**HCMC UNIVERSITY OF TECHNOLOGY AND EDUCATION**



**FACULTY OF INFORMATION TECHNOLOGY**

Logo

Description automatically generated

**FINAL TERM PROJECT**

**Topic of project:**

Design interactive whiteboards

**Lecturer name:**

**Mr. Nguyen Dang Quang**

|  |  |  |
| --- | --- | --- |
| **Student ID** | **Student name** | **Contribution (%)** |
| 22TM11031 | Nguyen Huu Thanh Phuc | 100% |
| 22TM11055 | Tran Viet Hung | 100% |

**Instructor's comments:**

**………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………**

**Thank you!**

To successfully complete this topic and report, the group would like to send sincere thanks to lecturer Nguyen Dang Quang who directly supported the group throughout the project process. The group thanked the teacher for providing timely advice and support to guide the group in accordance with the requirements of the chosen topic.

The project and report were carried out by the group in a fairly short period of time, with limited knowledge and many other technical limitations and experience in implementing an identification application. Therefore, in the process of creating the project, there are inevitable shortcomings, so the group is looking forward to receiving valuable comments from the teacher so that the group can improve their knowledge and be able to do more even better next time.

Finally, the group respectfully wishes the teacher good health and further success in his teaching career.

Once again, the team would like to sincerely thank you!

**Table of contents**

Contents

[**PART 1: PROJECT DESCRIPTIONS** 5](#_Toc167375328)

[**1.1** **Objectives** 5](#_Toc167375329)

[**1.2** **User benefits** 5](#_Toc167375330)

[**1.3** **Task assignment** 5](#_Toc167375331)

[**PART 2: REFERENCE SOFTWARE** 6](#_Toc167375332)

[**MIRO** 6](#_Toc167375333)

[**2.1 Introducing Application** 6](#_Toc167375334)

[**2.2 Advantage và disadvantage** 6](#_Toc167375335)

[**PART 3: DESIGN** 8](#_Toc167375336)

[**3.1.** **Formulating the idea** 8](#_Toc167375337)

[**3.2.** **Details in relation to algorithms** 8](#_Toc167375338)

[**3.2.1. JavaSwing** 8](#_Toc167375339)

[**3.2.2. Socket** 8](#_Toc167375340)

[**PART 4: IMPLEMENT** 9](#_Toc167375341)

[**4.1.** **Use case datagram** 9](#_Toc167375342)

[**4.2.** **Process description** 9](#_Toc167375343)

[**4.3.** **Class design** 13](#_Toc167375344)

[**4.3.1. Client** 13](#_Toc167375345)

[**4.3.2.** **Server** 15](#_Toc167375346)

[**4.4.** **Graphic User Interface** 16](#_Toc167375347)

[**PART 5: Conclusion** 17](#_Toc167375348)

[**5.1.** **Student evaluation** 17](#_Toc167375349)

[**REFERENCES** 18](#_Toc167375350)

# **PART 1: PROJECT DESCRIPTIONS**

* 1. **Objectives**

The drawing board provides users abilities to customize their drawings; we could find some applications such as Windows Paint, Artweaver, GIMP, etc. However, these programs are only available for one user at a time. Draw Together can allow multiple users to draw on board simultaneously through the network, which helps users spread their ideas to other people more comfortable

* 1. **User benefits**

Similarly to other drawing applications programs, Draw Together provides users with the ability to draw, move, and modify graphics objects. Essential drawing functions are:

Select any shape and stoke on the artboard and move them to another

Position

Change color: Users can change color with HSB, RGB, and Web format

Change the thickness of a stroke or border width of a shape

Pen: Draw strokes on canvas

Shape: Draw a rectangle and circle

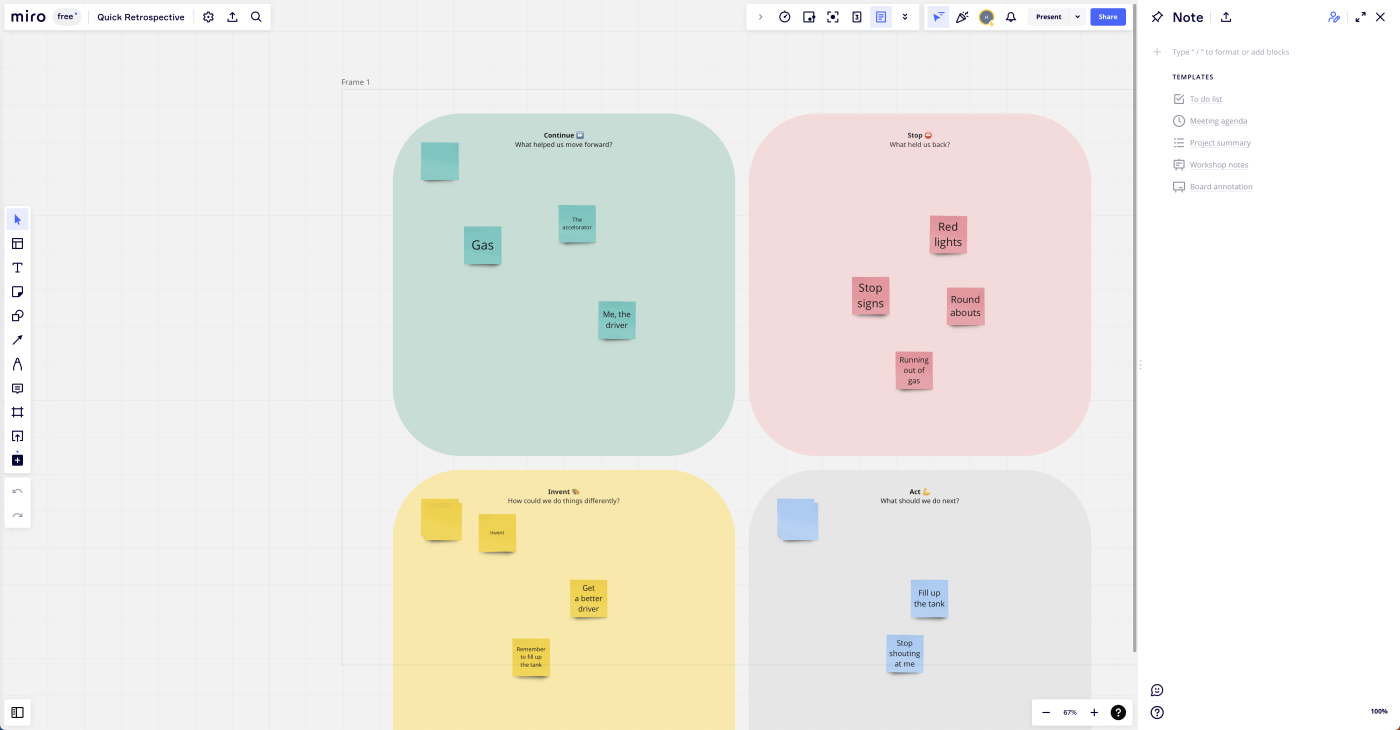
With the ability to connect to the Internet, the Draw Together application also allows users to invite one or more friends to draw together.

