CSC1016S Assignment 5: Polymorphism, Subtyping and Inheritance

Assignment Instructions

This assignment involves constructing programs in Java using polymorphism, sub classes and inheritance.

Question 1

Write a program to demonstrate the use of inheritance by creating and outputting 3 simple objects, where the classes for the second and third objects inherit from the class for the first object.

The first object is a Vehicle with a number of passengers and colour. The second object is a Car with additional number of doors. The third object is a Plane with a manufacturer and model number. Note: do not inherit Plane from Car!

Use the provided *Question1.java* and *Vehicle.java* files. Use constructors and override methods as appropriate.

Sample Output:

```
Blue 2 passengers
Black 4 passengers 4 doors
White 416 passengers Boeing 737
```

Question 2

Write a program to store some details of pizzas (menu item number, size, base, extra cheese, extra garlic), curries (menu item number, size, curry type) and softdrinks (menu item number, size, flavour, bottle or can) in a single array, with the options of listing all menu items or deleting a particular menu item.

Your program must continuously prompt the user to choose an option from a list and act on that option, until the exit option is chosen. (See the sample output below.) If a menu item is not found, output "Not found" instead of "Done".

You must use inheritance and polymorphism to model your Pizza, Curry and SoftDrink classes (use those exact class names) as subclasses of the same base class, which forms the basis for the array.

Note: Be very careful to reproduce the output exactly. Copy-and-paste is highly recommended to avoid minor typographical errors that drive you crazy when submitting online!

Sample Input/Output:

```
Welcome to Great International Food Court
MENU: add (P)izza, add (C)urry, add (S)oft drink, (D)elete, (L)ist,
(Q)uit
p
Enter the menu item number
123
Enter the size
12"
Enter the base
Hand-tossed
Enter extra cheese
Yes
```

CONTINUED

```
Enter extra garlic
Yes
Done
MENU: add (P)izza, add (C)urry, add (S)oft drink, (D)elete, (L)ist,
(Q)uit
c
Enter the menu item number
456
Enter the size
Large
Enter the curry type
Vindaloo
Done
MENU: add (P)izza, add (C)urry, add (S)oft drink, (D)elete, (L)ist,
(Q)uit
s
Enter the menu item number
789
Enter the size
Large
Enter the flavour
Coke
Enter whether it is a bottle or can
Bottle
Done
MENU: add (P)izza, add (C)urry, add (S)oft drink, (D)elete, (L)ist,
(Q)uit
d
Enter the menu item number
456
Done
MENU: add (P)izza, add (C)urry, add (S)oft drink, (D)elete, (L)ist,
(Q)uit
l
Pizza: 123, 12", Hand-tossed, Yes, Yes
Soft Drink: 789, Large, Coke, Bottle
Done
MENU: add (P)izza, add (C)urry, add (S)oft drink, (D)elete, (L)ist,
(Q)uit
q
```

CONTINUED

Question 3

Write a program to manage a set of graphical objects based on the commands in a specified file. This is a graphical equivalent of the famous edlin tool for text manipulation.

On the Vula page you will find VectorGraphics and Question3 classes that will handle the file I/O for you. When you run Question3, specify the file name like this:

```
java Question3 myfile.in
```

Each line of the file contains an instruction in one of the formats given below:

```
a <id> <x> <y> rectangle <x_length> <y_length>
a <id> <x> <y> hline <x_length>
a <id> <x> <y> vline <y_length>
a <id> <x> <y> ptline <x1> <y1>
w
d <id>
m <id> <x> <y>
x
```

Explanation:

'a' adds an object of a particular type (rectangle/hline/vline/ptline) with the specified parameters.

'w' renders/draws the objects and prints the composite image to the screen.

'd' deletes an object based on its id.

'm' moves an object to a new position based on its id.

'x' exits the program.

Note:

- A 'ptline' differs from the other objects in that the values after the object type are not lengths, but instead a second point described by *x1* and *y1*. This object represents a line from *(x, y)* to *(x1, y1)*.
- The answers the automatic marker expects were generated using an implementation of Bresenham's Line Drawing Algorithm. Consult this Wikipedia page for information on the algorithm.

Assume that the drawing canvas is 20 blocks wide (in the *x* direction) and 20 blocks high in the *y* direction), starting with (0,0) in the top left.

As noted, on the Vula page you will find classes that serve as a framework for this program. You are only required to create the Rectangle, HLine, VLine and PtLine subclasses of the given VectorObjectbase class. You should add comments to all classes you write.

Sample Input File:

```
a 1 5 5 rectangle 10 10
a 2 3 3 hline 14
a 3 3 16 hline 14
a 4 3 3 vline 14
a 5 16 3 vline 14
w
```

CONTINUED

```
d 1
m 2 3 5
m 3 3 14
m 4 5 3
m 5 14 3
a 6 6 6 ptline 13 13
a 6 13 6 ptline 6 13
w
x
```

Sample Output:

```
+--------------------+
+                    +
+                    +
+                    +
+    **************  +
+    *            *  +
+    * ********** *  +
+    * ********** *  +
+    * ********** *  +
+    * ********** *  +
+    * ********** *  +
+    * ********** *  +
+    * ********** *  +
+    * ********** *  +
+    * ********** *  +
+    *            *  +
+    **************  +
+                    +
+                    +
+                    +
+--------------------+
+--------------------+
+                    +
+                    +
+                    +
+      *         *   +
+      *         *   +
+    **************  +
+      **       **   +
+      * *     * *   +
+      *  *   *  *   +
+      *    **    *   +
+      *    **    *   +
+      *  *   *  *   +
+      * *     * *   +
+      **       **   +
+    **************  +
+      *         *   +
+      *         *   +
```

CONTINUED

Submit your assignment as a set of Java files named Car.java, Plane.java, Question2.java, Rectangle.java, HLine.java, VLine.java, PtLine.java containing classes with corresponding names, as well any other Java files you have created, in a Zip file.

Marking and Submission

Submit Car.java, Plane.java, Question2.java, Rectangle.java, HLine.java, VLine.java, PtLine.java (as well any other Java files you have created) in a single .ZIP folder to the automatic marker.

The zipped folder should have the following naming convention:

yourstudentnumber.zip

END

CONTINUED