Seneca College

Sept 23, 2016

Applied Arts & Technology Fall/16

SCHOOL OF COMPUTER STUDIES

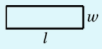
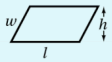
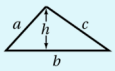
JAC444 **Due date: Oct 14**

Assignment 1

The first assignment contains two parts. It gives you the opportunity to practice basic Java concepts by implementing known structures from C++ such as **`singly linked list`**

Part A – 2 marks

Create an interface called **`Shape`** with the method **`getPerimeter`** that determines the perimeter of a geometrical shape. Develop classes such as **`Circle, Square, Rectangle,`** and **`Triangle`** that implements the **`Shape`** interface.

OBJECT	PICTURE	PERIMETER	AREA
Circle		$2\pi r = \pi d$	πr^2
Square		$4l$	l^2
Rectangle		$2l + 2w$	lw
Parallelogram		$2l + 2w$	lh
Triangle		$a + b + c$	$\frac{1}{2}bh$

Every class must have methods such as: ***`toString()`***, ***`equals(), hashCode()`*** and must have javadoc style documentation.

Part B – 8 marks

Develop a Java class named ***`ShapeLinkedList`*** that implements a single linked list. It must be capable of storing geometrical shapes.

```
/**
 * Link list containing geometrical Shapes
 */

public class ShapeLinkedList {

        public Node head; // Head is first node in linked list

        public ShapeLinkedList() { }

        public ShapeLinkedList(Node head) {
```

```
        }

        public boolean isEmpty() {
                return length() == 0;
        }

        public void insertAtEnd(Shape data) {
                // TODO to be implemented
        }

        public void insertAtBeginning(Shape data) {
                // TODO to be implemented
        }

        public Node tail() {
                // TODO to be implemented
                // returns the last node
        }

        public int length() {
                // TODO to be implemented
        }

        void insertAtIndex(int idx, Shape data) {
                // TODO to be implemented
        }

        Node findAtIndex(int idx) {
                // TODO to be implemented

        }

        void deleteAtIndex(int idx) {
                // TODO to be implemented
        }

        @Override
        public String toString() {
        }

        void deleteData(Shape s) {
                // TODO to be implemented
        }


        @Override
        public int hashCode() {
                // TODO to be implemented
        }

        @Override
        public boolean equals(Object obj) {
                // TODO to be implemented
        }

        // Node is nested class because it only exists along with linked list
        public static class Node {

                private Shape data;
                private Node next;

                // TODO develop all the methods that are needed
                // such as constructors, setters, getters
                // toString, equals, hashCode

        }
}
```

Your task is to create geometrical shapes, to store them in the link list and to calculate their perimeters.

Task 1: Try to create the following shapes and store them in `ShapeLinkedList` .
For example, given the integer array:

```
int[] values = {2,1,3,5,1,4,5,3,5,7,1,2,8,9};

ShapeLinkedList sll = new ShapeLinkedList();


Circle c1 = new Circle(values[0]);
Circle c2 = new Circle(values[1]);

Square sq1 = new Square(values[2]);
Square sq2 = new Square(values[3]);

Triangle t1 = new Triangle(values[4],values[5],values[6]);
Triangle t2 = new Triangle(values[7],values[8],values[9]);

Rectangle r1 = new Rectangle(values[10],values[11]);
Rectangle r2 = new Rectangle(values[12],values[13]);


sll.insertAtBeginning(r1);
sll.insertAtBeginning(r2);
sll.insertAtBeginning(c1);
sll.insertAtBeginning(c2);
sll.insertAtEnd(sq1);
sll.insertAtEnd(sq2);
sll.insertAtEnd(t1);
sll.insertAtEnd(t2);
```

Then print all the shapes and their perimeters from the list.
(For example: `Circle{r=1} has perimeter: 6.28318`)

Task 2: Delete the list tail and print the list.

Task 3: Delete the shape called `sq2` and print the list.

Marking criteria:

1. Every class must be properly documented using javadoc style created in a package having your Last Name followed by the first 3 digits of your student ID.
2. Every class must implement `toString, equals`, and `hashCode`
3. The implementation of all linked list methods
4. The mandatory usage of the data given in the task1 to 3.

Deliverables (both mandatory):

1. Zip only the Java files to file with a special name. The file should be named after your Last Name followed by the first 3 digits of your student ID. For example, if your last name is `Savage` and your ID is `354874345` then the file should be called `Savage354.zip` Upload the zip file to Blackboard

2. On your course page at the end you will find a section called Assessment. There you will find a codeboard.io project called Assignment 1 with the skeleton of your assignment.

   Cut and paste your code with documentation (it must be the same as in your zip file). The code must completely compile and run on codeboard.io