**SOBRE LAS LIBRERIAS:**

Este código Java implementa una interfaz gráfica para un juego de ajedrez utilizando el framework Swing. A continuación, explicaré brevemente qué hace cada una de las librerías utilizadas en este código:

1. **java.awt:** Proporciona clases para la creación de interfaces de usuario y gráficos en Java. En este código, se utiliza para gestionar elementos de la interfaz gráfica, como paneles y etiquetas.
2. **java.awt.event:** Contiene clases y interfaces para el manejo de eventos en AWT. En este código, se utiliza para manejar eventos del mouse, como clics, para interactuar con el tablero de ajedrez.
3. **java.awt.image.BufferedImage:** Permite la manipulación de imágenes en formato de mapa de bits. En este código, se utiliza para cargar las imágenes de las piezas de ajedrez desde archivos.
4. **java.io.File:** Proporciona funcionalidades para manipular archivos y directorios. En este código, se utiliza para cargar las imágenes de las piezas desde archivos en el sistema de archivos.
5. **java.io.IOException:** Representa una excepción que indica un error de entrada/salida (I/O). Se utiliza en este código para manejar posibles errores al leer las imágenes de las piezas.
6. **java.util.HashMap:** Implementa la interfaz Map y proporciona una estructura de datos basada en tablas hash para almacenar pares clave-valor. En este código, se utiliza para almacenar las relaciones entre los paneles del tablero y las piezas de ajedrez en sus posiciones correspondientes.
7. **java.util.Map:** Es una interfaz que representa una colección de pares clave-valor. En este código, se utiliza junto con HashMap para almacenar las relaciones entre los paneles del tablero y las piezas de ajedrez.
8. **javax.imageio.ImageIO:** Proporciona clases para leer y escribir imágenes. En este código, se utiliza para cargar imágenes de las piezas de ajedrez desde archivos.
9. **javax.swing:** Proporciona componentes y utilidades para crear interfaces gráficas de usuario (GUI) en Java. En este código, se utiliza para crear y gestionar la ventana principal del juego de ajedrez, así como para mostrar elementos como paneles, etiquetas y listas.

**SOBRE LA CLASE ChessBoardGUI:**

1. Este es el constructor de la clase **ChessBoardGUI**. Se ejecuta automáticamente cuando se crea un objeto de esta clase. Dentro del constructor, se inicializan los componentes de la interfaz gráfica, como la ventana, el tablero, las piezas, el temporizador y otros elementos.
2. **Métodos privados para la configuración y el funcionamiento del juego:**
   * **configurarInterfaz()**: Configura la disposición de los componentes en la ventana, como el tablero, la lista de movimientos, el temporizador y los mensajes del turno.
   * **inicializarTablero()**: Crea el tablero de ajedrez en la interfaz gráfica y coloca las piezas en sus posiciones iniciales.
   * **actualizarLabelTurno()**: Actualiza el mensaje que indica de quién es el turno (Blancas o Negras).
   * **iniciarTemporizador()**: Inicia el temporizador que cuenta el tiempo de juego para cada jugador.
   * **colocarPiezasIniciales()**: Coloca las piezas en sus posiciones iniciales en el tablero.
   * **obtenerNombrePieza(int fila, int columna)**: Determina qué pieza de ajedrez corresponde a una posición dada en el tablero.
   * **cargarImagen(String rutaImagen)**: Carga una imagen de una pieza de ajedrez desde un archivo en el sistema de archivos.
3. **Clases internas privadas:**
   * **EscuchaClicPieza**: Implementa **MouseListener** para manejar los clics del mouse en las casillas del tablero. Permite seleccionar y mover las piezas.
4. **Métodos para la manipulación de las piezas y el tablero:**
   * **moverPieza(JPanel desdeCasilla, JPanel aCasilla)**: Mueve una pieza de una casilla a otra, actualizando la interfaz gráfica y el estado interno del juego.
   * **cambiarTurno()**: Cambia el turno del jugador después de que se realiza un movimiento válido.
   * **obtenerNombreCasilla(JPanel casilla)**: Convierte la posición de una casilla en el tablero (fila y columna) a un formato de nombre de casilla en el ajedrez (por ejemplo, "a1", "e7").
   * **esMovimientoValido(JPanel desdeCasilla, JPanel aCasilla)**: Verifica si un movimiento de una casilla a otra es válido según las reglas del ajedrez para la pieza seleccionada.