Object Oriented Programming Practicum

ICT2132

Diagramming Classes with UML

P.H.P. Nuwan Laksiri
Department of ICT
Faculty of Technology
University of Ruhuna

Lesson 03

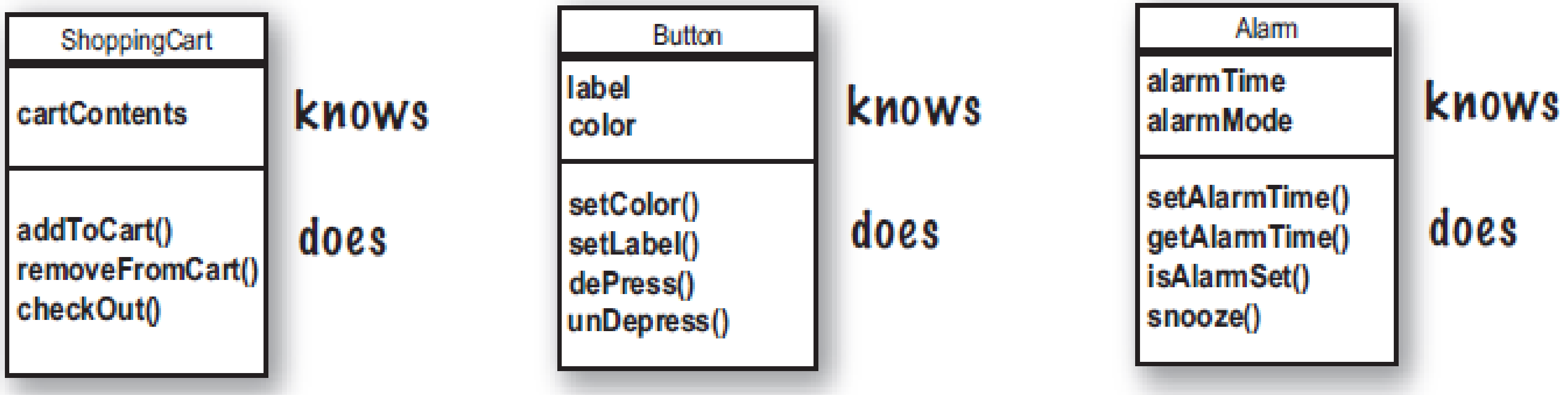
Outline

- Designing a Class
- Designing a Class - Example
- UML – Class Representation

Designing a Class

When you design a class, think about the objects that will be created from that class type. Think about:

- things the object **knows**
- things the object **does**



ShoppingCart

cartContents

addToCart()
removeFromCart()
checkOut()

knows

does

Button

label
color

setColor()
setLabel()
dePress()
unDepress()

knows

does

Alarm

alarmTime
alarmMode

setAlarmTime()
getAlarmTime()
isAlarmSet()
snooze()

knows

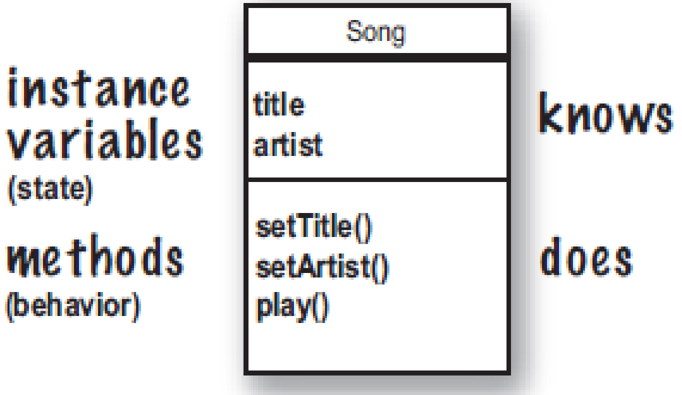
does

Things an object *knows* about itself are called

- instance variables

Things an object can *do* are called

- methods



instance variables (state)

methods (behavior)

