**Name: MINH DUC NGUYEN**

**CSC 143 Winter 2021**

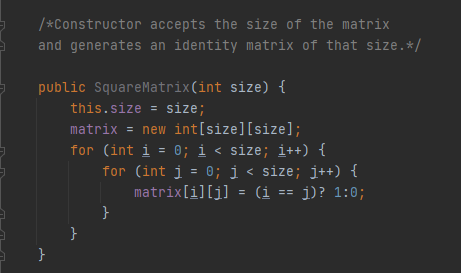
**LAB 02: Objects, Inheritance-Interface**

**MATRIX OBJECT**

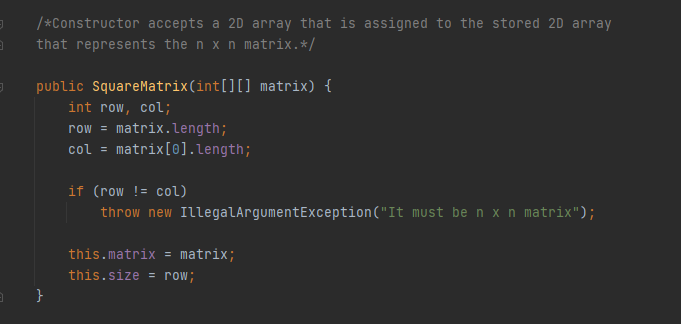
**SquareMatrix class:**

***Constructors:***

SquareMatrix(int size): generates an identity matrix of that size.

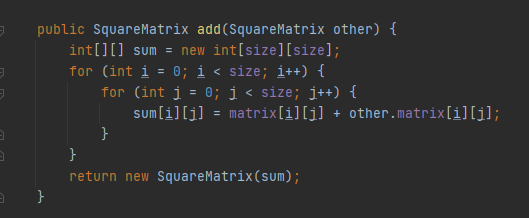


SquareMatrix(int[][] matrix): represents the n x n matrix

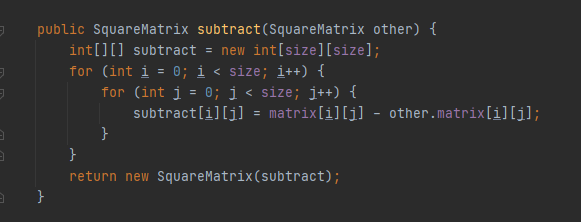


***Methods:***

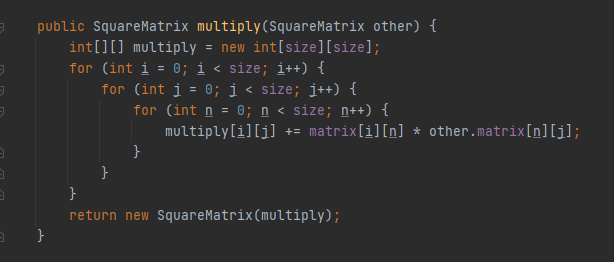
Add: add two matrices

****

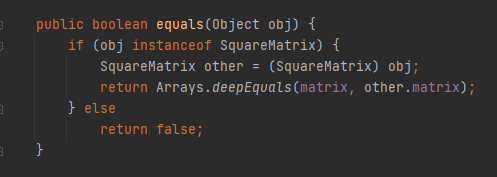
Subtract: subtract two matrices



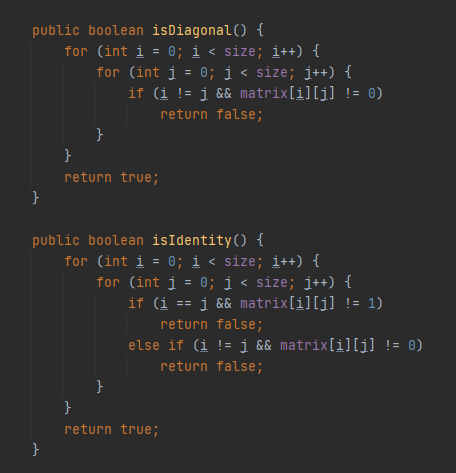
Multiply: multiply two matrices



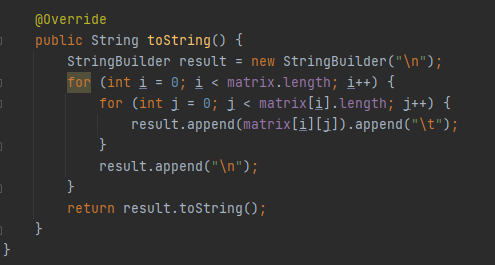
Equals: check if two matrices are identical whether or not