* 1. **Task assignment**

|  |  |  |
| --- | --- | --- |
| Student’s name | Evaluate contribution | Task works |
| Tran Viet Hung | 100% | Build the code program  Edit code  Write a report |
| Nguyen Huu Thanh Phuc | 100% | Design Interface  Edit code  Write a report |

**PART 2: REFERENCE SOFTWARE**

**MIRO**



## **2.1 Introducing Application**

Andrey Khusid (CEO) and Oleg Shardin (Board Member & former COO) founded Miro in 2011. Prior to Miro, Khusid, and Shardin founded an award-winning design agency called Vitamin Group in 2005. Vitamin Group offered clients web, product development and app design services.

## **2.2 Advantage và disadvantage**

**Miro** is built for more than just brainstorming. It also includes a set of features that supports managers who have to guide the process from ideation to implementation. With Miro, you can monitor what changes have been made, assign tasks, and carry on conversations with team members from within the interface.

If you want to present the results of your ideation, you'll love how easy it is to select and drag a portion of the board to create "frames," so you can pan through the board in a more organized manner during presentations.

However it is free for 3 editable tables, plans start at 10 USD per month for users

**PART 3: DESIGN**

* 1. **Formulating the idea**

After receiving Mr. Nguyen Dang Quang's project project on creating an interactive whiteboard application in Java language. This is a topic familiar to everyone's daily life, we can see this whiteboard through many applications, for example: zoom, meet... After a period of rescue and implementation, my team has created an application with the following functions:

* Create a pen that can write or draw freehand depending on the hover point
* Create available shapes such as squares, rectangles
* Create a color palette so that users can choose colors?
* Create a delete button

Overall, this is an easy-to-use application and has an intuitive interface. Make it easy for people to write down ideas after online meetings or conversations.

* 1. **Details in relation to algorithms**

**3.2.1. JavaSwing**

Swing is a Java Foundation Classes [JFC] library and an extension of the Abstract Window Toolkit [AWT]. Java Swing offers much-improved functionality over AWT, new components, expanded components features, and excellent event handling with drag-and-drop support

**3.2.2. Socket**

Socket.IO is an event-driven library for real-time web applications. It enables real-time, bi-directional communication between web clients and servers. It consists of two components: a client, and a server. Both components have a nearly identical API

# **PART 4: IMPLEMENT**

* 1. **Use case datagram**

Table 1 – Use case Shapes description

|  |  |
| --- | --- |
| Use case name | Shape |
| Description | Allows user to draw a rectangle or an ellipse on the board |
| Preconditions | Click one of the specific shape to draw that shape |
| Actor | User |
| Condition affecting  Termination outcomes |  |

Table 2 – Use case Color description

|  |  |
| --- | --- |
| Use case name | Color |
| Description | Allows user to choose the color of the shapes and lines |
| Preconditions | Click tho color button to choose the color |
| Actor | User |
| Condition affecting  Termination outcomes |  |

Table 3 – Use case Clear description

|  |  |
| --- | --- |
| Use case name | Clear |
| Description | Allows user to clear what were being drawn on the board |
| Preconditions | Click the Clear button to clear all shapes and lines |
| Actor | User |
| Condition affecting  Termination outcomes |  |

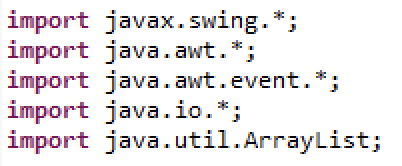
* 1. **Process description**

**4.2.1. Rendering**

Jpanel is a component that include all the buttons and an drawing area, which appears as a GUI of the application.

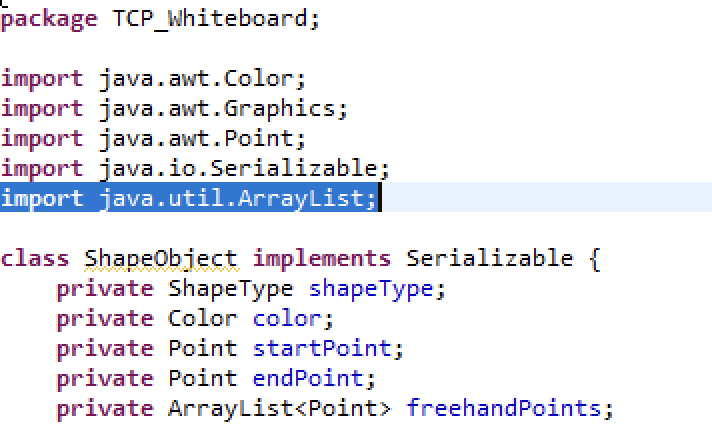


To implement the drawing actions, I created a class called DrawingCanvas. Then I imported the nessary libraries to this class.

