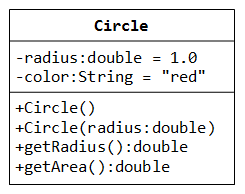
**Name:**

**Advanced Programming in Java**

**Lab Exercise 1/7/2025**

1. **The Circle Class**

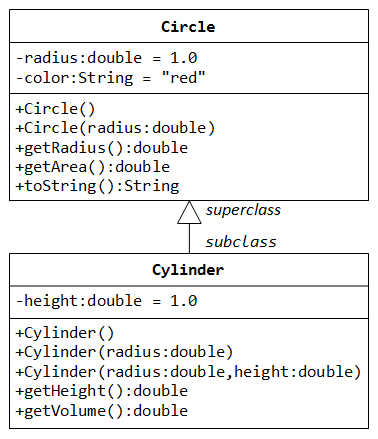


A class called **circle** is designed as shown in the following class diagram. It contains:

* Two private instance variables: radius (of type double) and color (of type String), with default value of 1.0 and "red", respectively.
* Two *overloaded* constructors;
* Two public methods: getRadius() and getArea().

Write a Circle class as well as a CircleTest class.

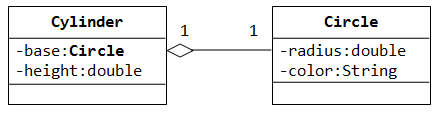
#### The Circle and Cylinder Classes Using Inheritance



In this exercise, a subclass called Cylinder is derived from the superclass Circle as shown in the class diagram (where an an arrow pointing up from the subclass to its superclass). Study how the subclass Cylinder invokes the superclass' constructors (via super() and super(radius)) and inherits the variables and methods from the superclass Circle.

You can reuse the Circle class that you have created in the previous exercise. Make sure that you keep "Circle.class" in the same directory.

#### Circle and Cylinder using Composition



Try rewriting the Circle-Cylinder of the previous exercise using composition (as shown in the class diagram) instead of inheritance. That is, "a cylinder is composed of a base circle and a height".

**Birthday Cake**

Create a function which constructs a *rectangular* birthday cake, based on someone's name and age! Build it out of *strings* in an array and make sure to surround the birthday message with the character that fits the rule:

If the age is an even number, surround the message with "#".

If the age is an odd number, surround the message with "\*".

Other important rules:

The message should be in the format: {age} Happy Birthday {name}! {age}

Leave a space between the edge of the cake and the age numbers.

Examples

getBirthdayCake("Jack", 10) ➞ [

"##############################",

"# 10 Happy Birthday Jack! 10 #",

"##############################"

]

getBirthdayCake("Russell", 19) ➞ [

"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*",

"\* 19 Happy Birthday Russell! 19 \*",

"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

]

getBirthdayCake("Isabelle", 2) ➞ [

"################################",

"# 2 Happy Birthday Isabelle! 2 #",

"################################"

]

Notes

The number of characters in the banner should be the same length as the message to pass the tests.