area of the Square is 28.31.

Would you like to continue? (Y or N)

n

Thanks for using the program. Today is Aug 25 at 22:20 PM

sam@sambuntu:~/Downloads/scalf-project1/src$ 
```

Test Case 5: 3D Shape – Valid Input

Similar to Test Case 4, it does not matter if the user provided invalid input as they will be continually prompted until valid input is received. Once valid input is received, the program will display the volume of the selected three-dimensional rounded to two decimal places.



```
8

You have selected a Cone.

What is the radius?

2.25

What is the height?

11.387

The volume of the Cone is 60.37.

Would you like to continue? (Y or N)

y
*********Welcome to the Java OO Shapes Program*********

Select from the menu below:

         1. Construct a Circle
         2. Construct a Rectangle
         3. Construct a Square
         4. Construct a Triangle
         5. Construct a Trapezoid
         6. Construct a Sphere
         7. Construct a Cube
         8. Construct a Cone
```

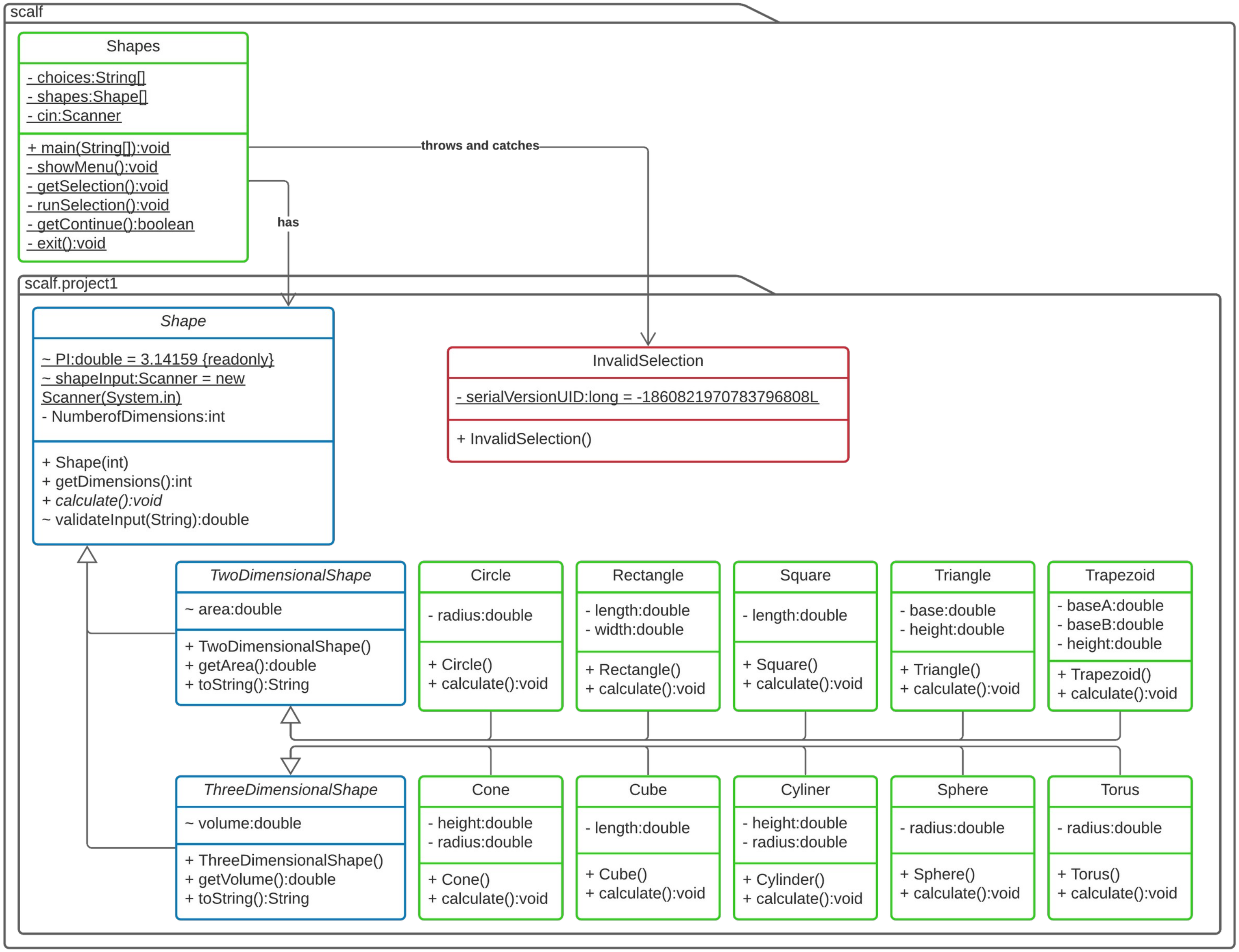
Lessons Learned

When I first started creating my program, I was using OpenJDK 8. It took me a while to find this out, but the printf method in the PrintStream class, which is called with System.out.printf, has a different parameter set than Oracle JDK. I was getting compilation errors and couldn't figure out why I was unable to pass multiple arguments for the string. For example "System.out.printf("Today is %s %d.", month, day);" would not compile with OpenJDK 8. I found the reason was OpenJDK 8's method is printf(String format, Object[] args), whereas Oracle JDK's method is printf(String format, Object … args). The difference may be subtle but has very great effects. For one, in an array of objects (Object[]) every object must be the same type. This is not need to be the case with "Object …," which will take any number of different objects. Another effect is that with Object[], the passed parameter must be a single array argument, whereas Object… accepts comma-separated arguments.