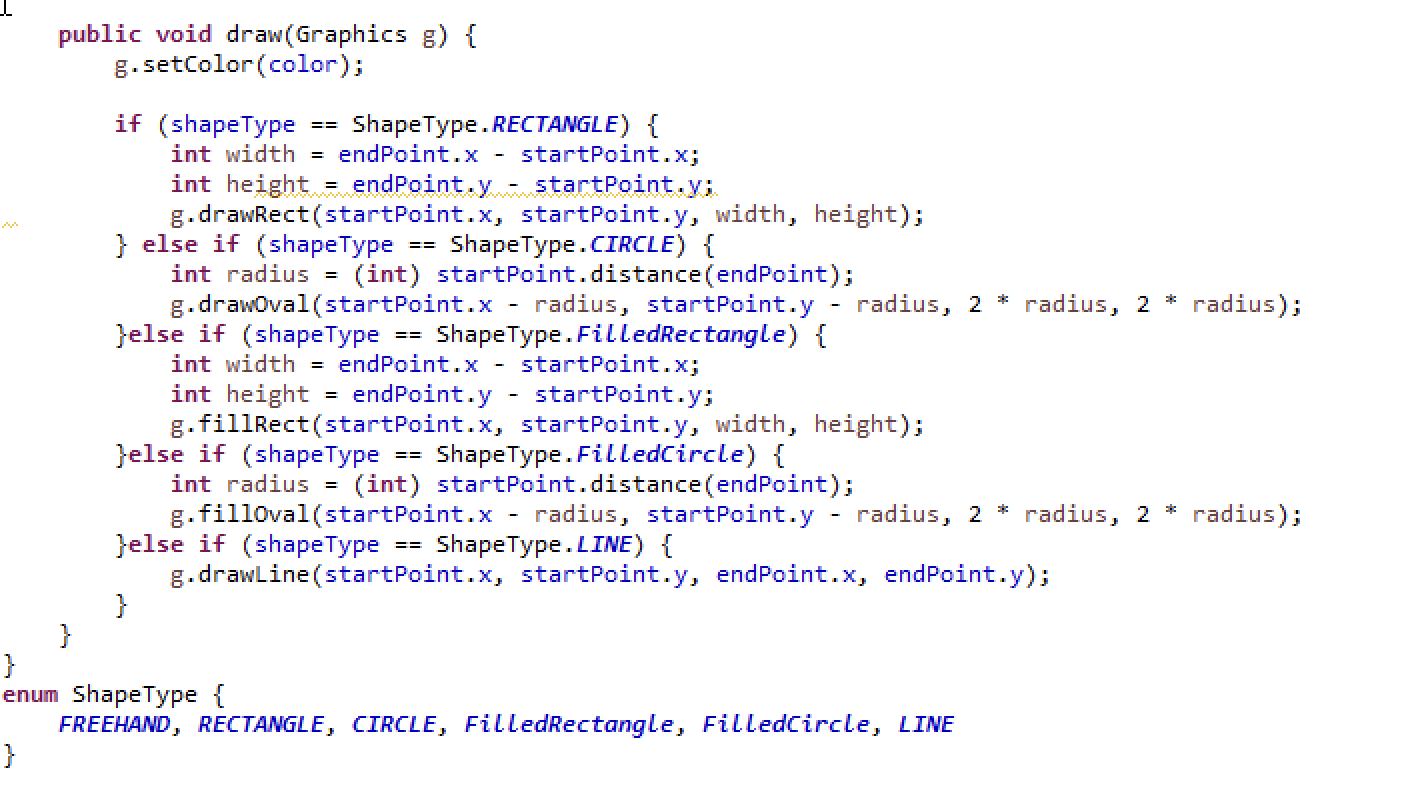


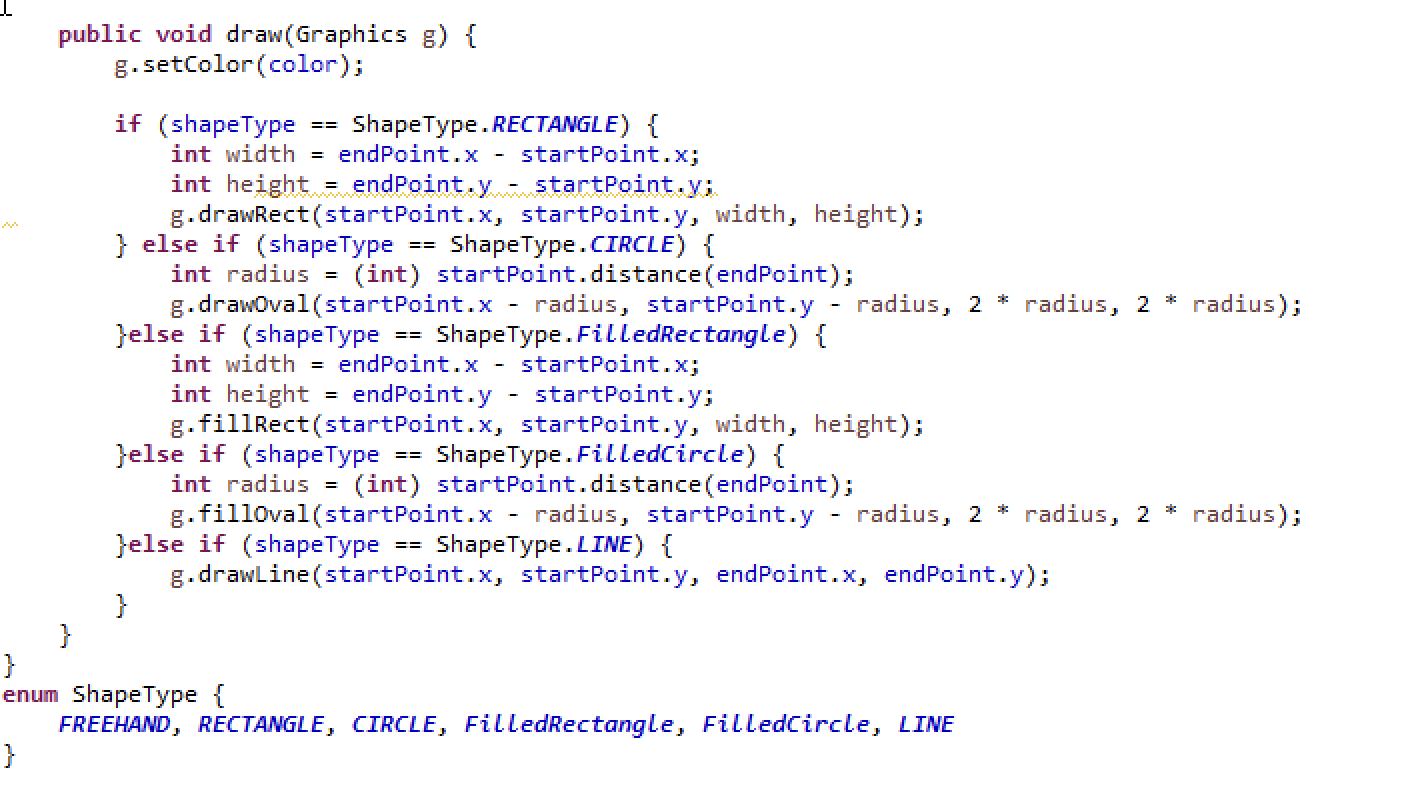
Each object is stored in an list. The list has one purpose is to store all shapes’ information from one client, then the list will be sent to the Server. Afterwards, the Server will send the list to other Clients to draw all shapes on their boards.

