JAV745 Fall 2019: Lab 2 - (4%)

Dr. Eden Burton
School of ICT, Seneca College of Applied Arts and Technology
Fall 2019

Due Tuesday October 1, 2019 - 11:30 pm

Instructions

Please read the instructions carefully and follow the naming conventions specified for each question. Solutions must be submitted in the Blackboard Dropbox created for Lab 2.

Note that the deadline is strictly enforced. The system tracks the exact time that submissions are uploaded. **There is a 10% per day penalty for late submissions.**

Additional Notes

- You will be required to demonstrate your work to the professor. Be prepared to answer questions about the code you have written.
- You may use any IDE for development but note that demonstrations and professor testing will be done exclusively on the command line. Ensure that you test your programs on the command line before submission.
- Ensure that your programs are documented using *JavaDoc* standards

Question Descriptions

Question 1) Modify the BankTeller application coded in Session 3 (it is posted in the Course Document section of Blackboard) to use objects instead of multiple arrays to store account information. Your application should store a single array of account objects.

The following requirements should also be satisfied.

(a) an account withdraw and account deposit must be supported
(b) a "show balance" choice should be added to the application and implemented.
(c) account balance should never go below zero. Transactions that cause this should be rejected
(d) opening date and time of last transaction should be stored with each account. This information should be added to the transaction confirmation text shown to the user.

Question 2) Create a class hierarchy which models points in a Cartasian coordinate system `https://www.mathsisfun.com/data/cartesian-coordinates.html`. Some points are in a two-dimensional space (where x and y coordinates are needed) while others are in a three-diminsional space (where x, y and z coordinates are needed).

Some (not all) of objects require a point to be coloured. There are two colour systems available.

- *RGB*, This model stores a red, green and blue component. In standard implementations, the intensity of each base colour ranges from 0-255. White is represented with all base colours set at 255, or full intensity. Black is with all base colours set at 0, or no intensity.
- *CMYK*. This subtractive model is used for printing applications, and has cyan, magenta, yellow and black components. Each component is expressed as a percentage (0%-100%) of the ink required to create a colour. White therefore requires no ink, so all components would be set to 0%.

Your class hierarchy should be documented in a *UML class diagram*. The diagrams should be created using *StarUML* or some other UML-compliant drawing tool.

Generate Java code with definitions of your classes from your diagram. This should include....

- a constructor that allows the user to fully initialize the object. Colored points should have an additional constructor which allows the user to omit the colour information. In this case, the colour should be set to white.
- a member function which allows the point to be moved anywhere on the plane.
- a member function which allows the user to change the colour.
- a means of printing out the number of white points in the system (this must be embedded in the classes!!!)

Additional notes...

Your classes should not allow the user to set invalid values. It is sufficient to print a warning to the console and disallow the change.

You must also create an application to test your class. It should create at least the following

- points in a two dimensional (2D) space
- points in a three dimensional (3D) space
- 2D points coloured using the RGB model
- 3D points coloured using the RGB model
- 2D points coloured using the CMYB model
- 3D points coloured using the CMYB model
- some white and non white points to demonstrate that the number counting functionality works correctly.

Submit only the Java source files. Submit files individually. DO NOT submit a zip file. Submit a pdf file with your UML class diagram embedded