I originally learned Java programming with Oracle JDK, but when I recently switched to Linux because of some difficult driver issues, I started to use OpenJDK. I was under the impression that the basic functions of both would be the same since Oracle JDK is based on OpenJDK. This was obviously not the case and it was a lesson learned the hard way.

Another lesson I learned was that sometimes errors thrown in a never-ending loop, like "while (true)," that involve getting user input from the console can cause the loop to never pause for input. By capturing the using the Scanner objects "nextLine()" method within the catch exception block, I was able to stop the horrible, infinite loop of red text. I still do not fully understand the reasoning behind this. I had a line that got the user input inside the try block, but the program never pause to wait for the input, using the previously input information, instead.

Appendix A: UML Class Diagram


scalf
Shapes
- choices:String[]
- shapes:Shape[]
- cin:Scanner
+ main(String[]):void
- showMenu():void
- getSelection():void
- runSelection():void
- getContinue():boolean
- exit():void
throws and catches
has
scalf.project1
Shape
~ PI:double = 3.14159 {readonly}
~ shapeInput:Scanner = new Scanner(System.in)
- NumberofDimensions:int
+ Shape(int)
+ getDimensions():int
+ calculate():void
~ validateInput(String):double
InvalidSelection
- serialVersionUID:long = -1860821970783796808L
+ InvalidSelection()
TwoDimensionalShape
~ area:double
+ TwoDimensionalShape()
+ getArea():double
+ toString():String
Circle
- radius:double
+ Circle()
+ calculate():void
Rectangle
- length:double
- width:double
+ Rectangle()
+ calculate():void
Square
- length:double
+ Square()
+ calculate():void
Triangle
- base:double
- height:double
+ Triangle()
+ calculate():void
Trapezoid
- baseA:double
- baseB:double
- height:double
+ Trapezoid()
+ calculate():void
ThreeDimensionalShape
~ volume:double
+ ThreeDimensionalShape()
+ getVolume():double
+ toString():String
Cone
- height:double
- radius:double
+ Cone()
+ calculate():void
Cube
- length:double
+ Cube()
+ calculate():void
Cyliner
- height:double
- radius:double
+ Cylinder()
+ calculate():void
Sphere
- radius:double
+ Sphere()
+ calculate():void
Torus
- radius:double
+ Torus()
+ calculate():void