Assignment Volumes CSIS-1400
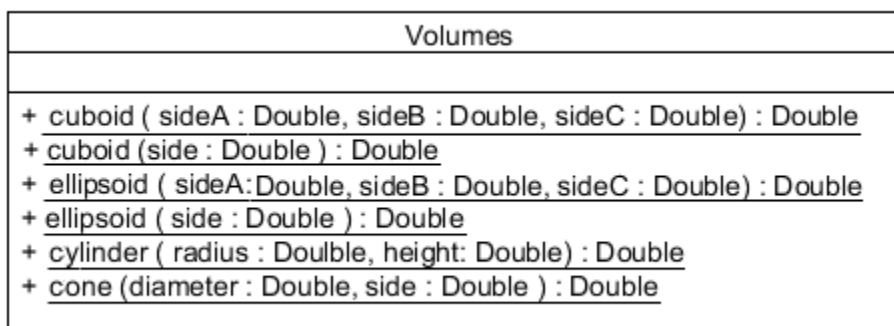
Learning Objectives:

- Practice method overloading
- Declare methods with multiple parameters
- Declare static methods and call them in the same class as well as from another class
- Call members of the Java API based on the information provided in the Java documentation
- Formulate algorithms

Description:

Create a project called **AVolumes** with two java files called **Volumes.java** and **VolumesMenu.java**

Volumes.java:

Implement the class as described in the UML class diagram below. **Do not add, remove or modify members** of the class

Volumes
+ cuboid (sideA : Double, sideB : Double, sideC : Double) : Double
+ cuboid (side : Double) : Double
+ ellipsoid (sideA:Double, sideB : Double, sideC : Double) : Double
+ ellipsoid (side : Double) : Double
+ cylinder (radius : Doulble, height: Double) : Double
+ cone (diameter : Double, side : Double) : Double

All members in the Volumes class are underlined. This indicates that all of them are static. This works well because none of the methods relies on data from a specific instance.

Notice that the type Double is capitalized. You see that in UML diagrams, which are language-independent diagrams.

Use the primitive type `double` (Java) to implement Double from the UML class diagram.

All methods return a value. **No print statements should be used in class Volumes.**

Hint: all methods need to be declared with the keyword static and

     all methods need to be called on the type (ClassName + dot operator + methodName)
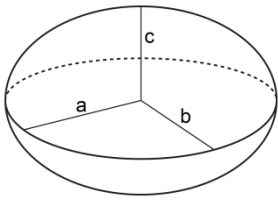
Ad cuboid:
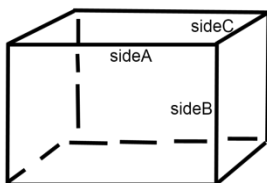
cuboid is an overloaded method.
First implement the 3-parameter cuboid method
Then implement the one-parameter method by calling the 3-parameter method
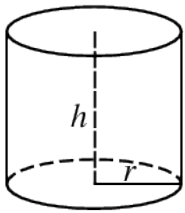
Ad ellipsoid:

$$\text{volume } = \frac{4}{3}\pi abc.$$

ellipsoid is an overloaded method.
First implement the 3-parameter ellipsoid method
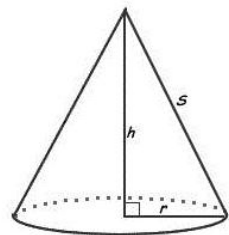Then implement the one-parameter method by calling the
3-parameter method

Ad cylinder:

$$\text{volume } = \pi r^2 h$$

Ad cone:

Caveat: The method cone receives the diameter and the side as parameters, however, the volume formula provided uses the radius and the height.

$$\text{volume } = \frac{1}{3}\pi r^2 h$$

Here are some members of the Math class that can be helpful:

static double	**PI** The double value that is closer than any other to pi, the ratio of the circumference of a circle to its diameter.
static double	**pow**(double a, double b) Returns the value of the first argument raised to the power of the second argument.
static double	**sqrt**(double a) Returns the correctly rounded positive square root of a `double` value.
static double	**hypot**(double x, double y) Returns sqrt($x^2 + y^2$) without intermediate overflow or underflow.

