Drawing Editor

The drawing editor project is a GUI program where the user can use it to draw the most common drawing shapes using the mouse. The shapes supported are Circle, Rectangle, Square and Line. The class Shapes extends JFrame class, which displays a window with different components like labels and buttons that can be added to create GUIs. Shapes also implement the MouseListener and MouseMotionListener interfaces that handle the events when the mouse is in/out of motion.

The MouseListener interface has 5 abstract methods:

* void mousePressed(MouseEvent e): when mouse key is pressed.
* void mouseReleased(MouseEvent e): when mouse key is released.
* void mouseClicked(MouseEvent e): when mouse key is pressed/released.
* void mouseExited(MouseEvent e): when mouse exited the component.
* void mouseEntered(MouseEvent e): when mouse entered the component.

The MouseMotionListener has 2 abstract methods:

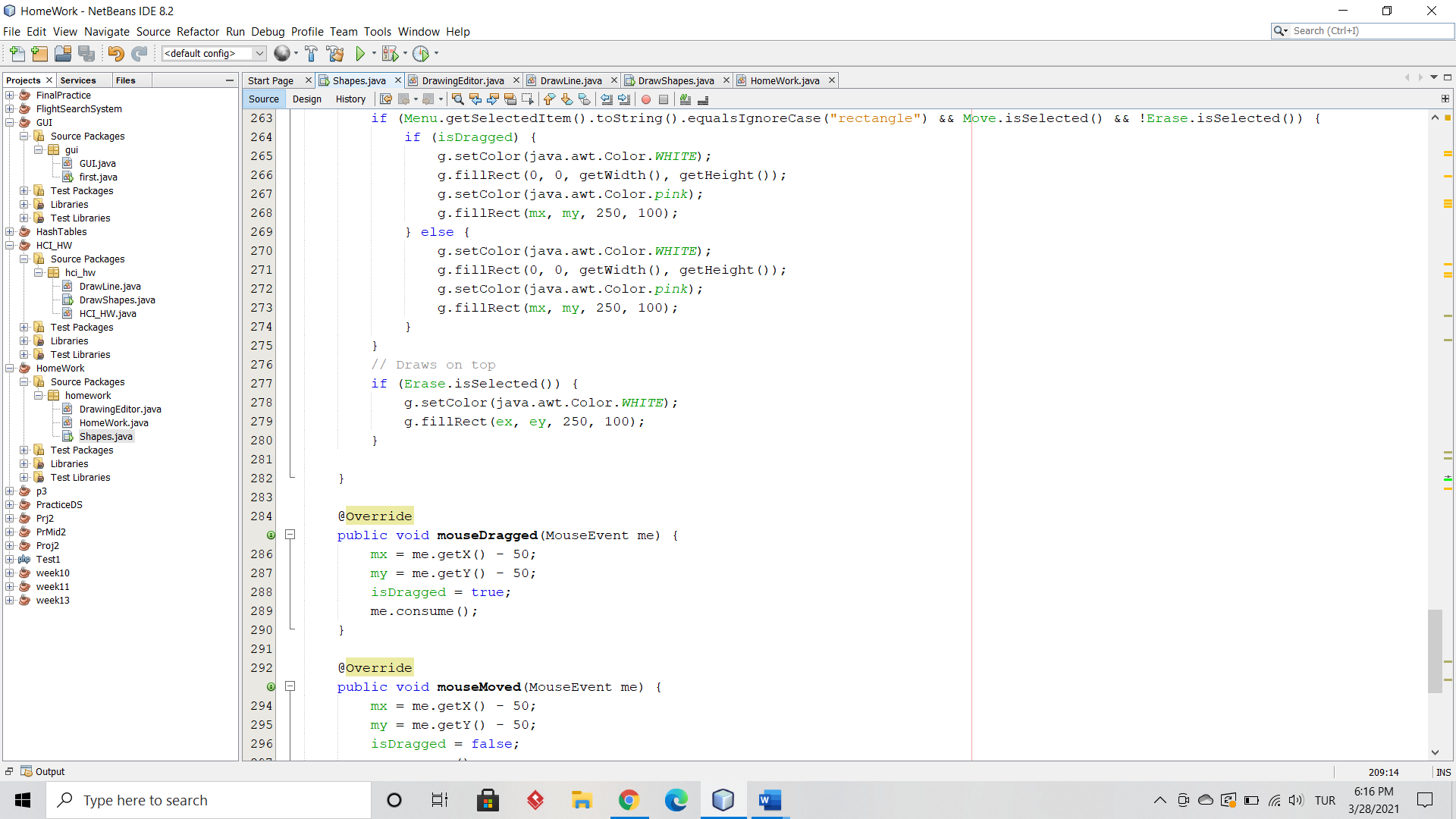
* void mouseDragged(MouseEvent e): used when the mouse button is

