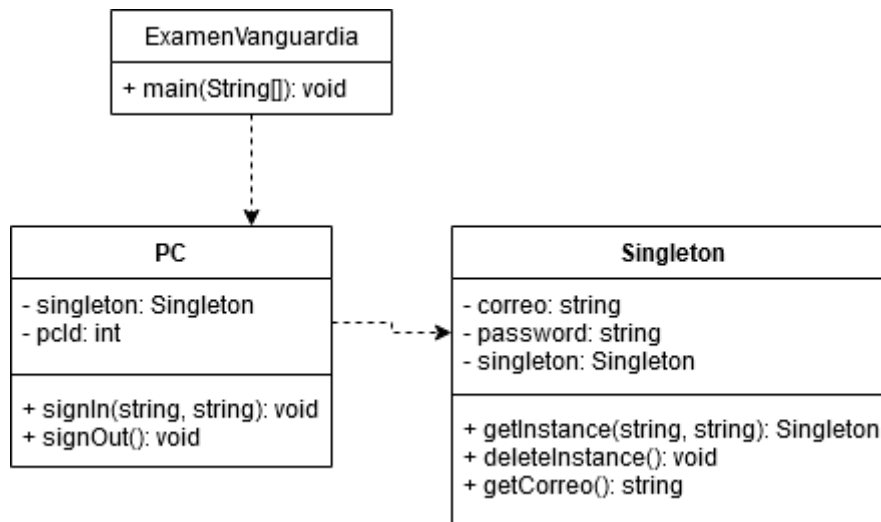


Carlos Enrique Varela Hernández 11711380

Singleton

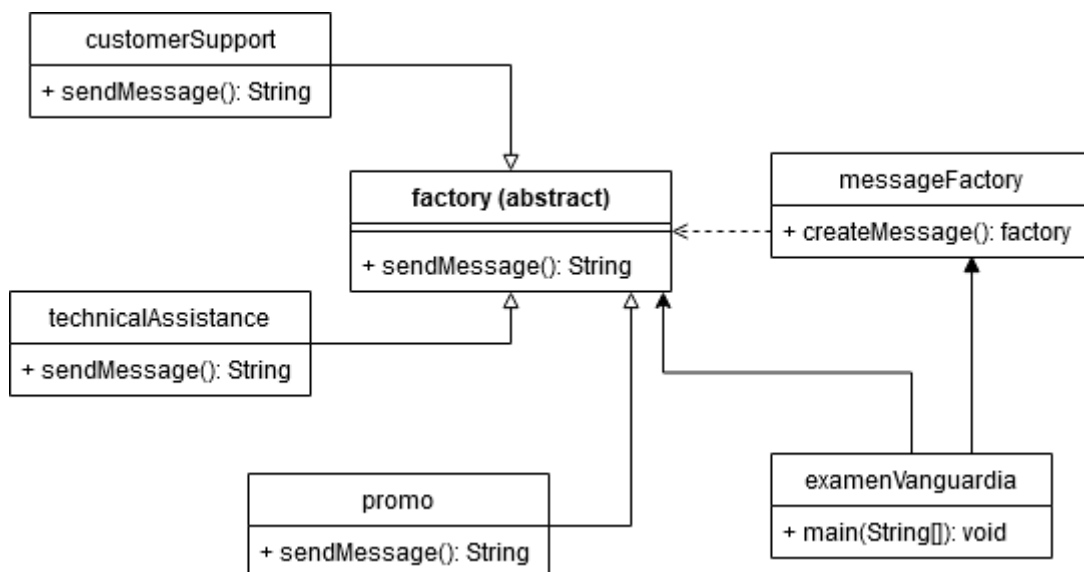
Para este patrón de diseño se toma el caso de los inicios de sesión de WhatsApp web, donde un solo usuario puede iniciar sesión en una computadora, se necesita cerrar sesión en la computadora actual para poder iniciar en otra o poder iniciar sesión otro usuario.

Se toma en consideración que el objeto Singleton es un usuario que actualmente puede crearse (en este caso iniciar sesión) solamente una vez.



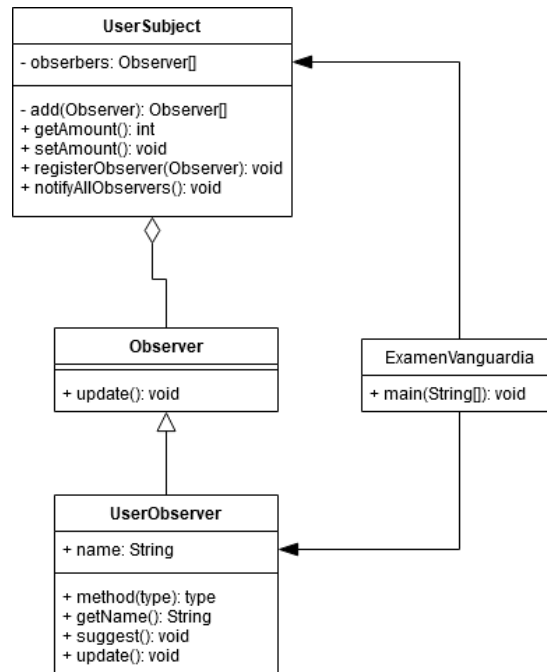
Factory

Para este patrón de diseño se toma el caso de las contestadoras automáticas, donde dependiendo de un tipo de número ingresado en el teléfono se manda un mensaje de un tipo distinto. Por ejemplo, si marca 1 se manda un mensaje de atención al cliente, si marca 2 un mensaje de atención técnica residencial y etc.