Song

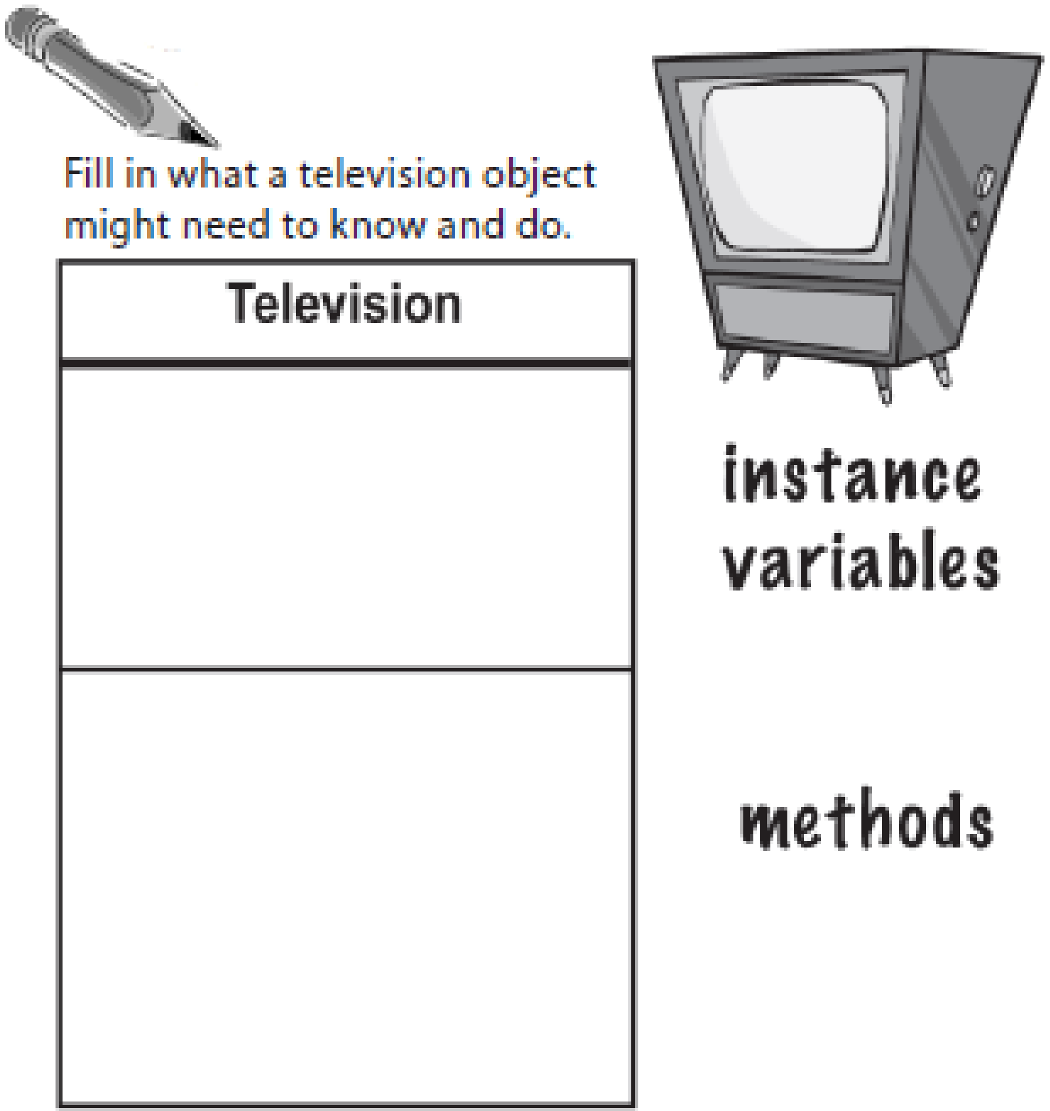
title
artist

setTitle()
setArtist()
play()

knows

does

Designing a Class - Example



Fill in what a television object might need to know and do.