pressed in the component and dragged. Events are passed until the user

releases the mouse button.

* void mouseMoved(MouseEvent e): used when the mouse cursor is moved

from one point to another.



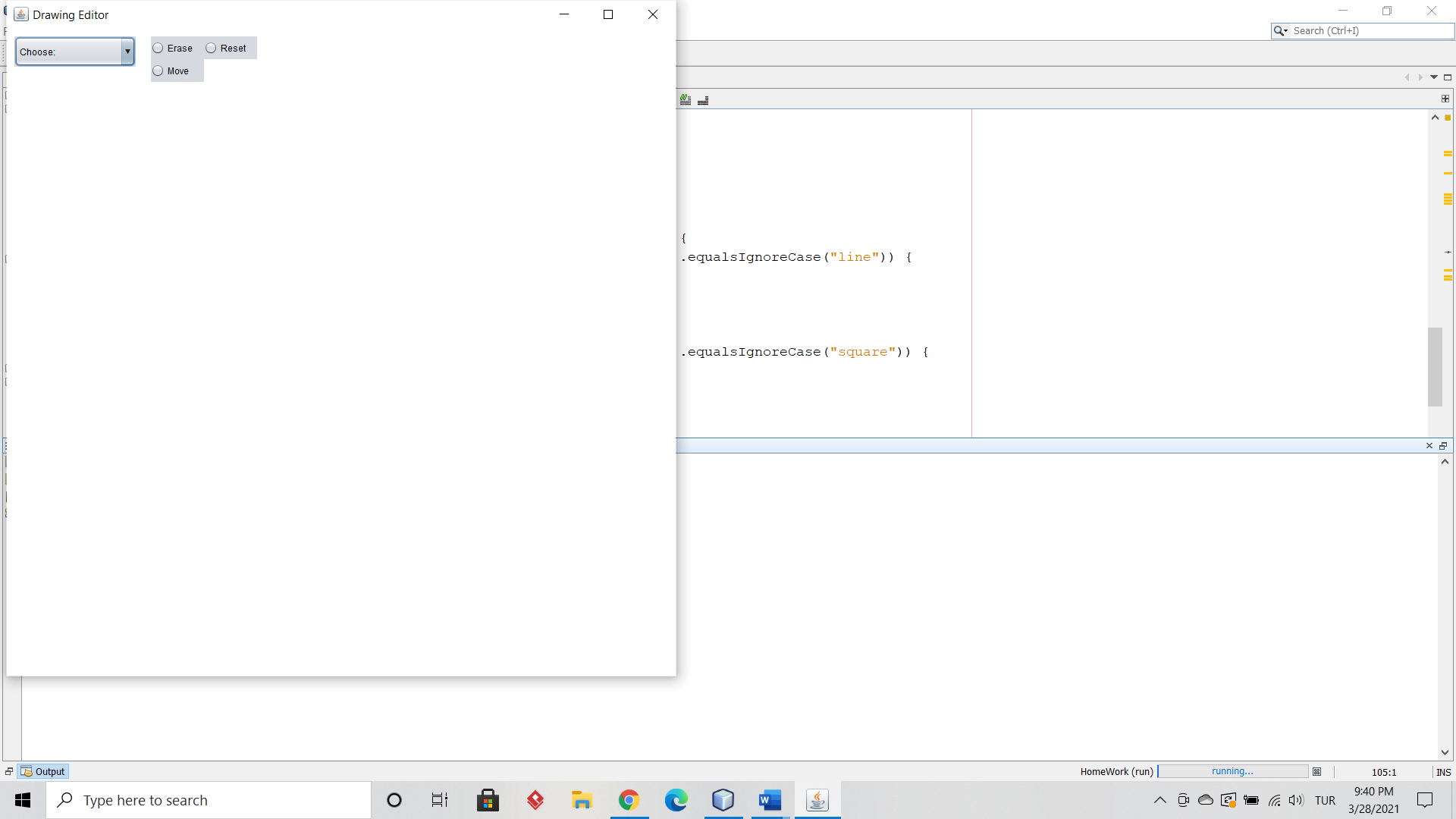
All abstract methods are implemented.

