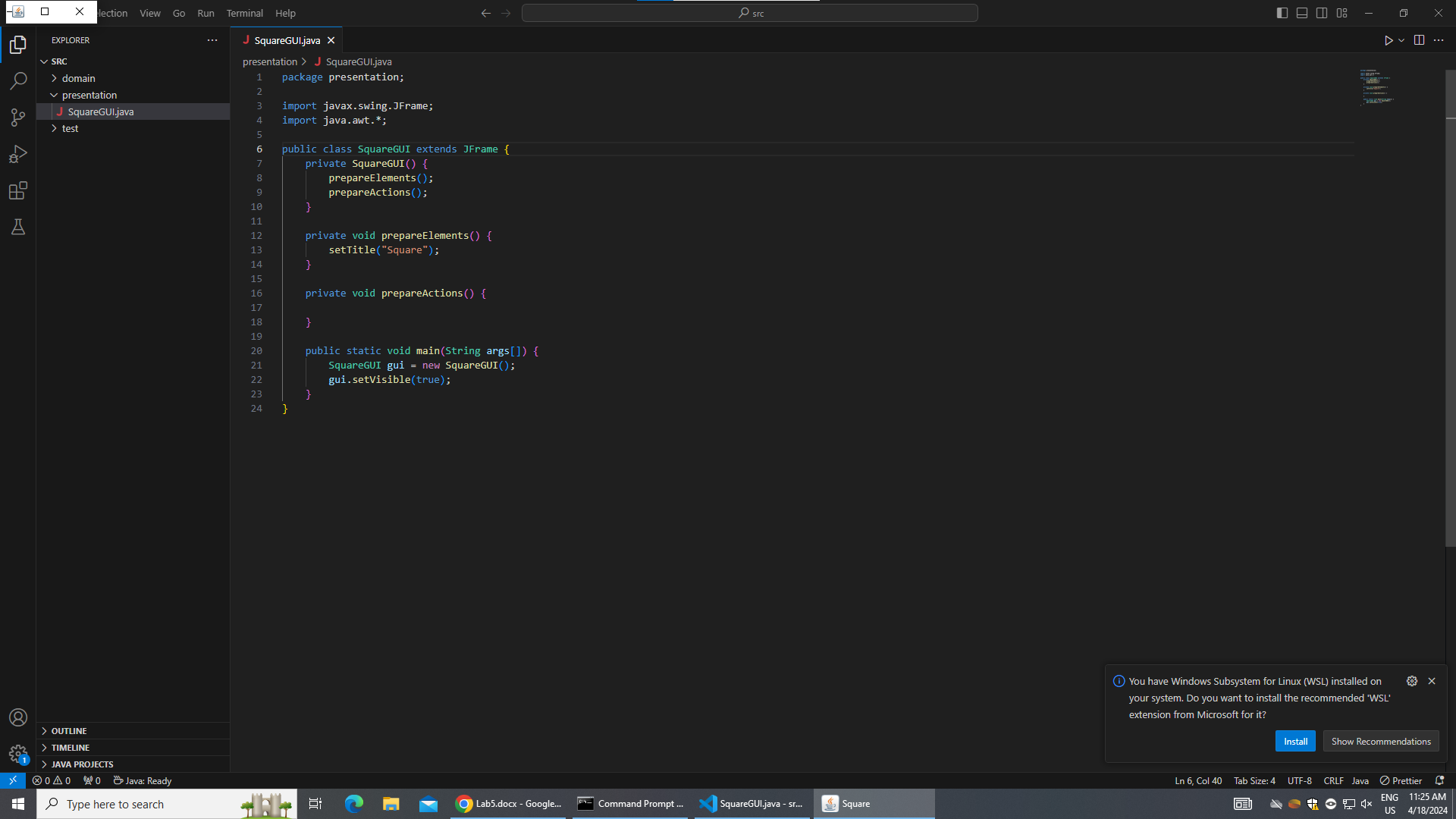
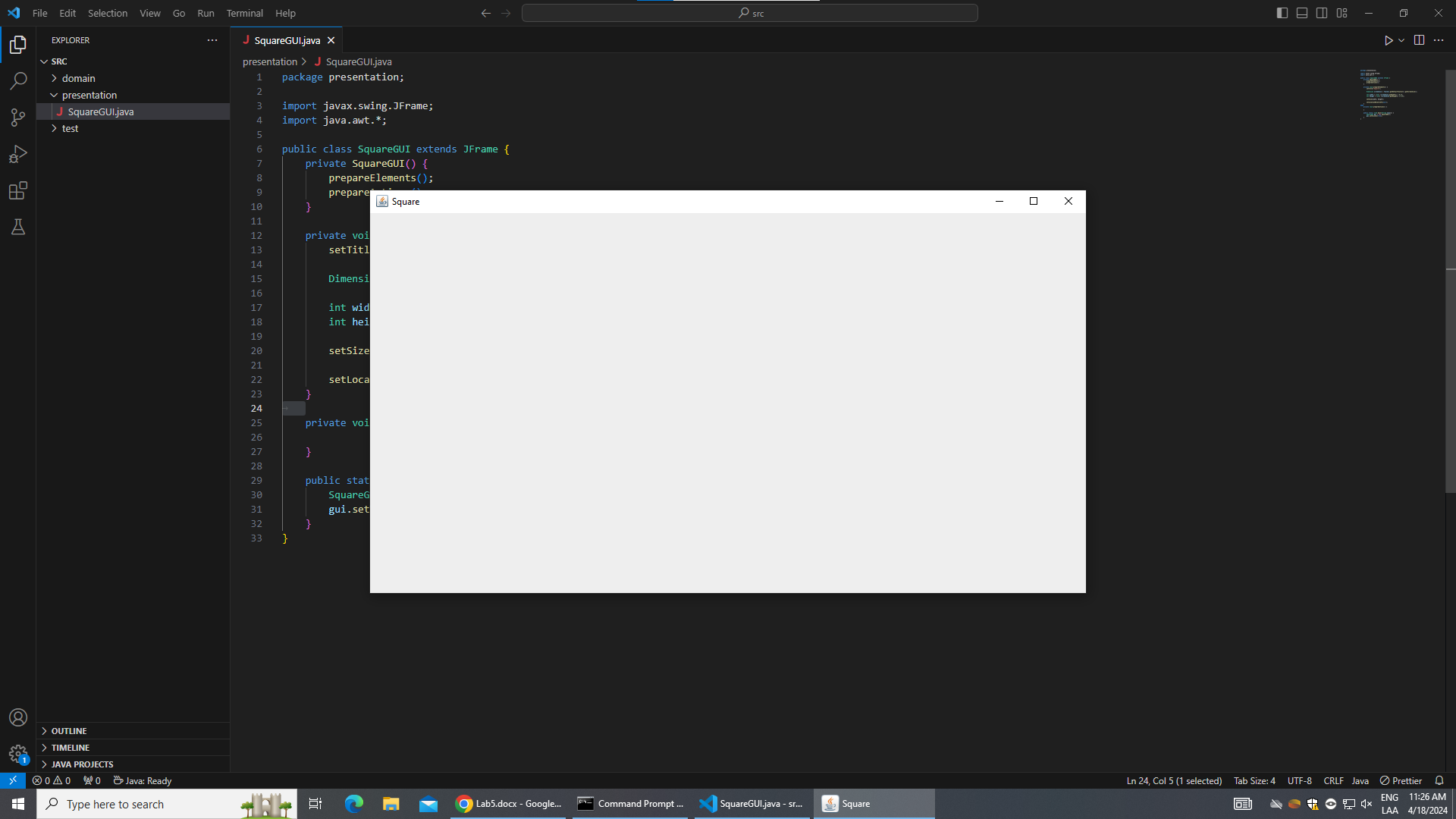
**Ciclo 0: Ventana vacía – Salir [En \*.java y lab05.doc]**