isDiagonal and isIdentity: check if the matrix is diagonal or identity



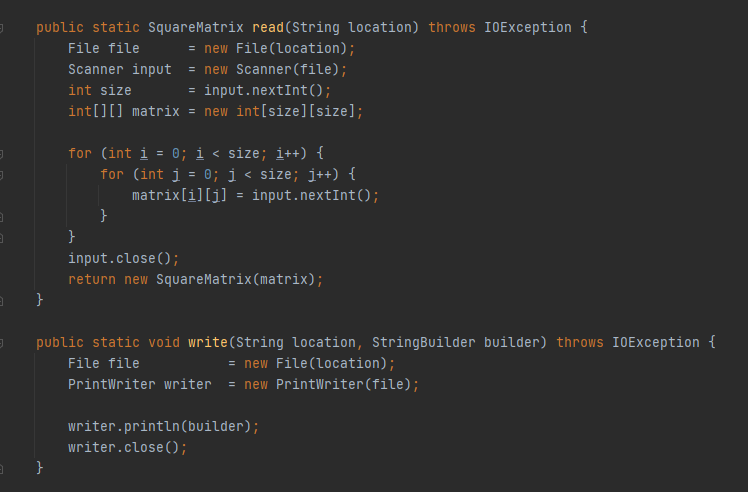
toString: represent the matrix



**MainProgram class:**

***Methods:***

read and write methods: to read the data in the input file, and write the output into a new output file

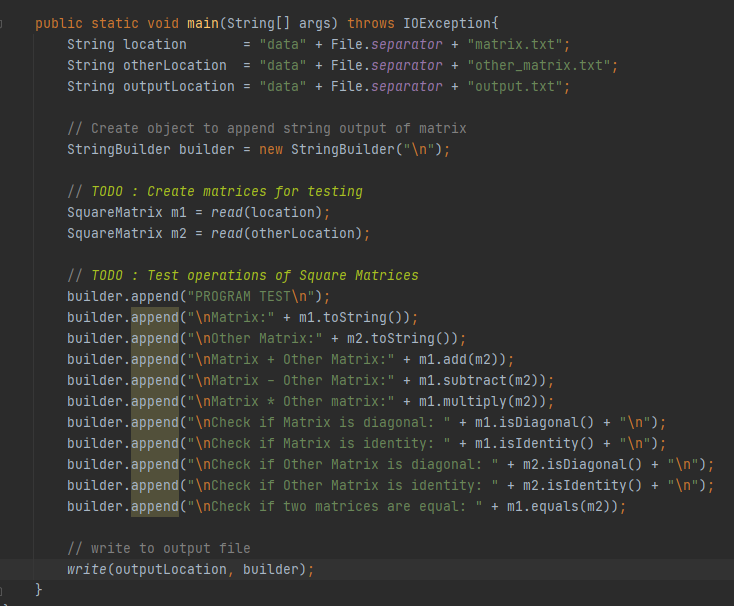


Main():

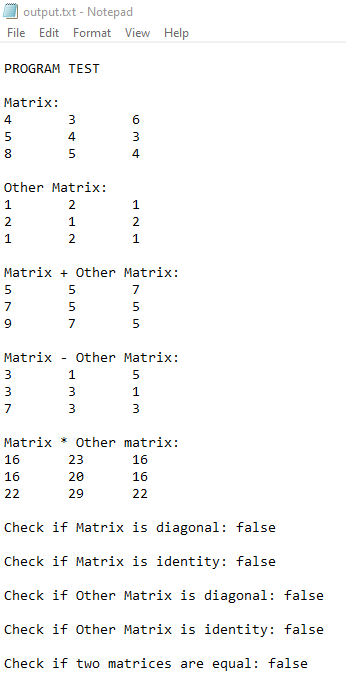
First, read the given data in the input files.

Then, use StringBuilder to append the matrices, and test the operations to make sure everything works correctly.

After that, use write method to put all the things above into the output file.