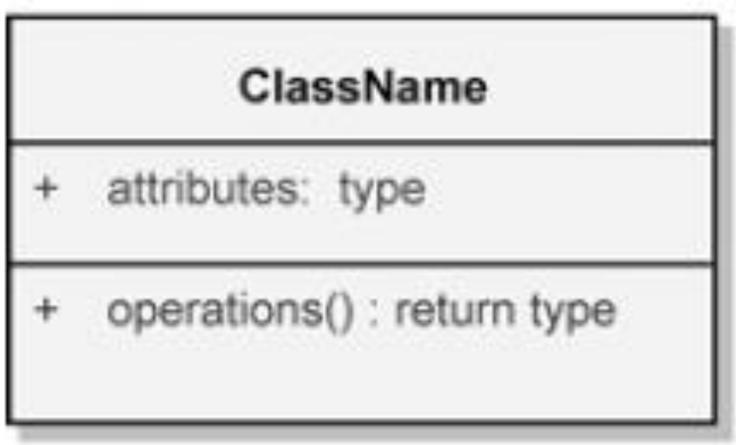
Television

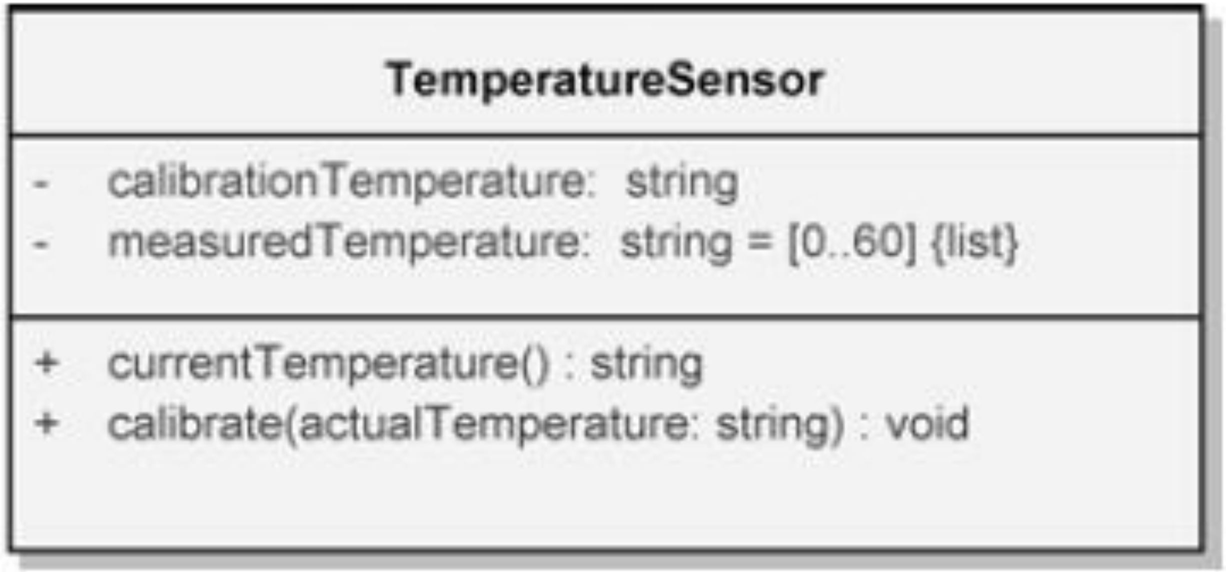
instance variables

methods

UML – Class Representation

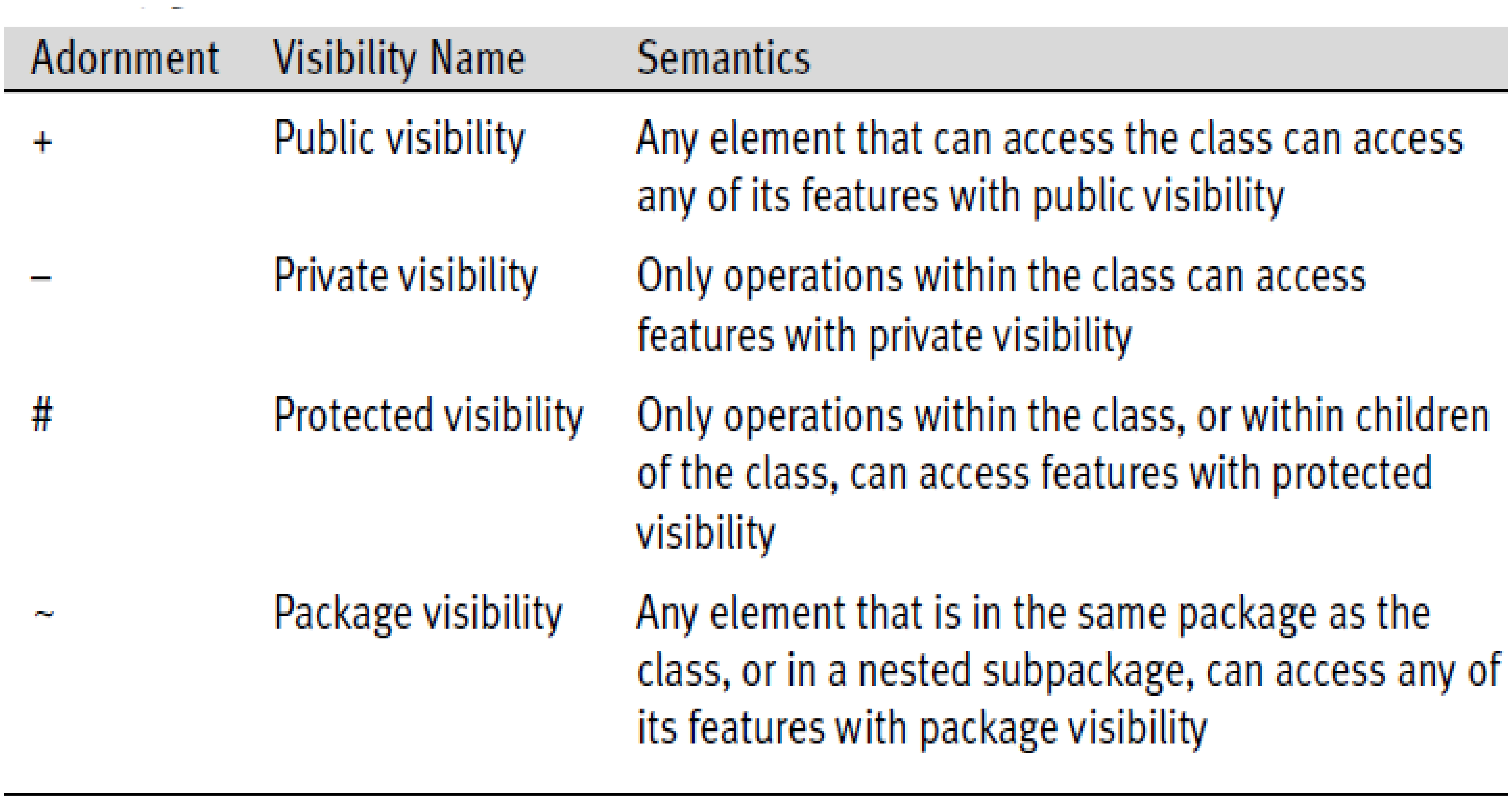
- Each class is represented by a rectangle, subdivided into three compartments.
  - Name
  - Attributes
  - Operations




TemperatureSensor
- calibrationTemperature: string
- measuredTemperature: string = [0..60] {list}
+ currentTemperature() : string
+ calibrate(actualTemperature: string) : void