El objetivo es implementar la ventana principal de Square con un final adecuado desde el icono de cerrar. Utilizar el esquema de prepareElements-prepareActions.

1. Construyan el primer esquema de la ventana de Square únicamente con el título “Square”. Para esto cree la clase SquareGUI como un JFrame con su creador (que sólo coloca el título) y el método main que crea un objeto SquareGUI y lo hace visible. Ejecútenlo. Capturen la pantalla. (Si la ventana principal no es la inicial en su diseño, después deberán mover el main al componente visual correspondiente)



2. Modifiquen el tamaño de la ventana para que ocupe un cuarto de la pantalla y ubíquenla en el centro. Para eso inicien la codificación del método prepareElements. Capturen esa pantalla.



3. Traten de cerrar la ventana. ¿Termina la ejecución? ¿Qué deben hacer en consola para terminar la ejecución?

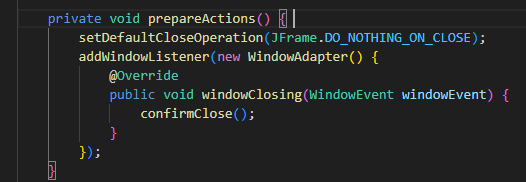
Si damos click en el botón de x, en la parte superior derecha, la ventana si se cierra, sin embargo, el programa sigue ejecutando, pues no se ha terminado de correr el método main(). El JVM puede lanzar múltiples procesos de concurrencia ejecutándolo en diferentes hilos.

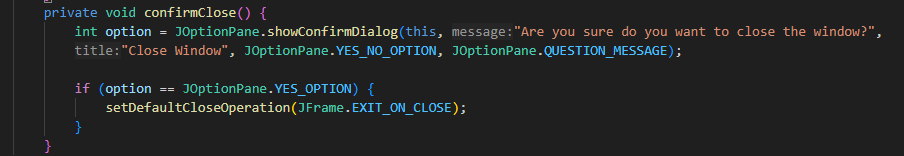
4. Estudien en JFrame el método setDefaultCloseOperation. ¿Para qué sirve? ¿Cómo lo usarían si queremos confirmar el cierre de la aplicación? ¿Cómo lo usarían si queremos simplemente cerrar la aplicación?

El setDefaultCloseOperation se utiliza para establecer el comportamiento de cierre de una ventana JFrame en una aplicación java swing. Como parametros tenemos:

* JFrame.DO\_NOTHING\_ON\_CLOSE: La ventana no se cierra cuando el usuario lo intenta.
* JFrame.HIDE\_ON\_CLOSE: La ventana se oculta, pero la aplicación se sigue ejecutando.
* JFrame.DISPOSE\_ON\_CLOSE: La ventana se cierra liberando todos los recursos relacionados con ella, pero la aplicación se sigue ejecutando
* JFrame\_CLOSE\_ON\_CLOSE: La aplicación se cierra por completo cuando el usuario cierra la ventana principal.

5. Preparen el “oyente” correspondiente al icono cerrar que le pida al usuario que confirme su selección. Para eso inicien la codificación del método prepareActions y el método asociado a la acción (exit). Ejecuten el programa y cierren el programa. Capturen las pantallas.





**Ciclo 1: Ventana con menú – Salir [En \*.java y lab05.doc]**

El objetivo es implementar un menú clásico para la aplicación con un final adecuado desde la opción del menú para salir. El menú debe ofrecer mínimo las siguientes opciones :Nuevo, Abrir – Salvar y Salir . Incluyan los separadores de opciones.

1. Expliquen los componentes visuales necesarios para este menú. ¿Cuáles serían atributos y cuáles podrían ser varibles del método prepareElements? Justifique.

Debería tener un objeto de tipo JMenuBar (que es donde se almacenan las opciones de la barra superior de la interfaz). Además, debería tener un elemento de tipo JMenu, que despliegue las opciones que se especifican. Por último, debería tener los ítems que se van a ejecutar dependiendo de la opción (Nuevo, Abrir – Salvar y Salir).