The Source contains the code and the Design contains a preview to the GUI design (Figure1). A combo box and three radio buttons were used for this program.

Figure 1

When you run the program the “Drawing Editor” window appears, and by default the Choose drop-down menu box, Erase component ,Move component and Reset component as shown below in Figure 2. Incase the components didn’t appear move the mouse across the top part of the window and they should appear.

The window’s size is set to 900x900 and cannot be resized. The window’s background is set to white. When you click on the Choose option, the drop-down menu appears. The **addMouseListener(MouseListener ml)** and **addMotionListener(MouseMotionListener ml)** methods are invoked in the constructor to recieve the mouse events and mouse motion events from every component.



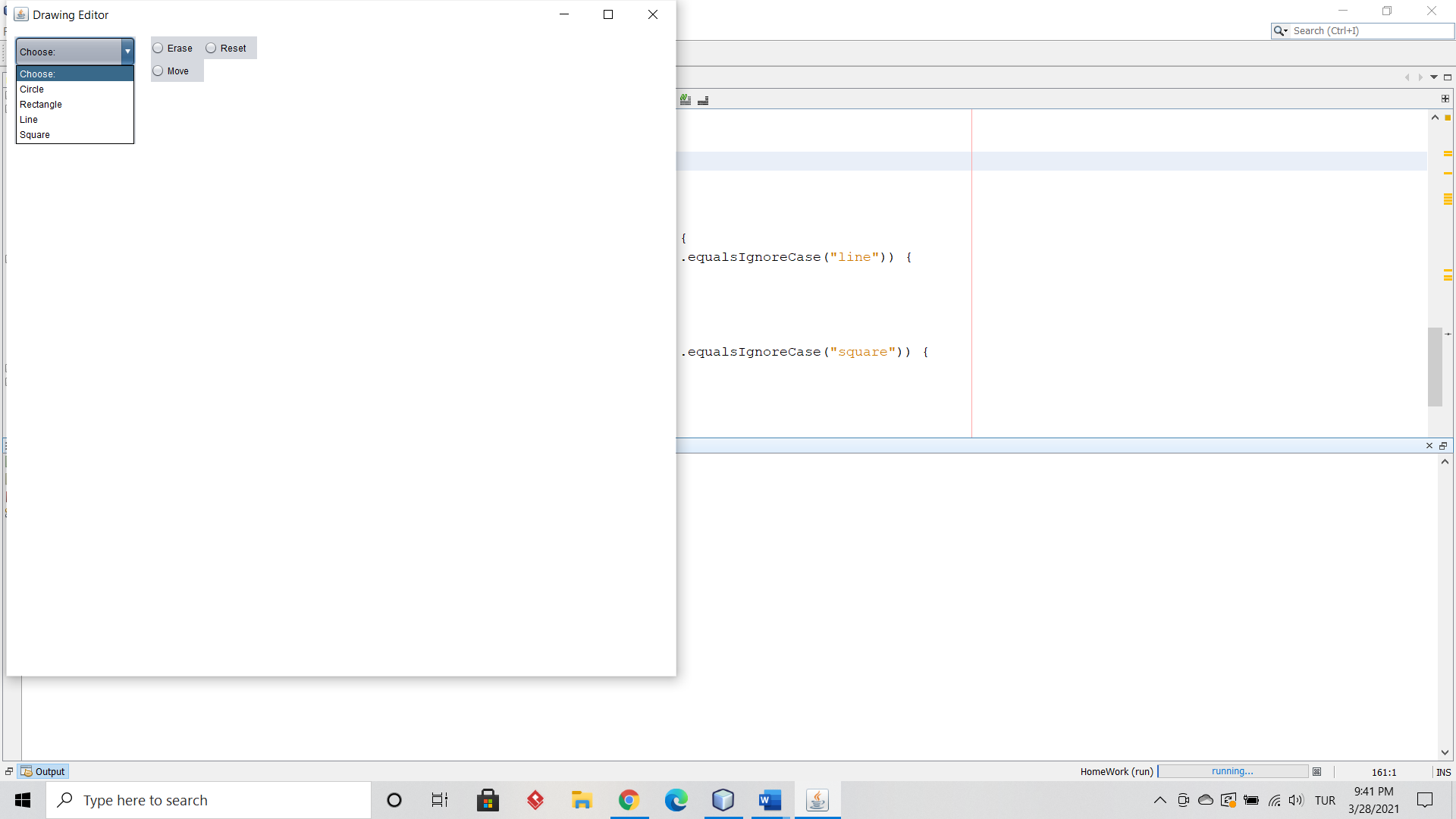


Figure 2a)

Figure 2b)

To draw, choose the shape type from the Menu box and click anywhere on the window. The sizes for all shapes are fixed except for the line. Inorder to draw a line click, drag and release anywhere on the window. The program checks which option was selected from the dropdown box using the **getSelectedItem()** (which is a built in method in the component Menu) and get the horizontal (p1) and vertical (p2) positions of the released mouse key event from the **mouseReleased()** method. The shape is drawn at that position. For the Line, the program stores the inital and final points of the line in an array (int[] points) , the inital point is taken from the **mousePressed()** method and the final point is taken from the **mouseReleased()** method. These shapes were created using the overridden **paint()** method from the Graphics class. This method uses other built in methods in the Graphics class. The **drawLine()** method is invoked for drawing a line, the **fillOval()** method is invoked for drawing a circle, and the **fillRec()** method is invoked for drawing a rectangle and square with different sizes. The color of the shapes were set to pink. The **repaint()** method is called after every mouse event to create a new shape.