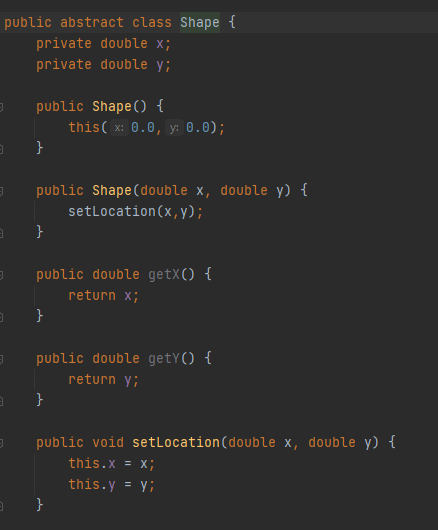
***OUTPUT:*** output of SquareMatrix Program test

******

**INHERITANCE/POLYMORPHISM**

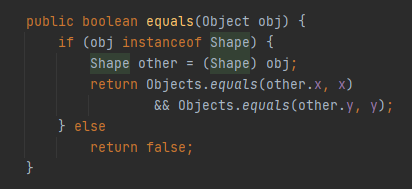
**Shape class:** It is an abstract class

***Constructors and getter methods:***



***Methods:***

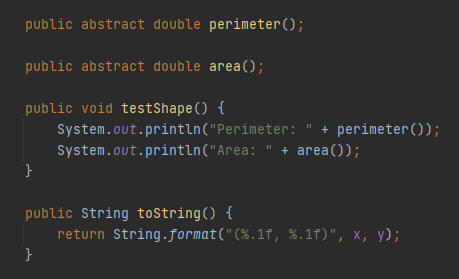
Equals: check if two shapes’ contents are equal or not

****

Abstract methods perimeter and area: return the perimeter and area of the shape (specific shapes declared in subclasses)

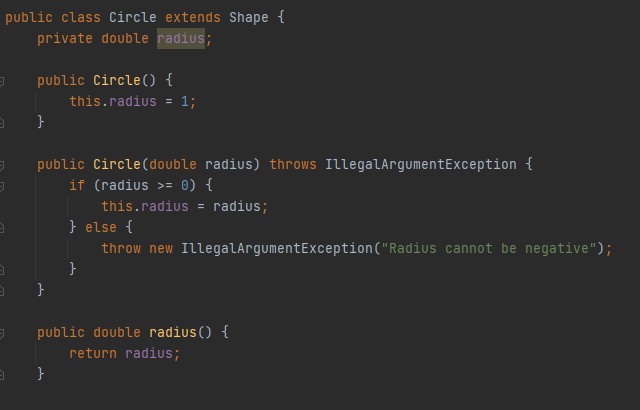
testShape: print out the perimeter and area of the shape (along with other shapes’ information, in subclasses)

toString: represent the shape’s data

****

**Circle class:** extends the Shape class

***Constructors and getter method:***

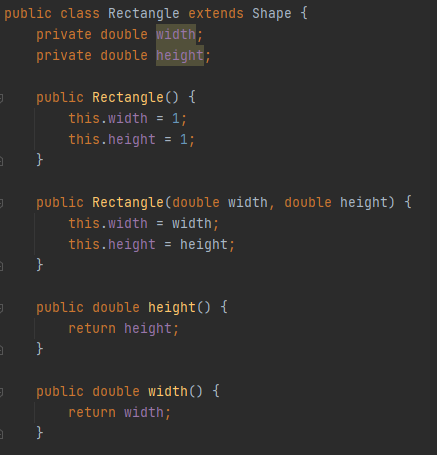


**Methods:** Override the methods in abstract class Shape, with detailed operations

****

**Rectangle class:** extends the Shape class

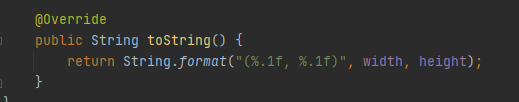
***Constructors and getter methods:***

****

***Methods:*** Override the methods in the abstract class Shape, with detailed operations

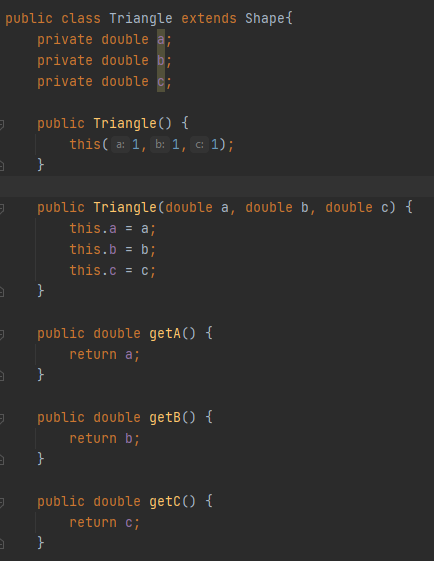


toString:



**Triangle class**: extends the Shape class

***Constructors and getter methods:***

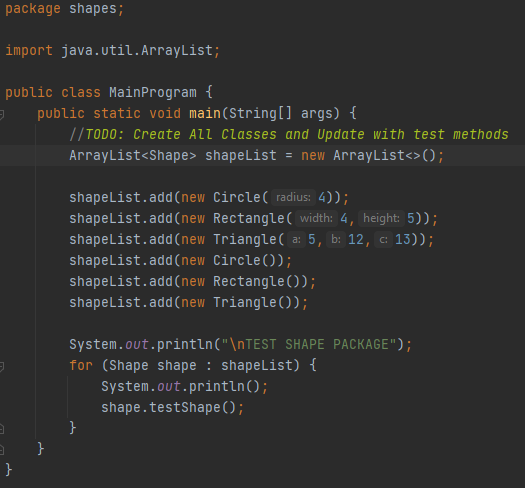
****

***Methods:*** Override the methods in the abstract class Shape, with detailed operations

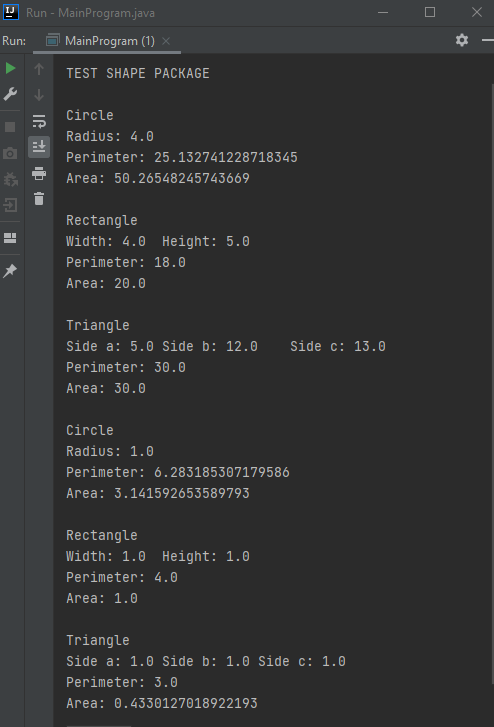
****

**MainProgram class:**

Test the above classes’ operations



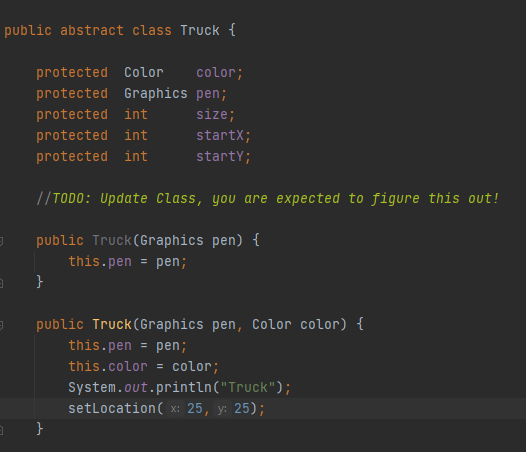
***OUTPUT:***



**GUI / INHERITANCE**

**Truck class:** It is an abstract class

***Constructors and Data Fields:***



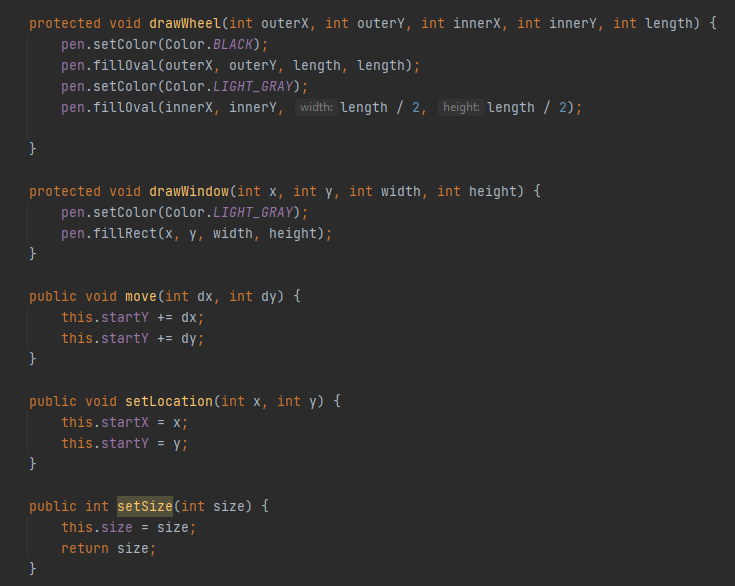
***Methods:***

drawWheel, drawWindow: to draw parts of the truck

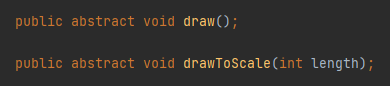
move: move graphics object to a different location in GUI Window