After having the GUI and a list to store the information, I created the objects by writing a new class called ShapeObject which implemented Serializable.

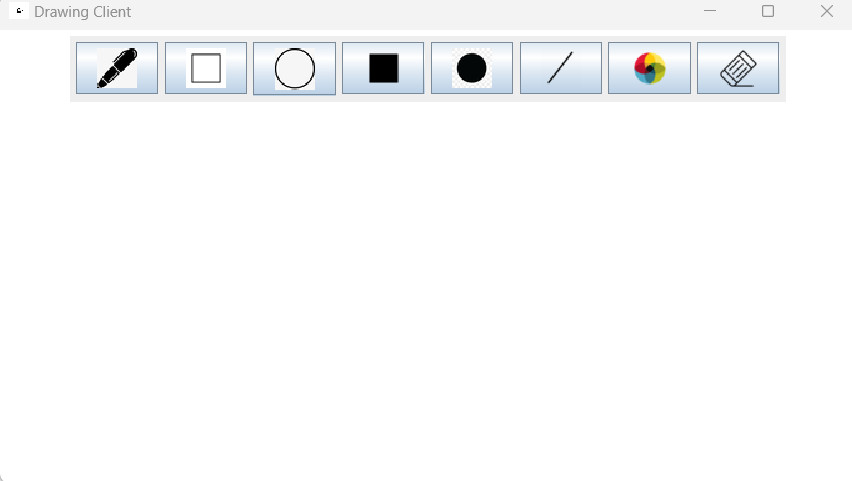


ShapeObject class defined all the information of the objects such as color, location and their type of shape. In terms of the type of shape, I create an “enum” in order to help the programe distinguish shapes easier.



FREEHAND, RECTANGLE, CIRCLE, FilledRectangle, FilledCircle, LINE

Now we can draw all the optional shapes to the canvas, and we have the result:

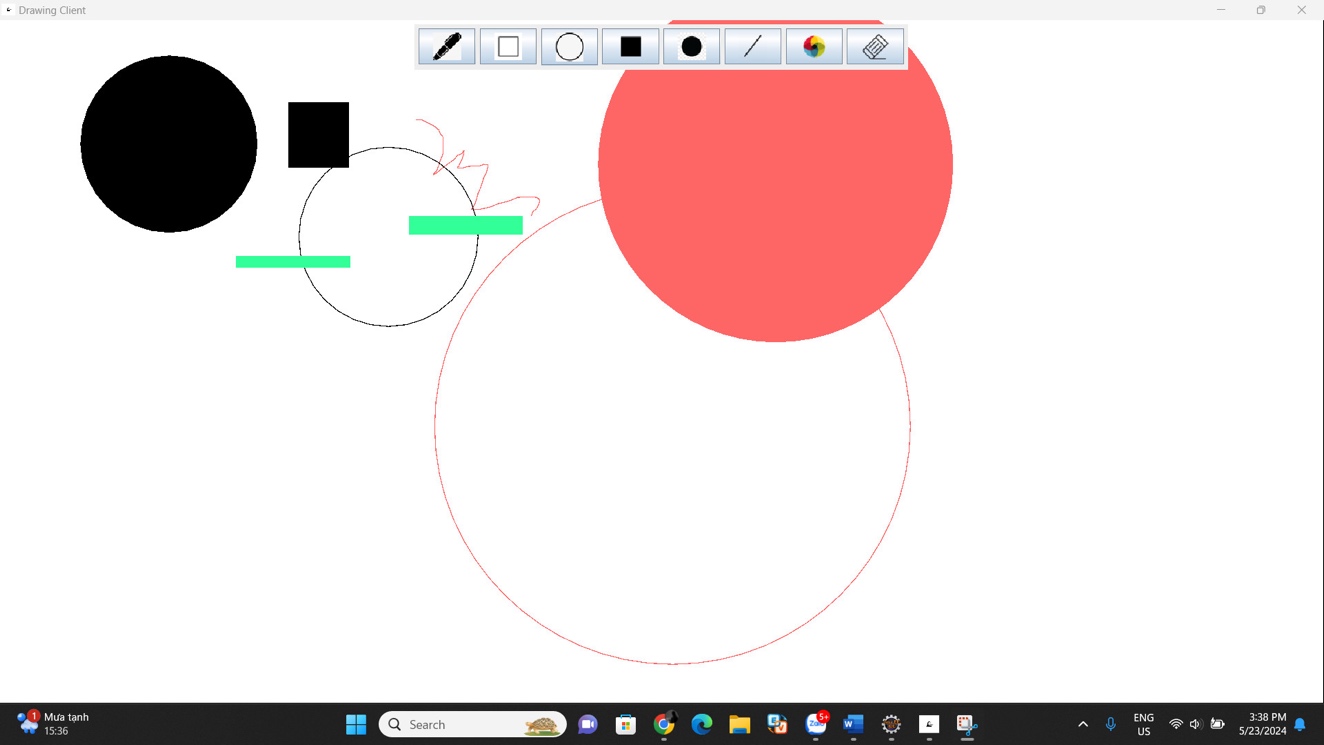


For Freeline drawing, I also create a arraylist(freehandLines) to store all the points of the freeline because the line is a sequence of points. The arraylist is included in a class called DrawingObject implementating Serializable.





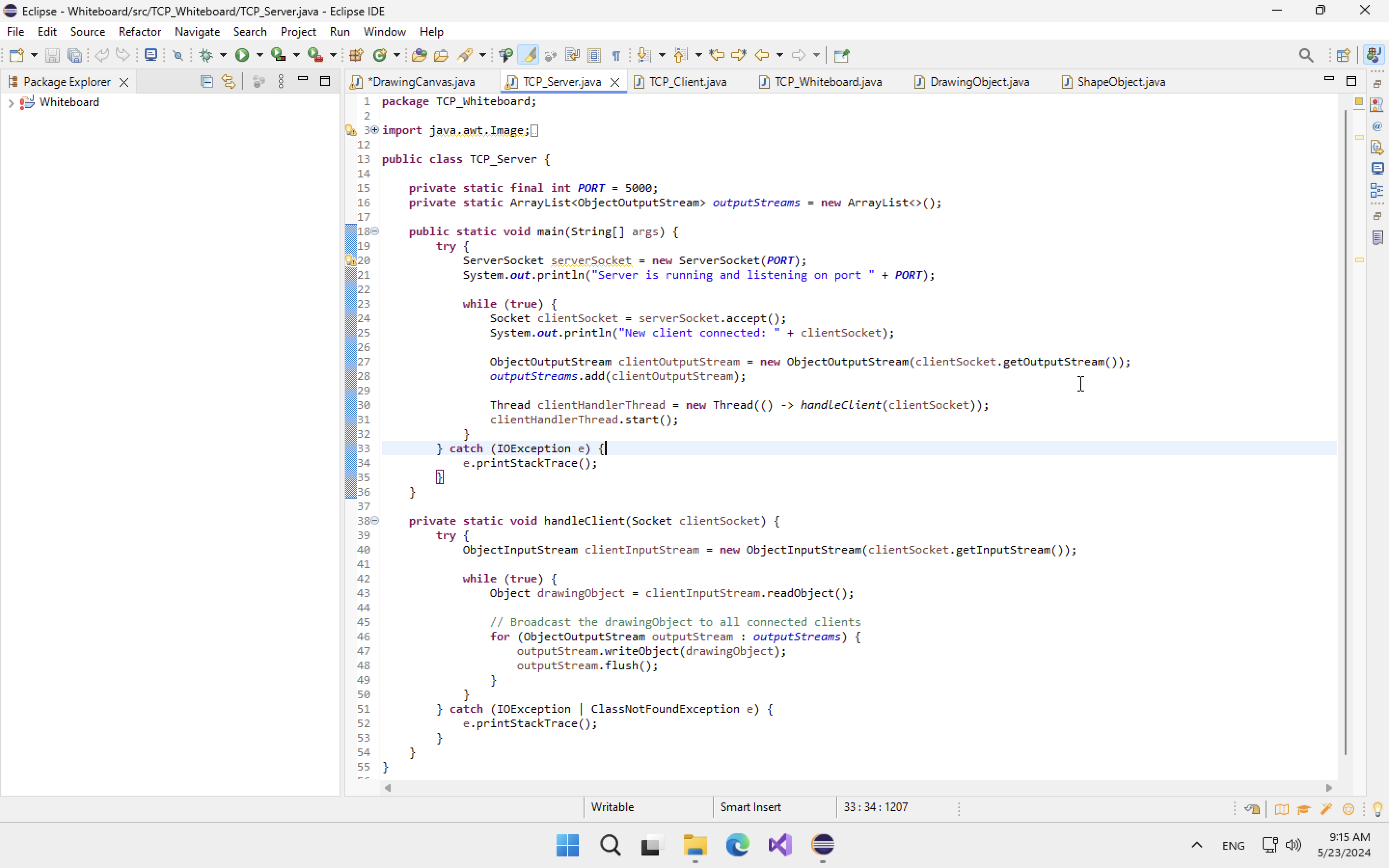
And the result is:



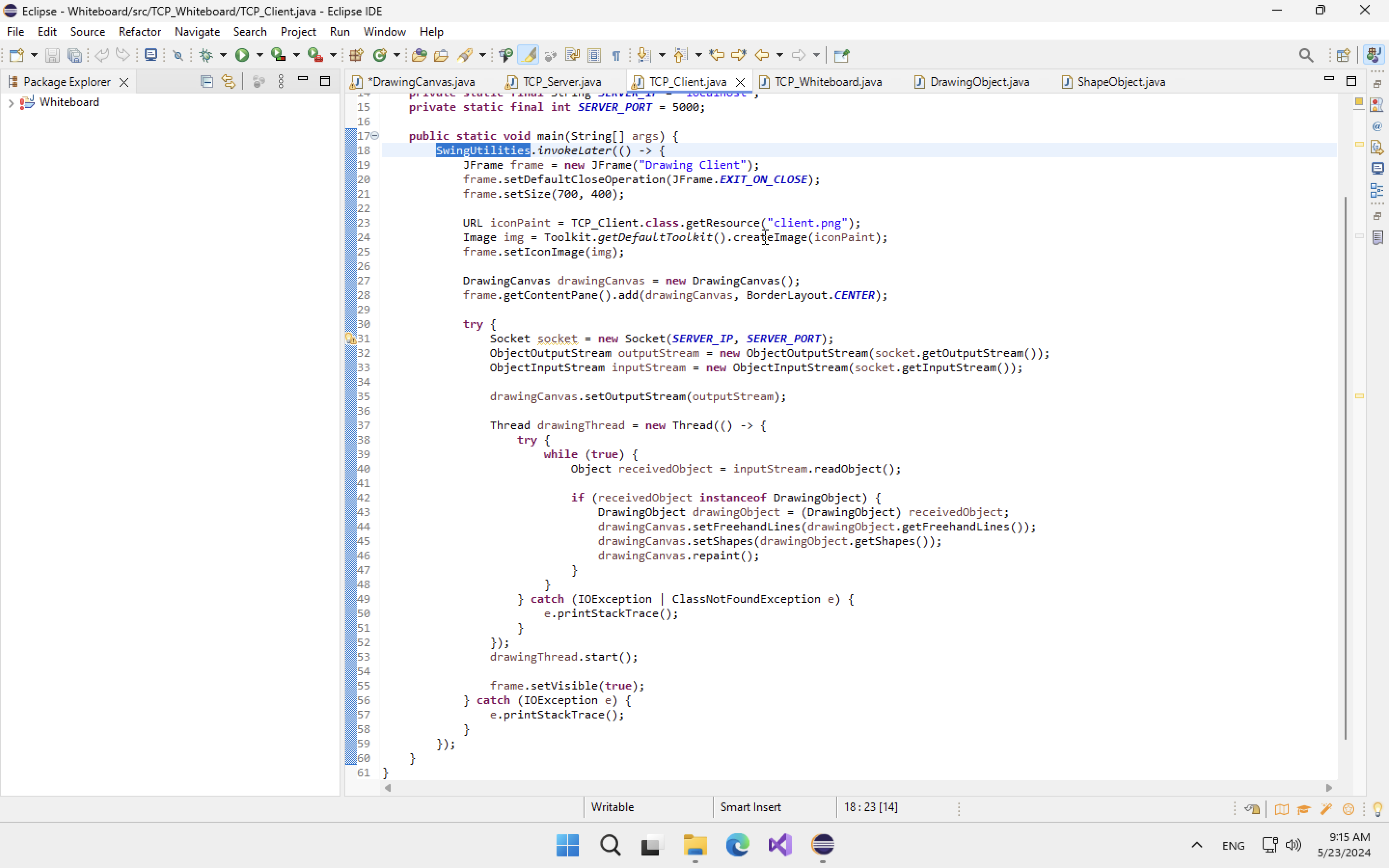
**4.2.2. Remote drawing**

To allow users to draw together, we need a server to be a middle man between users. Server receives graphics elements from a user and broadcast it to other users who are connecting to the same artboard.

In this project, I use basic Java Socket Programming technique to create a Server. Moreover, the Server also has an arraylist to handle a list of clients wwho want to connect.

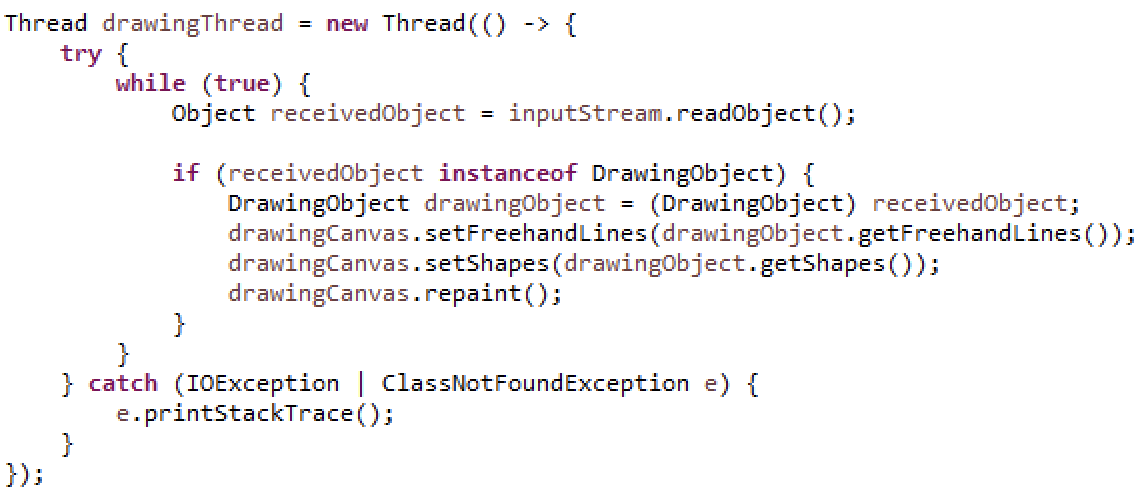


In the client-side, which is the Remote Draw application, I create a thread for reading data from the server and sending data to the server called drawingThread.



In the server-side, which is a console application, I create a thread called clientHandlerThread to handle request from a specific client.





* 1. **Class design**

In other to handling all features from drawing and remote drawing, object-oriented programming is the best technique helping us to implement those features. Below is the detail of all classes that I defined myself in this project, including the methods and the purposes of them.

### **4.3.1. Client**

Tabel 4-List of classes are used in the client applicatiion.



|  |  |  |
| --- | --- | --- |
| No. | Class Name | Purpose |
|  | ShapeObject  Implement: Serializable | Serialize shapes as objects and convert objects to byte stream |
|  | DrawingObject  Implements: Serializable | Serialize lists of objects and points which will be sent over a network |
|  | DrawingCanvas  Extends: JPanel | Create a GUI for user to interact with the application by using mouse |

|  |  |
| --- | --- |
| Enum | Purpose |
| ShapeType | Help defining and working more effective with a collection of shapes |

Tabel 5-List of methods are used in the ShapeObject class.