Los atributos son:

private JMenuBar menuBar;

private JMenu fileMenu;

private JMenuItem newItem;

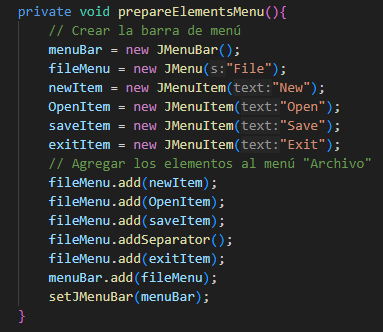
private JMenuItem OpenItem;

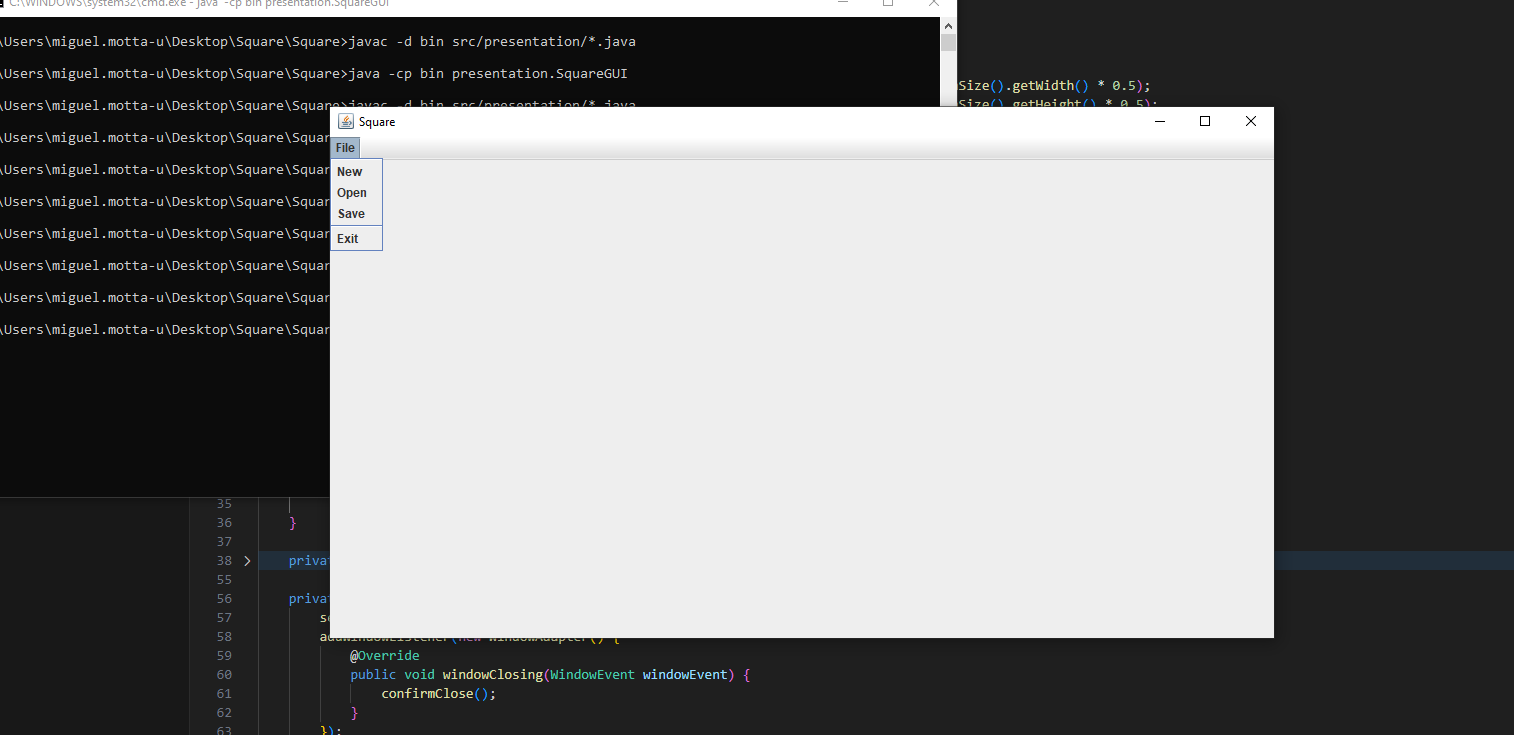
private JMenuItem saveItem;

private JMenuItem exitItem;

Y en el método, no tendrá variables de momento puesto que el método prepareElementsMenu() construye el menu por sí solo.

2. Construya la forma del menú propuesto (prepareElements - prepareElementsMenu) . Ejecuten. Capturen la pantalla.





3. Preparen el “oyente” correspondiente al icono cerrar con confirmación (prepareActions - prepareActionsMenu). Ejecuten el programa y salgan del programa. Capturen las pantallas.