setLocation: set the starting location of the object

setSize: set size to scale and draw graphics object to

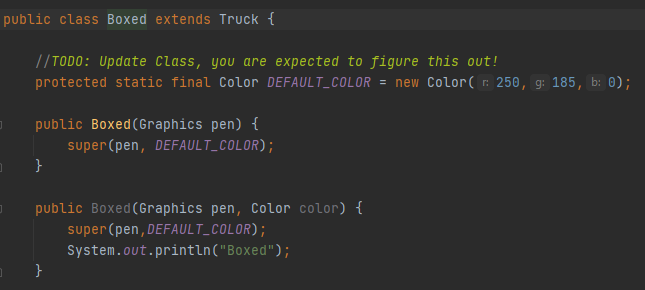


draw and drawToScale: abstract methods of abstract class Truck



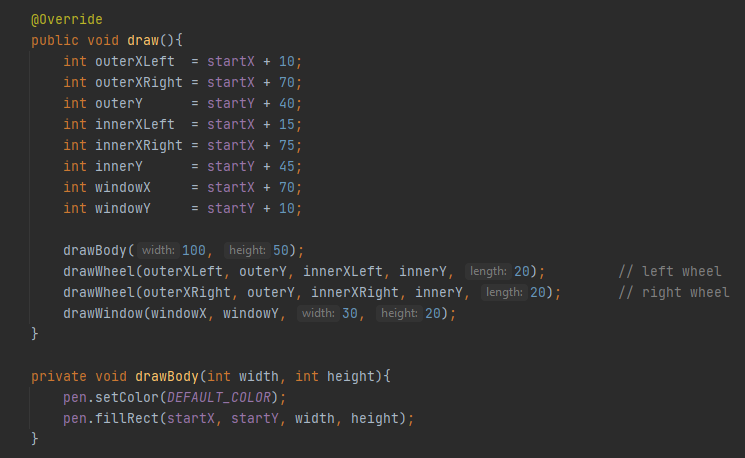
**Boxed class**: extends the Truck class

***Constructors:*** use the color YELLOW to make the “Boxed” Truck

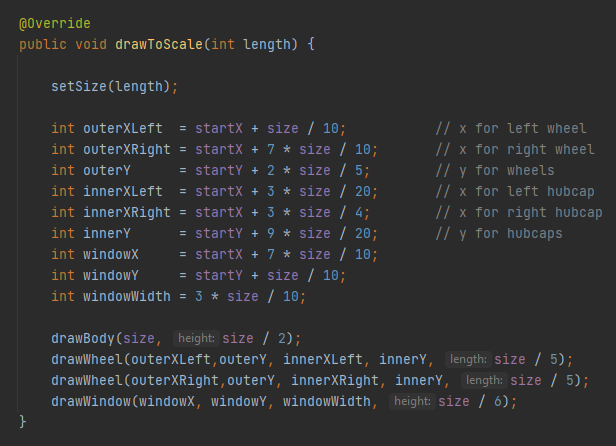


***Methods:***

Override the draw method in Truck, with detailed actions, using graphics object

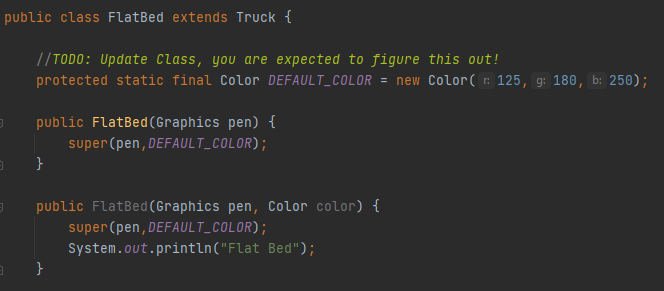
******

Override the drawToScale method in Truck, resize the "Boxed" truck with graphics object



**FlatBed**: extends the Truck class

***Constructors:*** use the color LIGHT BLUE to make the “FlatBed” Truck



***Methods:***

Override the draw method in the abstract class Truck, with helper method drawFlatBedBody, to draw the FlatBed Truck using graphic objects.



