EJERCICIO 1

En sí Java no proporciona un mecanismo de exportación selectiva similar al que existe en Eiffel, ya que el control de acceso se basa en modificadores de acceso, como ‘public’, ‘protected’ o ‘private’. Estos modificadores operan a nivel de clase, no a nivel de método específico.

El patrón que hemos llevado a cabo se basa en el patrón método fábrica, en el que los tres tipos de clientes A, B y C, heredan de una interfaz Cliente que define todo tipo de cliente. Por otro lado, están las clases FabricaX, FabricaXA, FabricaXB y FabricaXC, en la que FabricaX es la clase base abstracta que contiene los métodos en común que todas las clases van a poder acceder, a partir de esa, surgen las otras fábricas que extienden a esa inicial, aportando en cada caso los métodos específicos para cada cliente. Por lo que la clase ClienteA va a poder acceder a todos los métodos públicos de FabricaX y a todos los métodos de FabricaXA, y así con cada tipo de cliente.

La ventaja de la solución propuesta es que te aseguras de que todo cliente puede acceder únicamente a los métodos que tiene permiso, ya que la creación de una clase por fábrica y por cliente limita ese aspecto con total seguridad. Sin embargo, tiene un inconveniente, y es que se crean demasiadas clases con poca información a nivel de código, meramente organizativas.