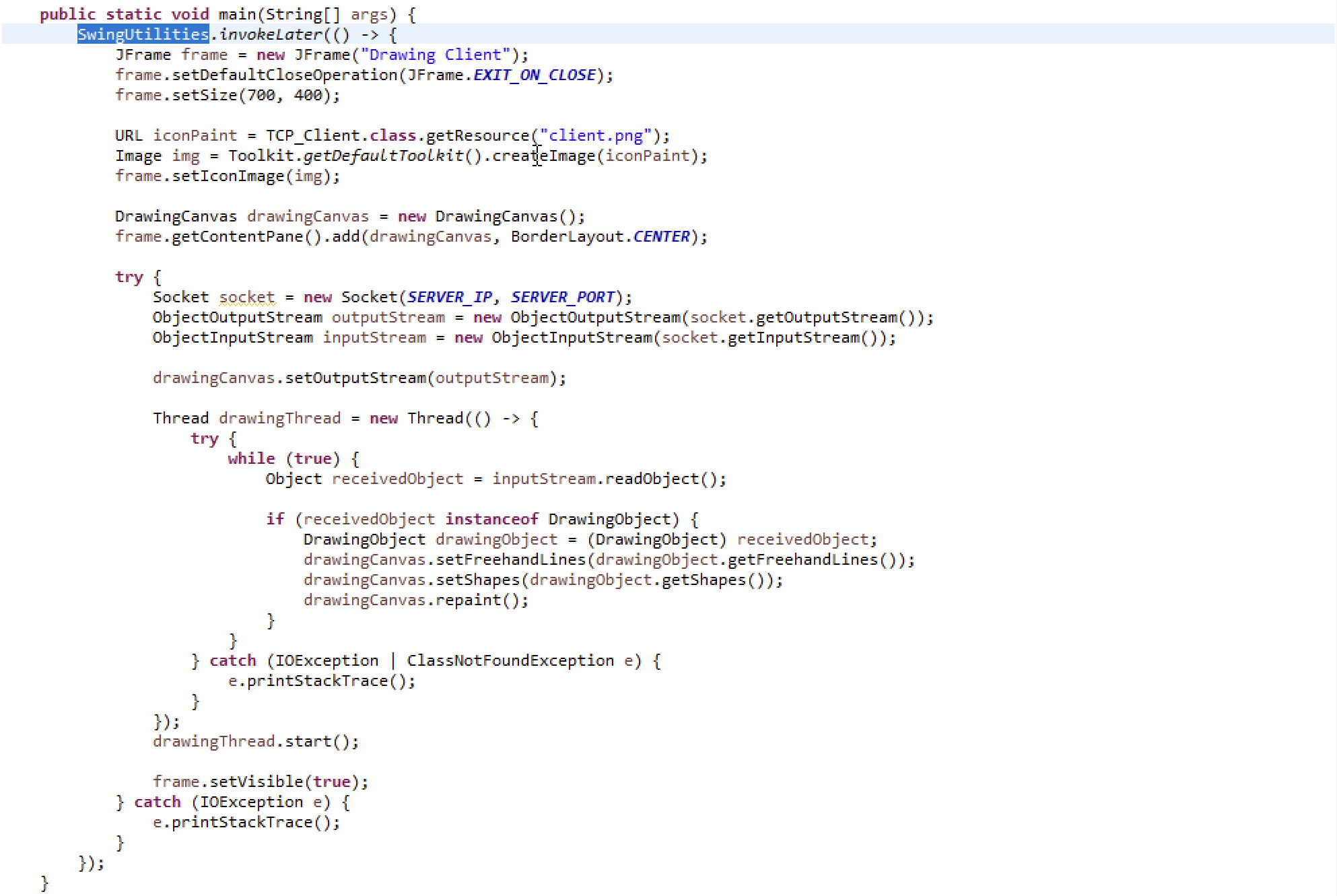
|  |  |  |  |
| --- | --- | --- | --- |
| No. | Method | Purpose | File Name |
|  | ShapeObject() | Play a role as a constructor | ShapeObject.java |
|  | Draw() | Draw the shape depend on the Enum | ShapeObject.java |

Tabel 6-List of methods are used in the DrawingObject class.

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Method | Purpose | File Name |
|  | DrawingObject() | Play a role as a constructor | DrawingObject.java |
|  | getFreehandLines() | Get the list of points | DrawingObject.java |
|  | getShapes() | Get the list of shapes | DrawingObject.java |
|  | Getcolor() | Get the information of the color | DrawingObject.java |

Tabel 7-List of methods are used in the DrawingCanvas class.