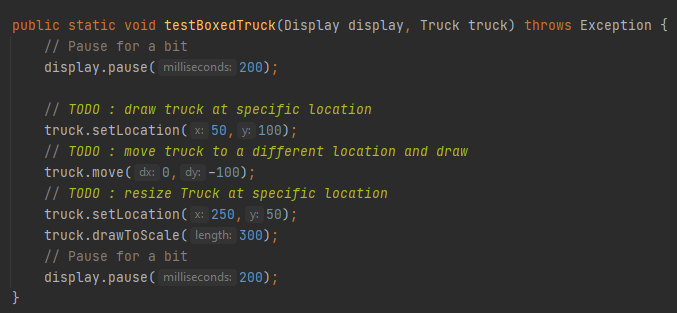
Override the drawToScale method in Truck, with the helper method drawResizeBody, to resize the "FlatBed" truck with graphics object



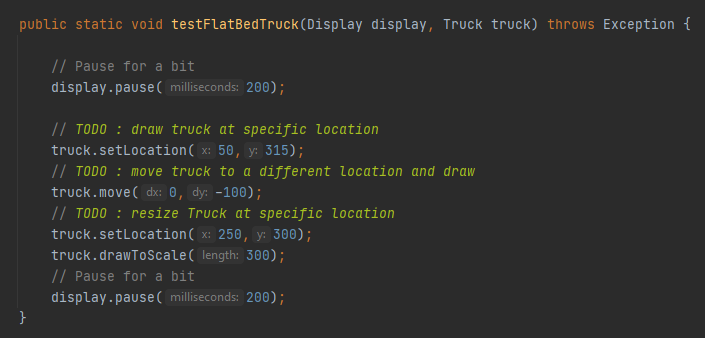
**Display class:** already be given by instructor

**MainProgram class:**

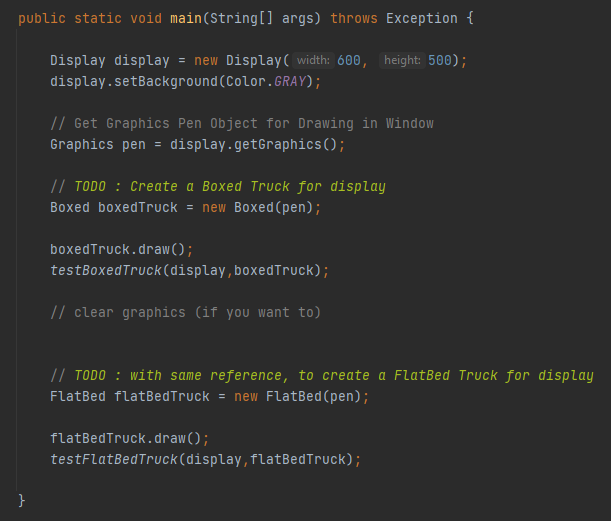
**testBoxedTruck:** Update the test method for Boxed Truck

****

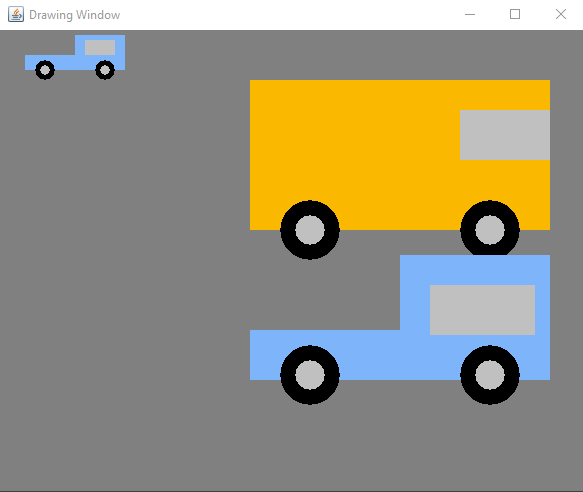
**TestFlatBedTruck:** Update the test method for FlatBed Truck

****

**Main:** create Boxed and FlatBed Truck objects, and do the above tests

****

***OUTPUT:***

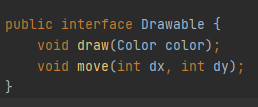


**INTERFACE:**

**Drawable Interface:** create an interface with two methods draw and move

Draw: draw outlines of graphics object in GUI Window

Move: translate object to a different location

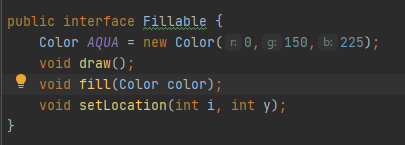
****

**Fillable Interface:** create an interface with three methods and one constant color AQUA

Draw: draw outlines of graphics object using a color AQUA in GUI Window

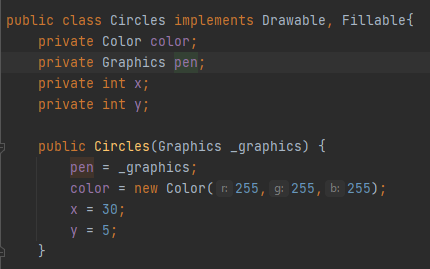
Fill:  draw a sequence on circles in GUI Window.

