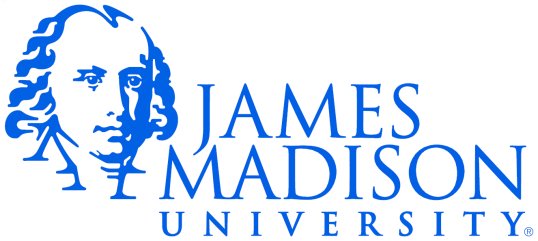
CS 149

Professor: Alvin Chao

CS149 – Even More Objects



Model 1 Variable Scope

	Where declared?	Where used?	Example
static variables ("class variables")	declared outside of all methods (typically at the start of the class)	anywhere in the class (or in other classes if also `public`)	`circleCount` in the `Circle` class
instance variables ("attributes")	declared outside of all methods (typically after any static variables)	any non-static method (or in static methods when another object has been created)	`radius` in the `Circle` class
parameters	declared inside the ()'s of a method signature	anywhere within the method where they are declared	`radius` in the `Circle` constructor
local variables	declared inside a method (or inside another block of code, like a `for` loop)	anywhere within the method or code block after they are declared	`temp` in the `swapInts` method

Previously we explored how classes define attributes and methods. Static variables and meth- ods apply to the whole class, whereas non-static variables and methods apply to specific objects.

Circle Class

- 1. Identify one static variable from the Circle class.
  - a) What is the name and purpose of the variable?
  - b) What is the scope of the variable?
  - c) What is one example of somewhere it cannot be used?
- 2. Identify one instance variable from the Circle class.
  - a) What is the name and purpose of the variable?
  - b) What is the scope of the variable?
  - c) What is one example of somewhere it cannot be used?

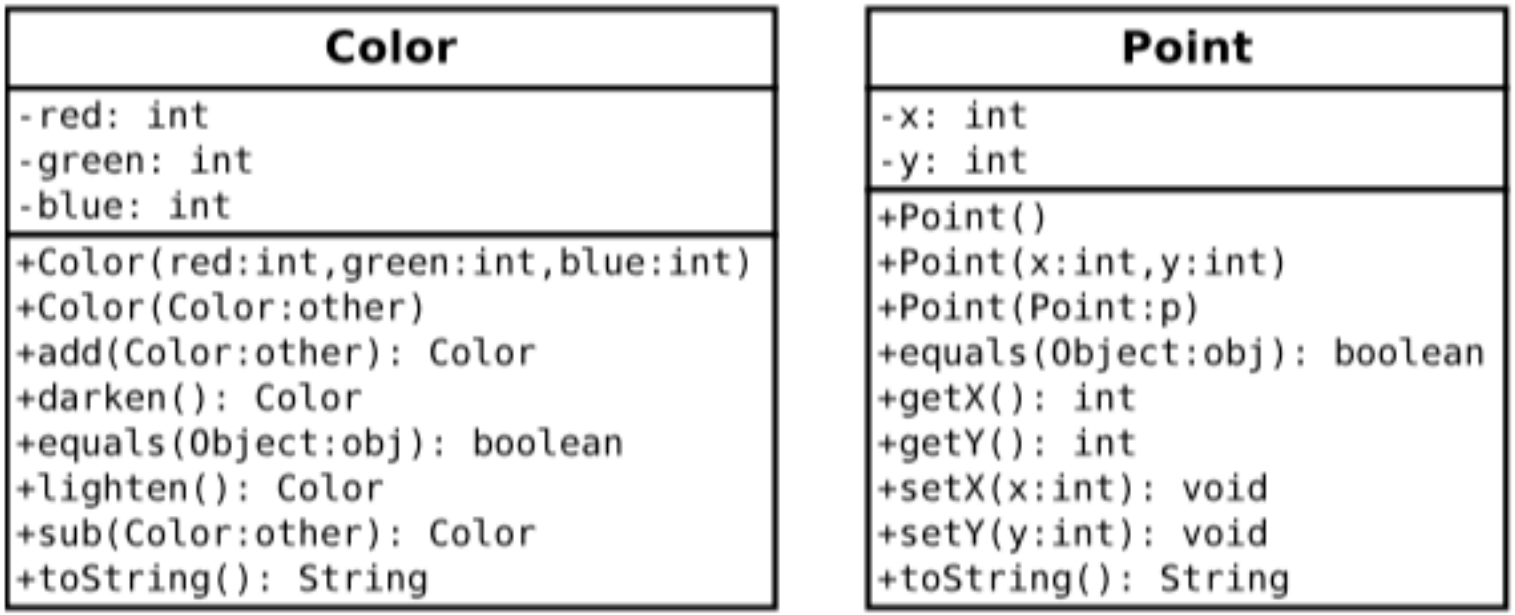
```
1  public class Circle {
2
3      private static int circleCount = 0;
4
5      private double radius;
6
7      public Circle(double radius) {
8          circleCount++;
9          if (radius > 0) {
10             this.radius = radius;
11         } else {
12             this.radius = 1;
13         }
14     }
15
16     public static int getCircleCount() {
17         return circleCount;
18     }
19
20     public double getRadius() {
21         return this.radius;
22     }
23
24     public static void swapInts(int x, int y) {
25         System.out.println("\tInside swapInts");
26         System.out.println("\tswapping integers " + x + " and " + y);
27         int temp = x;
28         x = y;
29         y = temp;
30         System.out.println("\tfinished swapping " + x + " and " + y);
31     }
32
33     public static void swapCircles(Circle c1, Circle c2) {
34         System.out.println("\tInside swapCircles");
35         System.out.println("\tswapping circles " + c1 + " and " + c2);
36         double r = c1.radius;
37         c1.radius = c2.radius;
38         c2.radius = r;
39         System.out.println("\tfinished swapping " + c1 + " and " + c2);
40     }
41
42     public String toString() {
43         return String.format("Circle(%.0f)", this.radius);
44     }
45 }
```

