INTPROG TB2: OBJECT ORIENTED PROGRAMMING

Practical Worksheet 9: Constructor and Method Overloading

1. Introduction

The aim of this practical session is to get experience with writing constructors and methods which have been overloaded. As overloading is essentially writing additional constructors and methods with different type signatures, there will not be an example this week – you should all be able to write constructors and methods quite easily now!

As usual, work through the practical questions at your own speed. Make sure you understand everything included within the example before you move onto the questions at the end of the worksheet. If you have questions, either ask your practical tutor, or ask a question using the anonymous question form on Moodle.

2. Practice Questions

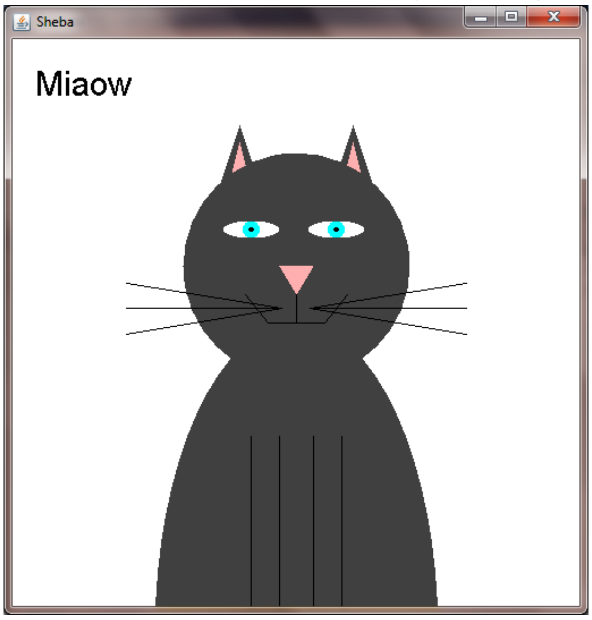
The following questions are designed to give you some practice at implementing method and constructor overloading. For all the questions, make sure that you include appropriate JavaDoc comments. If you have any questions, please make sure you ask the members of staff within the practical sessions.

1. Create a project called Rectangle. This project will need to have a rectangle class (which has two instance variables of type double, called side1 and side2). You will also need a RectangleMain class, which will contain the main method used to test all your overloaded methods and constructors are working as anticipated.
   Hint: Instead of calling your rectangle class "Rectangle", call it "RectangleClass". If you don't you will end up with an error in the Canvas class.
   (a) Overload the constructor with the following type signatures:
      i. Rectangle() – this should create a 3 x 4 rectangle as a default
      ii. Rectangle(double) – this should create a square
      iii. Rectangle(double, double) – this should create a rectangle with sides which are defined based on the parameters
   (b) Write the following overloaded methods (for the purpose of this part of the question, you can assume that all measurements are in centimetres):
      i. getArea() – this should calculate the area of the rectangle
      ii. getArea(double) – this should calculate the area of the rectangle using the conversion factor (which is the parameter)
      iii. getPerimeter() – this should calculate the area of the rectangle
      iv. getPerimeter(double) – this should calculate the area of the rectangle using the conversion factor (which is the parameter)
   (c) Write 4 methods which can be used to draw the chosen rectangle in a window:
      i. drawRectangle() – draws the rectangle in red in a 200 x 200 window
      ii. drawRectangle(Color) – draws the rectangle with the chosen fill colour, in a 200 x 200 window
      iii. drawRectangle(Color, int, int) – draws the rectangle with the chosen fill colour, in a window which is sized depending on the parameters entered when the method is executed. For this one, you will need to work out an appropriate scaling mechanism to make sure that rectangle is drawn appropriately.

2. Download the Cat project from Moodle. Inspect the code and ensure you understand what it is doing.

   (a) Overload the constructor so that there are two more options for initialising the Cat:
      i. One where the name of the cat is entered as a parameter
      ii. One where the name of the cat, the colour of its fur and the colour of its bow tie are entered as parameters

   (b) Write some overloaded methods which makes the cat say something, see the below image.



   You will need 2 methods:
      i. Default method, with no parameters, which makes the cat say "Miaow"
      ii. Method with a single parameter, which defines what the cat must say.

```
/* 2a */
public Cat(String catName)
{
    name = catName;
    fur = Color.ORANGE;
    canvas = new Canvas(name, 500, 500);
}

public Cat(String catName, Color catFur)
{
    name = catName;
    fur = catFur;
    canvas = new Canvas(name, 500, 500);
}
```

```
/* 2b */
public void saySomething()
{
    saySomething("Miaow");
}

public void saySomething(String words)
{
    canvas.setFontSize(30);
    canvas.drawString(words, 20, 50);
}
```