setLocation: set the starting location of circle

****

**Circles class:** implements the two above interfaces

***Constructors:***



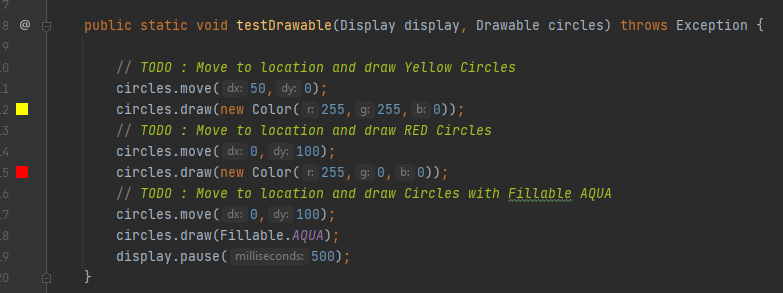
***Methods:*** Override the methods in the interfaces using graphic objects



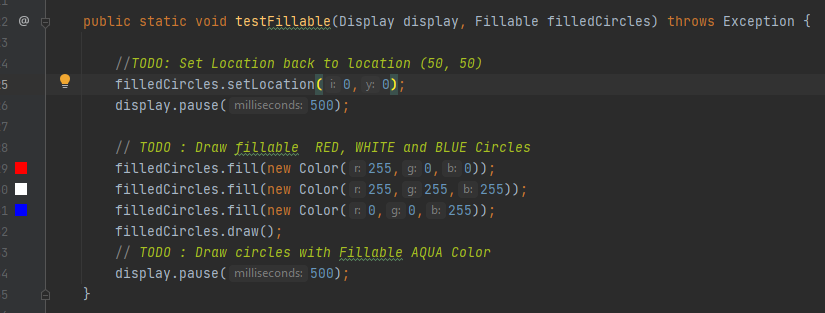
**Display class:** already be given

**MainProgram class:**

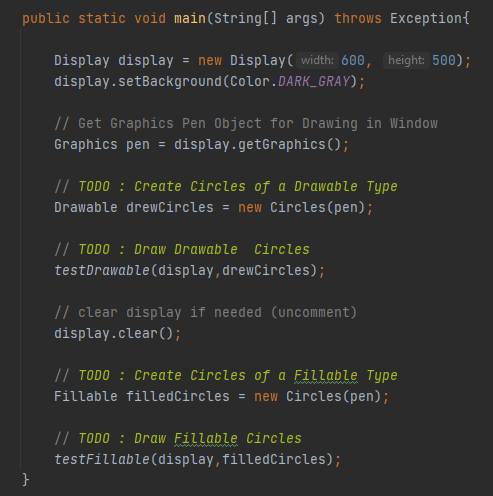
***testDrawable:***  create circles of a "Drawable" Type

****

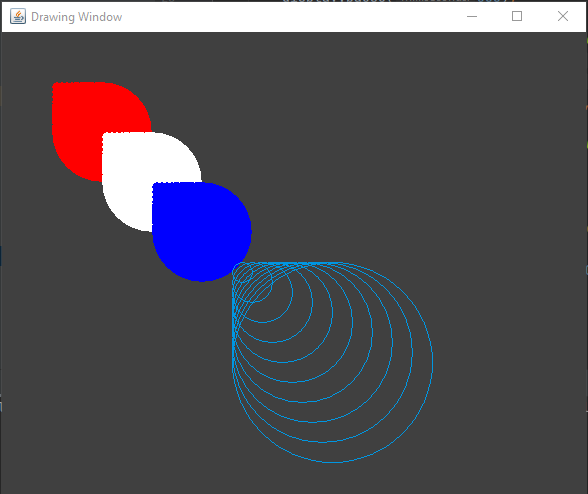
***testFillable:*** create circles of a "Fillable" Type