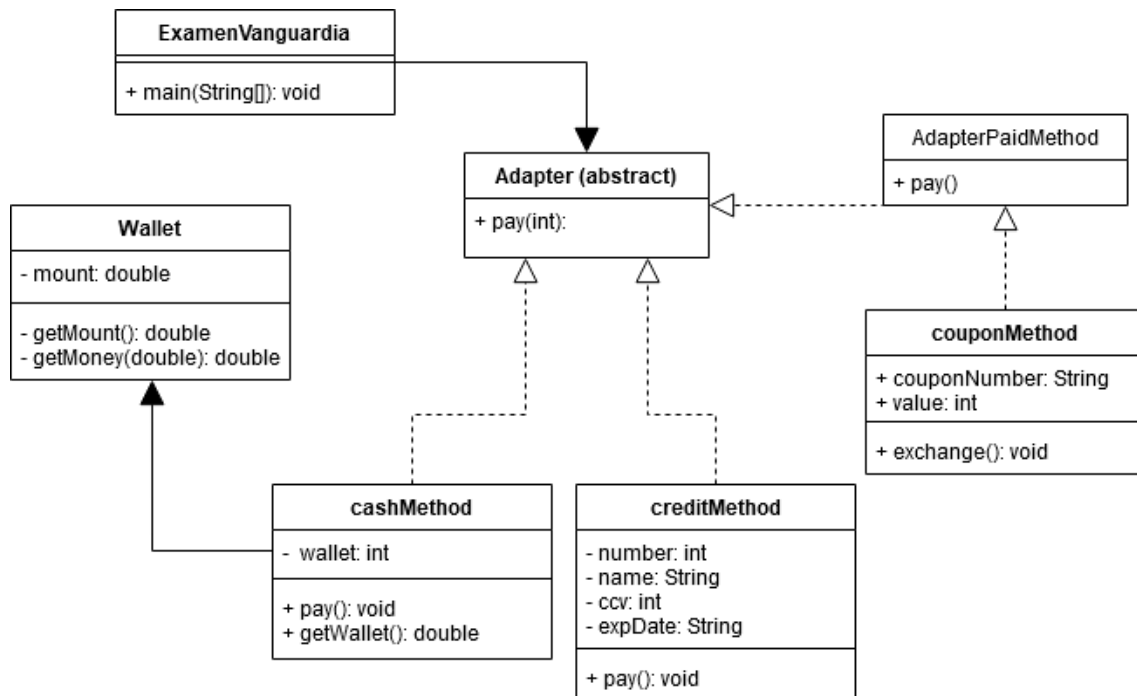
Observer

Para este patrón se considera el caso de las subastas en línea de ebay, donde cada usuario puede dar dinero y este dinero es sumado al dinero actual y notificado a los demás usuarios quienes pueden agregar más dinero, siempre y cuando este sea mayor al dinero total actual.



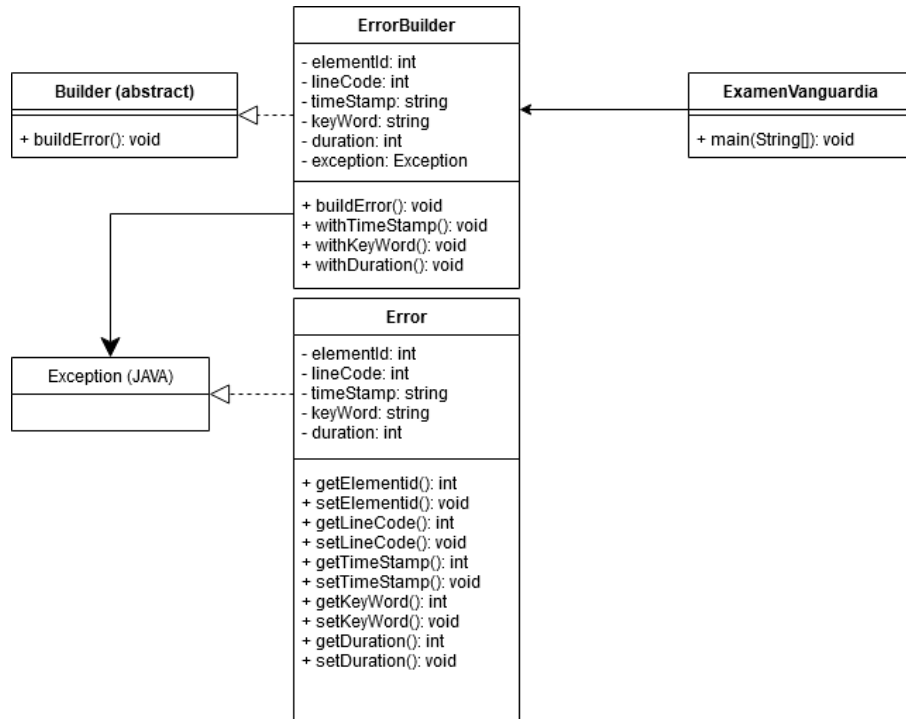
Adapter

Para este patrón de diseño se toma el caso de un sistema de pagos con diferentes opciones de pagos distintos, efectivo virtual con una billetera virtual, tarjeta de crédito o cupón.



Builder

Builder es un patrón de diseño perfecto para crear clases u objetos con gran cantidad de parámetros que no necesariamente todos tienen que ser llenados de manera inmediata al crearse, es por ello por lo que se toma como caso una clase “error” que puede usarse en un try-catch múltiple de veces siempre que sea necesario.



State

Para este patrón de diseño se toma el caso de uso de un sistema de envíos como por ejemplo Amazon en el que el paquete puede cambiar de diversos estados, en proceso, transporte y entregado.