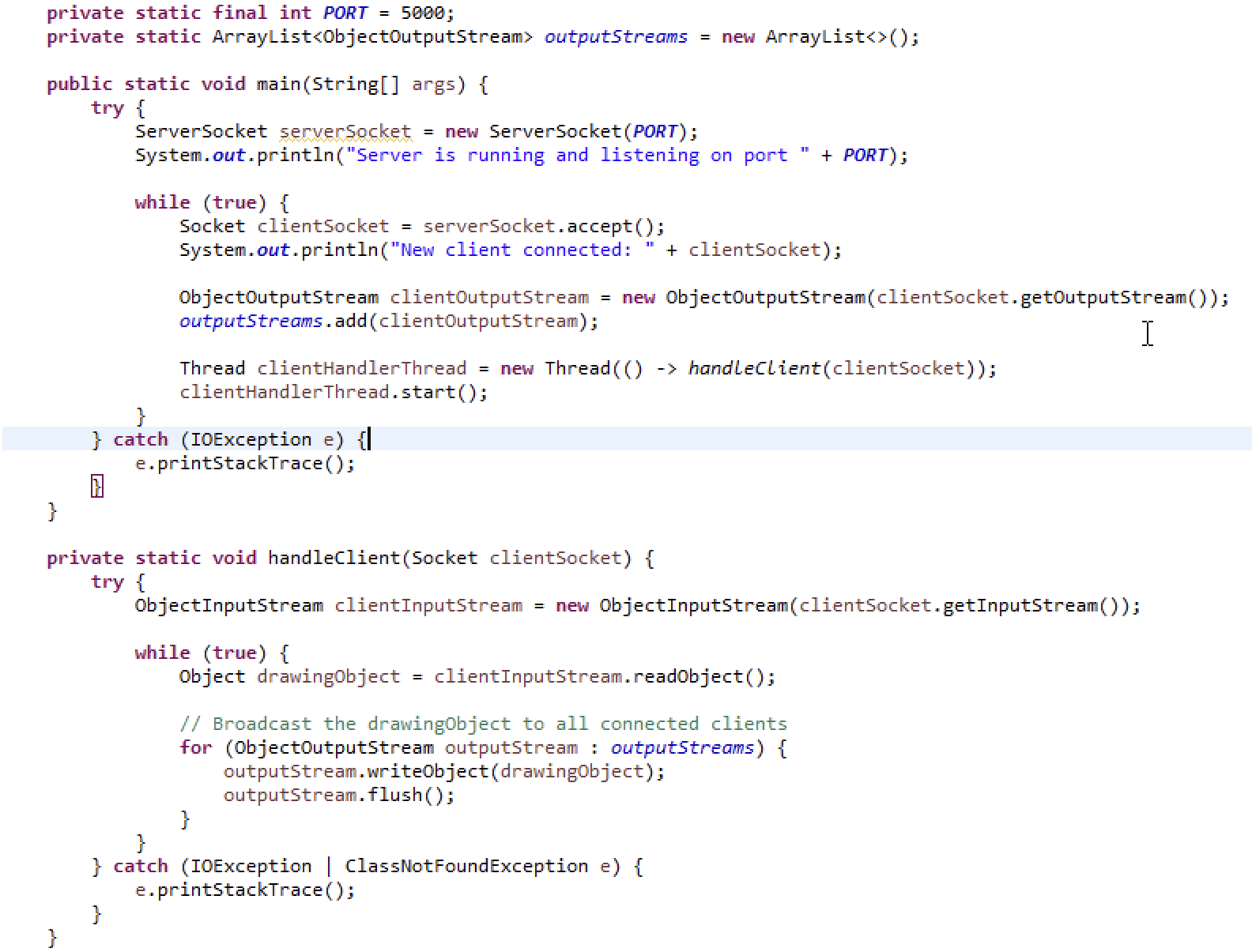
|  |  |  |  |
| --- | --- | --- | --- |
| No. | Method | Purpose | File Name |
|  | DrawingCanvas() | Play a role as a constructor | DrawingCanvas.java |
|  | mouseDragged() | Update continuously the location of points for freehand draw | DrawingCanvas.java |
|  | mousePressed() | Set the location of the start point when the mouse is pressed | DrawingCanvas.java |
|  | mouseReleased() | Set the location of the end point when the mouse is released | DrawingCanvas.java |
|  | setupButton() | Set up the GUI for users’ interaction | DrawingCanvas.java |
|  | setCurrentShapeType() | Assign the type of shape after user click a specific shape button | DrawingCanvas.java |
|  | addShape() | Add new shape to the Canvas | DrawingCanvas.java |
|  | setOutputStream() | Assign the output stream | DrawingCanvas.java |
|  | setCurrentColor() | Assign the shapes or lines’ color | DrawingCanvas.java |
|  | clearCanvas() | Clear the Canvas | DrawingCanvas.java |
|  | sendDrawingObject() | Send and broardcast the shapes, lines, color to the stream | DrawingCanvas.java |
|  | setFreehandLines() | Assign the freehand line | DrawingCanvas.java |
|  | setShapes() | Assign the shape | DrawingCanvas.java |
|  | paintComponent  (Graphics) | A method that help drawing the freehand line | DrawingCanvas.java |



* + 1. **Server**

Tabel 8-List of methods are used in Server class

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Method | Purpose | File Name |
|  | Main() | Create a Server and accept new client | TCP\_Server.java |
|  | handleClient() | Control the clients who connect to the server | TCP\_Server.java |



* 1. **Graphic User Interface**

Tabel 9-GUI of the Client

|  |  |  |
| --- | --- | --- |
| GUI | Purpose | Explaination |
|  | The main window of the application | The menu on the top of the window include a group of buttons that user can choose to draw a specific shape or line. |

# **PART 5: Conclusion**

* 1. **Student evaluation**
* Almost requirements are met.
* Design the application with Object-Oriented programming paradigm.
* The design of GUI is simple and easy to use.
* The application can allow multiple clients to draw together as long as they are connected the Server.
* The code is not so clean and reusable.
  1. **Difficulties**
* Java language is quite new to me so it takes a lot of time to be learnt. As a result, it slows down the project progress.
* Socket programming is a big problem because I was struggle in how to broadcast components among connected-clients.
  1. **Advantages**
* Simple GUI, easy to use.
* Allow multiple clients to connect and draw in real-time.
  1. **Disadvantages**
* Lost some basic function such as undo and redo.
* The application cannot import images to Canvas.
* The application cannot allow users to type text on Canvas.
  1. **Development ideas**
* The application can be upgraded to allow users chatting and downloading image to their computer.

# **REFERENCES**

Java Socket Programming

<https://www.javatpoint.com/socket-programming>

<https://www.baeldung.com/a-guide-to-java-sockets>

<https://www.youtube.com/watch?v=1jXYeRCI98&list=PLyxSzL3F7485hAzjQl7JrkcoaYmqoPn-5>

Java Language

<https://www.youtube.com/watch?v=xfOp0izFnu0&list=PLyxSzL3F748401hWFgJ8gKMnN6MM8QQ7F>