***Main:*** create objects and do the above tests



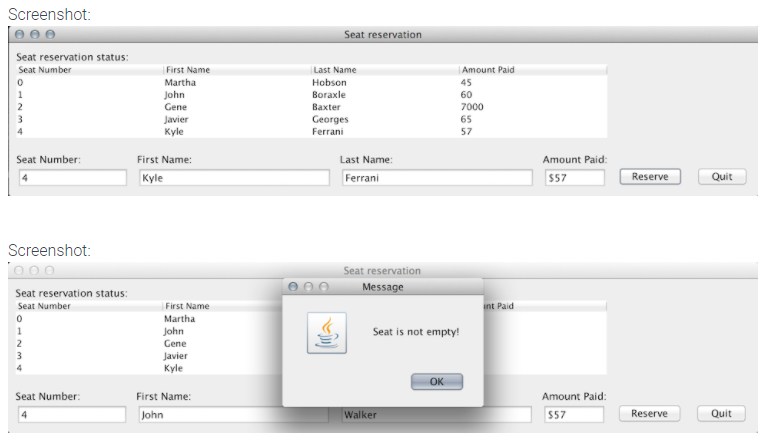
***OUTPUT:***



**ZYBOOKS:**

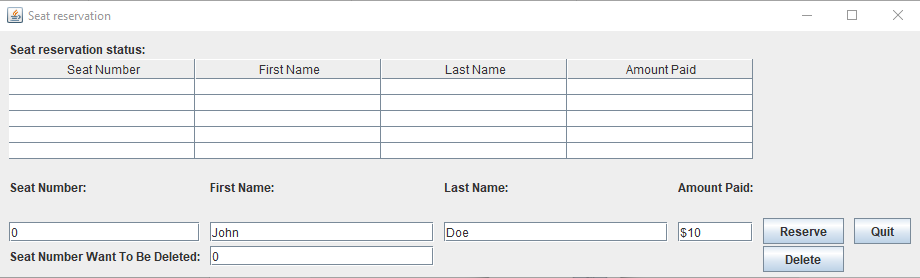
**SeatReservationFrame class:** create a program to manage the seat reservation. We can enter the seat number, customers’ names, and the amount paid. Then it will add this information to the table to save the data. Also, it will notice if the seat has already been added, or is not in the table.

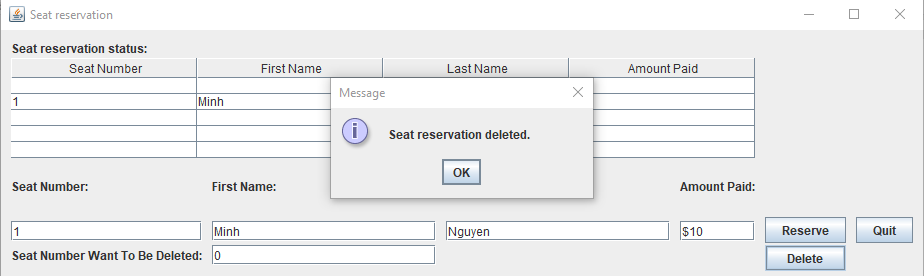
Below is the original program (on ZyBooks):



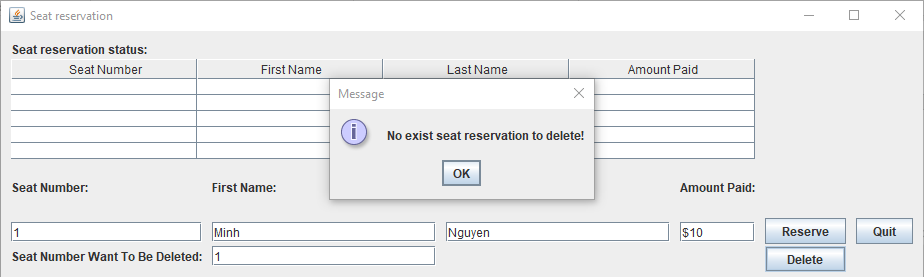
Modify the SeatReservationFrame program to have an additional JFormattedTextField and JButton component for the purposes of deleting a particular seat reservation. The JFormattedTextField component should allow the user to enter the seat number that should be deleted, and the JButton should trigger the deletion.

Below is the program after adding the “Delete” feature



After we enter the seat number want to be deleted, this seat’s data is removed

If we try to delete an empty seat, there is a popup like this



**Reading files with a GUI**

The GUI let us search and open the "temperatureLog.txt" file that is located in the data folder. It helps us to search through the disks, folders, and files. Additionally, it has some other features, such as create a new folder, go back (Up One Level) when we go to the wrong directory. Also, it can change the view of folders, like view as list, or details. And it also allows user to search for the file by its name or type.

Modify the "temperatureLog.txt" file in the "data" folder to add 3 more values (highlighted) you made up.