UML – Class Representation

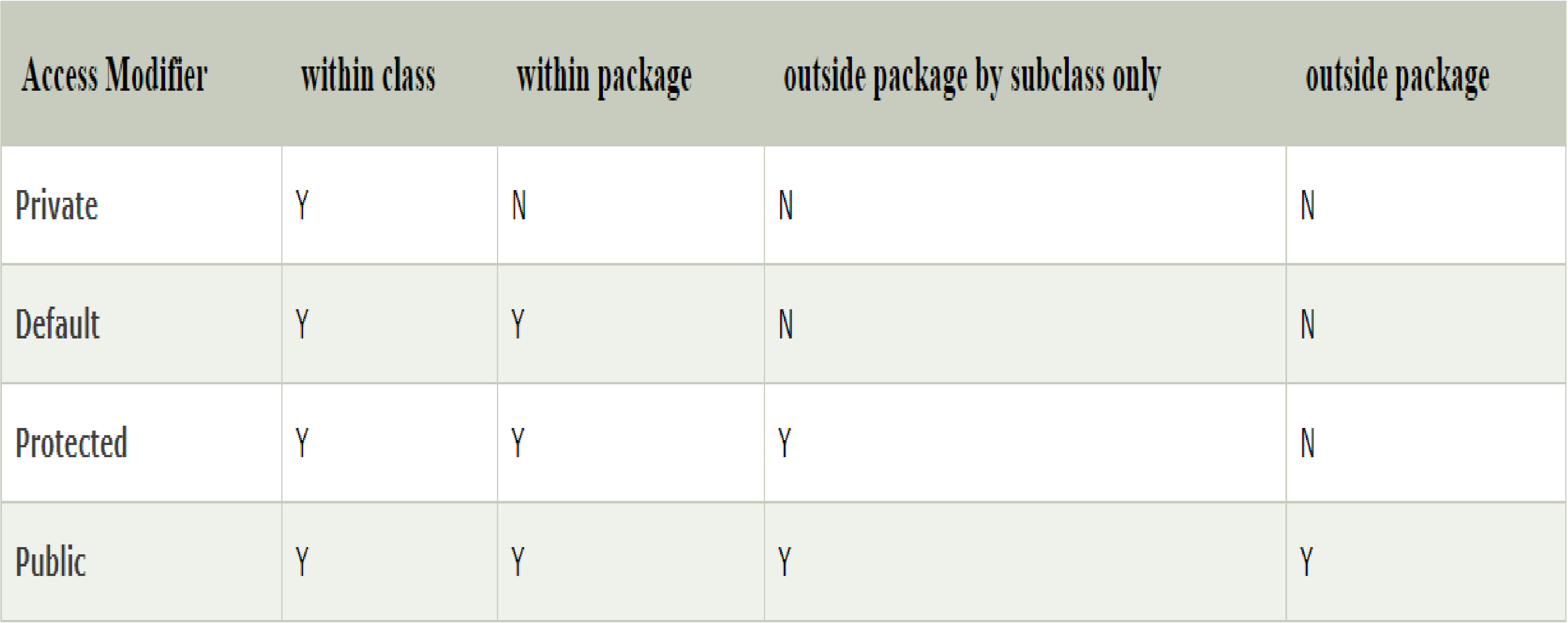
- Modifiers are used to indicate visibility of attributes and operations.



Adornment	Visibility Name	Semantics
+	Public visibility	Any element that can access the class can access any of its features with public visibility
–	Private visibility	Only operations within the class can access features with private visibility
#	Protected visibility	Only operations within the class, or within children of the class, can access features with protected visibility
~	Package visibility	Any element that is in the same package as the class, or in a nested subpackage, can access any of its features with package visibility

UML – Class Representation

- The access modifiers in java specifies accessibility (scope) of a data member, method, constructor or class.



Access Modifier	within class	within package	outside package by subclass only	outside package
Private	Y	N	N	N
Default	Y	Y	N	N
Protected	Y	Y	Y	N
Public	Y	Y	Y	Y

UML – Class Representation

Attributes (fields, instance variables)

visibility ***name*** **: data_*type***

Example:

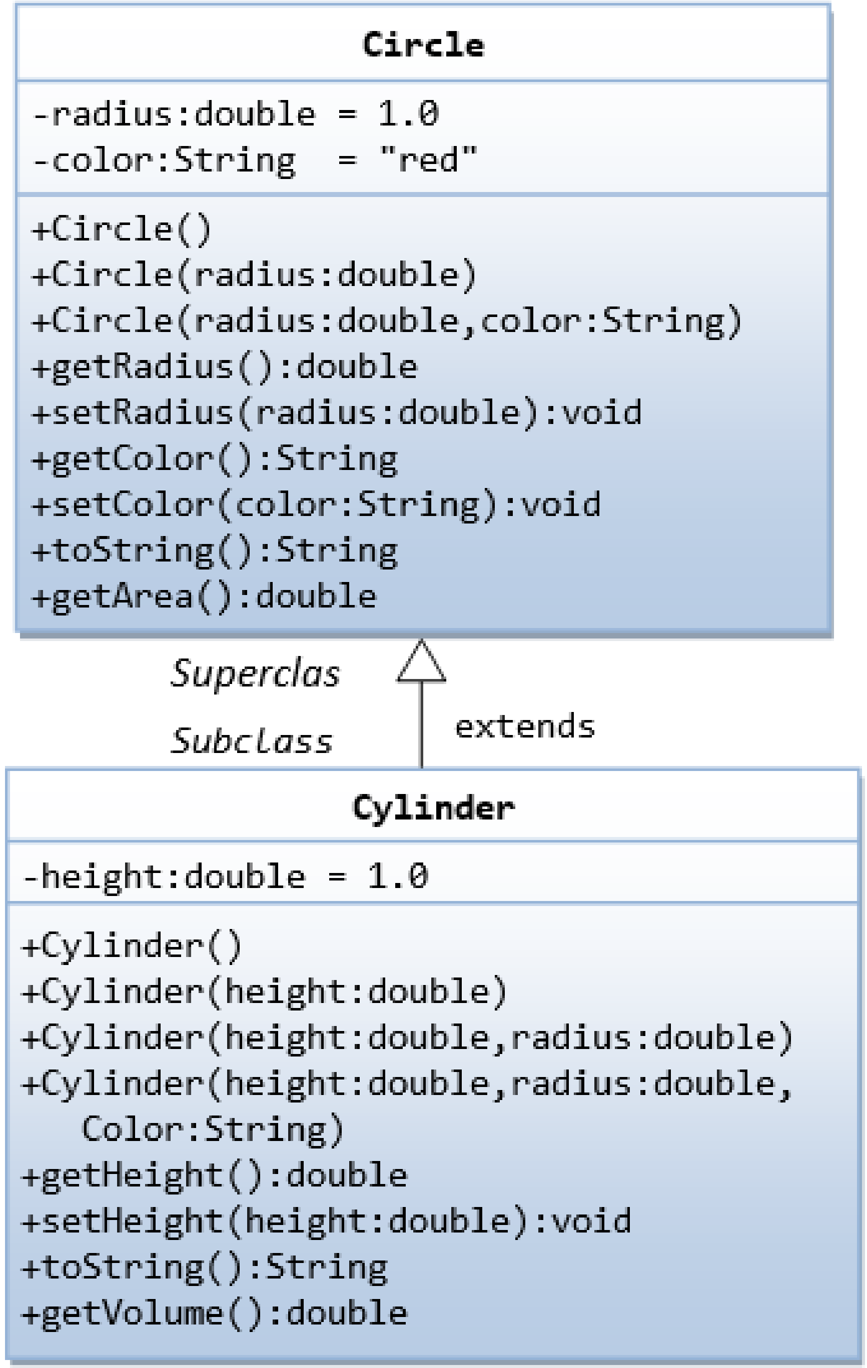
- balance : double

Operations / methods

visibility ***name*** **(*parameters*) : *return_type***

Example:

+ calDistance(p1: double): double



Circle

```
-radius:double = 1.0
-color:String  = "red"
```

```
+Circle()
+Circle(radius:double)
+Circle(radius:double,color:String)
+getRadius():double
+setRadius(radius:double):void
+getColor():String
+setColor(color:String):void
+toString():String
+getArea():double
```

Superclas

Subclass

extends

Cylinder

```
-height:double = 1.0
```

```
+Cylinder()
+Cylinder(height:double)
+Cylinder(height:double,radius:double)
+Cylinder(height:double,radius:double,
   Color:String)
+getHeight():double
+setHeight(height:double):void
+toString():String
+getVolume():double
```

Summary

- Designing a Class
- Designing a Class - Example
- UML – Class Representation

References

- https://www.visual-paradigm.com/guide/uml-unified-modeling-language/uml-class-diagram-tutorial/#uml-class-diagram-notation

Questions ???

Thank You