Swap Driver

- Predict the output of the SwapDriver program.
  Why are the results different when swapping integers and swapping Circle objects?

```
1  public class SwapDriver {
2
3      public static void main(String[] args) {
4
5          // first try swapping integers
6          int a = 7, b = 4;
7          System.out.println("BEFORE swap:");
8          System.out.println("a = " + a);
9          System.out.println("b = " + b);
10         Circle.swapInts(a, b);
11         System.out.println("AFTER swap:");
12         System.out.println("a = " + a);
13         System.out.println("b = " + b);
14         System.out.println();
15
16         // next try swapping Circle radii
17         Circle first, second;
18         first = new Circle(7);
19         second = new Circle(4);
20         System.out.println("BEFORE swap:");
21         System.out.println("first = " + first);
22         System.out.println("second = " + second);
23         Circle.swapCircles(first, second);
24         System.out.println("AFTER swap:");
25         System.out.println("first = " + first);
26         System.out.println("second = " + second);
27         System.out.println();
28
29         System.out.printf("This program created %d circles",
30                           Circle.getCircleCount());
31         System.out.println();
32     }
33 }
```

Model 2 Class Design



Classes generally include the following kinds of methods:

- **constructor** methods that initialize new objects
- **accessor** methods (getters) that return attributes
- **mutator** methods (setters) that modify attributes
- **object** methods such as equals and toString

Color and Point

- Identify the constructors for the Color class. What is the difference between them? What arguments do they take? What do these methods return?
- 8. Identify an accessor method in the Point class.
  - a) Which instance variable does it get?
  - b) What arguments does the method take?
  - c) What does the method return?
- 9. IdentifyamutatormethodinthePointclass.
  - a) Which instance variable does it set?
  - b) What arguments does the method take?
  - c) What does the method return?

Credit Card Design

10. List two or more attributes that would be necessary for this CreditCard class. For each attribute, indicate what data type would be most appropriate.

a)

b)

11. When constructing (or updating) a CreditCard object, what values would you need to check? What are the valid ranges of values for each attribute?

a)

b)

12. List two accessor methods would be appropriate for the CreditCard class. Include arguments and return values, using the same format as a UML diagram.

a)

b)

13. List two mutator methods would be appropriate for the CreditCard class. Include arguments and return values, using the same format as a UML diagram.

a)

b)

- **Acknowledgements**

Parts of this activity are based on materials developed by Chris Mayfield and Nathan Sprague.

</end>