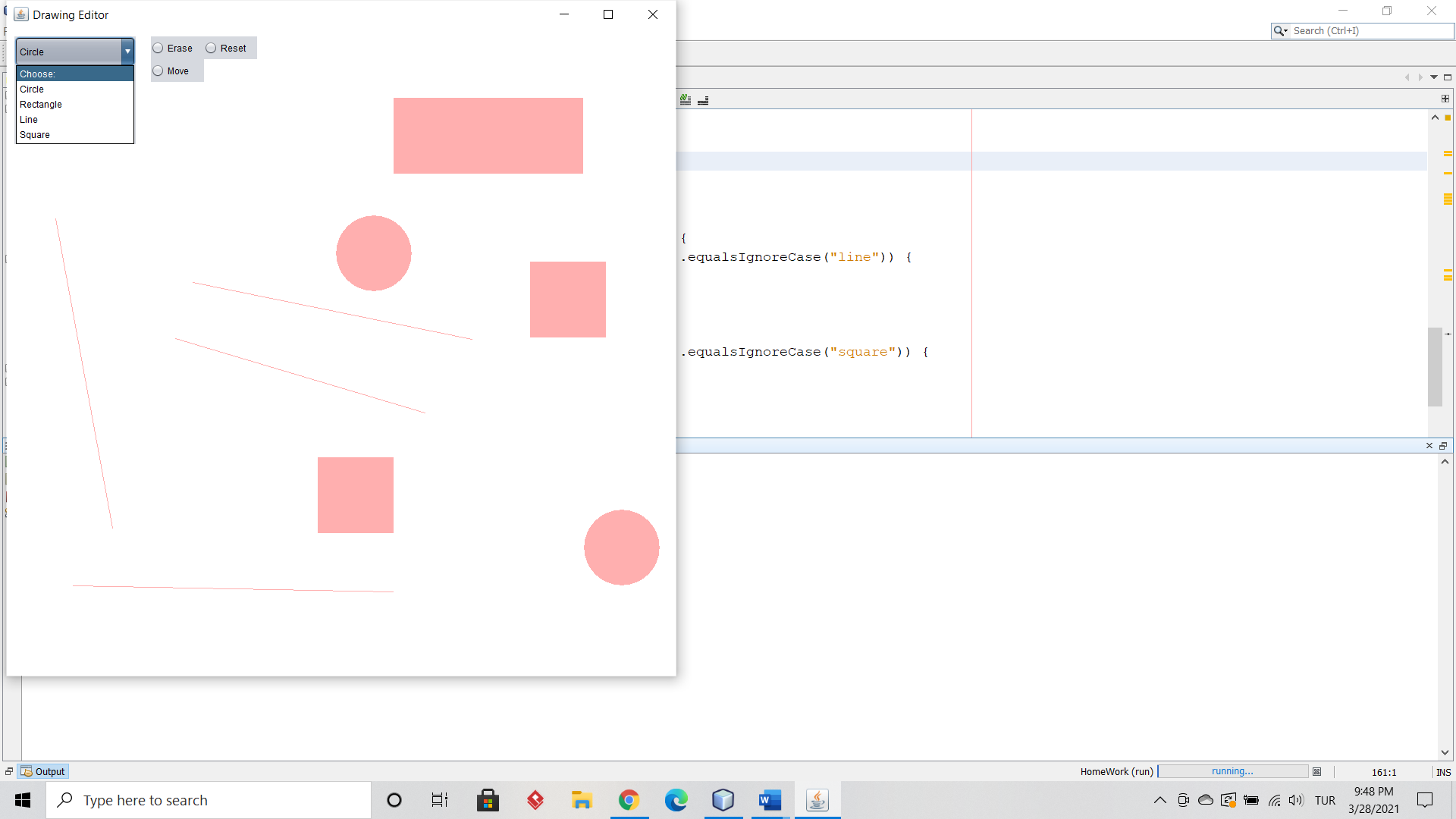


Figure 3

To move, choose the shape type from the menu box and select the Move option. The window resets and by clicking anywhere on the screen, the object will move. You can also drag the object anywhere around the screen. You can only move one object at a time. You cannot move a line. The move part is implemented by the **mouseDragged()** and **mouseMoved()** methods. The new horizontal (mx) and vertical (my) positions are taken and a new shape is drawn at the position. The boolean isDragged variable is used to check if the shape is being dragged (isDragged = true) or moved (isDragged = false). The program checks if the move option is selected by the component’s **isSelected()** built in method. Make sure you un-select the Move option when you’re done.

To Erase, choose the shape type from the menu box and select the Erase option. Click on the shape you want to erase, the program takes the horizontal (ex) and vertical (ey) positions of the shape from the **mouseClicked()** method and draws on top of it with white by the **paint()** method. The taken positions may not be accurate so many attempts may be needed to erase a shape. The program checks if the erase option is selected by the component’s **isSelected()** built in method . Make sure you un-select the Erase option when you’re done.

To clear the window, select the Reset option. This deletes all the drawing by painting the background to white using the **paint()** method. Make sure you un-select the Reset option when you’re done. Deletion using the Graphics class cannot be explicity done, however creating a new window on top on the previous one does the job. Make sure you un-select the Reset option when you’re done.

The program stops by clicking on the close button of the window, since JFrame has the option to hide or close the window with the help of **setDefaultCloseOperation(int x)** method. JFrame.EXIT\_ON\_CLOSE is passed as a parameter.