# Documentación de Clases:

### Clase Material:

Esta clase representa un material con un nombre y una cantidad.

- Atributos:
  - nombre (String): El nombre del material.
  - cantidad (int): La cantidad disponible de ese material.
- Constructor:
  - Material(String nombre, int cantidad): Crea un nuevo objeto Material con el nombre y la cantidad especificados.
- Métodos:
  - getNombre(): String: Devuelve el nombre del material.
  - getCantidad(): int: Devuelve la cantidad disponible del material.
  - agregarCantidad(int cantidad): void: Aumenta la cantidad del material en la cantidad especificada.

#### Clase **Silla**:

Esta clase representa una silla con un modelo, un ID y un precio.

- Atributos:
  - modelo (String): El modelo de la silla.
  - id (String): El ID único de la silla.
  - precio (double): El precio de la silla.
- Constructor:
  - **Silla(String modelo, String id, double precio)**: Crea una nueva instancia de Silla con el modelo, ID y precio especificados.
- Métodos:
  - **getModelo(): String**: Devuelve el modelo de la silla.
  - getId(): String: Devuelve el ID de la silla.
  - getPrecio(): double: Devuelve el precio de la silla.

# Clase Cliente:

Esta clase representa a un cliente con un ID y un nombre.

Atributos:

- id (String): El ID único del cliente.
- nombre (String): El nombre del cliente.
- Constructor:
  - **Cliente(String id, String nombre)**: Crea un nuevo objeto Cliente con el ID y el nombre especificados.
- Métodos:
  - getId(): String: Devuelve el ID del cliente.
  - getNombre(): String: Devuelve el nombre del cliente.

# Clase **Pedido**:

Esta clase representa un pedido que contiene un cliente y una lista de sillas.

- Atributos:
  - cliente (Cliente): El cliente que realizó el pedido.
  - **sillas** (List<Silla>): Una lista de sillas que forman parte del pedido.
- Constructor:
  - **Pedido(Cliente cliente)**: Crea un nuevo objeto Pedido con el cliente especificado y una lista de sillas vacía.
- Métodos:
  - agregarSilla(Silla silla): void: Agrega una silla al pedido.
  - getSillas(): List<Silla>: Devuelve la lista de sillas en el pedido.
  - getCliente(): Cliente: Devuelve el cliente que realizó el pedido.
  - calcularTotal(): double: Calcula el total a pagar por el pedido sumando los precios de todas las sillas en el pedido.

# Clase FabricaDeSillas:

Esta es la clase principal que representa la fábrica de sillas y contiene el punto de entrada principal del programa.

- Atributos estáticos:
  - **inventarioMateriales** (Map<String, Material>): Un mapa que almacena el inventario de materiales.
  - inventarioSillas (List<Silla>): Una lista que almacena el inventario de sillas.
  - clientes (List<Cliente>): Una lista que almacena la información de los clientes.
  - **pedidos** (List<Pedido>): Una lista que almacena los pedidos realizados.

• materialesNecesariosPorModelo (Map<String, List<Material>>): Un mapa que relaciona modelos de sillas con los materiales necesarios para construirlos.

# • Métodos estáticos:

- main(String[] args): void: El punto de entrada principal del programa.
- Métodos para gestionar los menús de inventario de materiales, inventario de sillas, clientes y pedidos.
- Métodos para agregar materiales, sillas, clientes y crear pedidos.
- Métodos para mostrar el inventario de materiales, sillas y clientes.
- Métodos para buscar clientes y sillas por ID y modelo respectivamente.

#### Diagrama UML de Clases: FabricaDeSillas - inventarioMateriales: Map<String, Material> - inventarioSillas: List<Silla> - clientes: List<Cliente> - pedidos: List<Pedido> - materialesNecesariosPorModelo: Map<String, List<Material>> + main(String[]): void + menulnventarioMateriales(Scanner): void + agregarMaterial(Scanner): void + mostrarInventarioMateriales(): void + menulnventarioSillas(Scanner): void + agregarSilla(Scanner): void Pedido + crearSillaDesdeMateriales(Scanner): void + mostrarInventarioSillas(): void Material - cliente: Cliente + menuClientes(Scanner): void - sillas: List<Silla> + agregarCliente(Scanner): void - nombre: String + mostrarClientes(): void - cantidad: int + menuPedidos(Scanner): void + Pedido(Cliente) + crearPedido(Scanner): void + agregarSilla(Silla): void + buscarClientePorID(String): Cliente + Material(String, int) + getSillas(): List<Silla> + buscarSillaPorModelo(String): Silla + getCliente(): Cliente + getNombre(): String + getCantidad(): int + calcularTotal(): double + agregarCantidad(int): void Silla Cliente - modelo: String - id: String - id: String - nombre: String - precio: double + Cliente(String, String) + Silla(String, String, double) + getId(): String + getModelo(): String + getNombre(): String + getId(): String + getPrecio(): double