More info about the Math class: http://download.oracle.com/javase/7/docs/api/java/lang/Math.html

Output:

```
1 .. cuboid(2, 4, 6)      2 .. cuboid(4)
3 .. ellipsoid(2, 4, 6)   4 .. ellipsoid(4)
5 .. cylinder(3, 4)       6 .. cone(6, 5)
0 .. exit
Your choice: 1
cuboid(2, 4, 6): volume = 48.0

1 .. cuboid(2, 4, 6)      2 .. cuboid(4)
3 .. ellipsoid(2, 4, 6)   4 .. ellipsoid(4)
5 .. cylinder(3, 4)       6 .. cone(6, 5)
0 .. exit
Your choice: 2
cuboid(4): volume = 64.0

1 .. cuboid(2, 4, 6)      2 .. cuboid(4)
3 .. ellipsoid(2, 4, 6)   4 .. ellipsoid(4)
5 .. cylinder(3, 4)       6 .. cone(6, 5)
0 .. exit
Your choice: 3
ellipsoid(2, 4, 6): volume = 201.1

1 .. cuboid(2, 4, 6)      2 .. cuboid(4)
3 .. ellipsoid(2, 4, 6)   4 .. ellipsoid(4)
5 .. cylinder(3, 4)       6 .. cone(6, 5)
0 .. exit
Your choice: 4
ellipsoid (4): volume = 268.1

1 .. cuboid(2, 4, 6)      2 .. cuboid(4)
3 .. ellipsoid(2, 4, 6)   4 .. ellipsoid(4)
5 .. cylinder(3, 4)       6 .. cone(6, 5)
0 .. exit
Your choice: 5
cylinder(3, 4): volume = 113.1

1 .. cuboid(2, 4, 6)      2 .. cuboid(4)
3 .. ellipsoid(2, 4, 6)   4 .. ellipsoid(4)
5 .. cylinder(3, 4)       6 .. cone(6, 5)
0 .. exit
Your choice: 6
cone(6, 5): volume = 37.7

1 .. cuboid(2, 4, 6)      2 .. cuboid(4)
3 .. ellipsoid(2, 4, 6)   4 .. ellipsoid(4)
5 .. cylinder(3, 4)       6 .. cone(6, 5)
0 .. exit
Your choice: 7
invalid input

1 .. cuboid(2, 4, 6)      2 .. cuboid(4)
3 .. ellipsoid(2, 4, 6)   4 .. ellipsoid(4)
5 .. cylinder(3, 4)       6 .. cone(6, 5)
0 .. exit
Your choice: 0
good bye
```

VolumesMenu.java:

VolumesMenu has 2 methods: main and menuPrompt

Ad menuPrompt:

The method menu has no parameters and a return type void. It prints the 4 lines of the menu and the prompt *Your choice:* (see Output).

Use the format specifier `%-25s` to print the strings of the left column left-aligned in a column of width 25

Ad main:

- Inside the main method create an instance of type Scanner. It is used to read in the user selection. Do not read in any user input. For this assignment we hard code the values that are passed to the methods of class Volumes.
- Declare a local variable of type int named selection
- Inside a do-while loop do the following:
  - Call the method menuPrompt
  - Read in the user choice and assign the result to the variable selection.
  - In a switch statement check for the number the user had entered
    - If it was 1 call the 3 parameter method cuboid and display the result; 'hard-code' 2, 4, and 6 as arguments
    - If it was 2 call the 1 parameter method cuboid and display the result; 'hard-code' the value 4 as argument
    - If it was 3 call the 3 parameter method ellipsoid and display the result; 'hard-code' 2, 4, and 6 as arguments
    - If it was 4 call the 1 parameter method ellipsoid and display the result; 'hard-code' the value 4 as argument
    - If it was 5 call the method cylinder and display the result; 'hard-code' 3 and 4 as arguments
    - If it was 6 call the method cone and display the result; 'hard-code' 6 and 5 as arguments

    Make the output look like the output on the left column. Pay attention to details like commas, numbers of digits in fractional part, new-lines etc.
- Repeat the do-while loop until the user enters 0 to exit.

Turning in:

Zip up your project and name the file **AVolumes** . Turn it in via Canvas.