

Anàlisi i Disseny amb Patrons

PAC 1: Principis de Disseny i Patrons d'Anàlisi

Pregunta 1 (20%)

Suposem que estem desenvolupant un programari que implementa un sistema de xarxa social per competir amb les grans corporacions actuals. La xarxa social es vertebrava al voltant de comunitats, a les quals els usuaris poden unir-se a una data concreta com a membres. Un usuari és membre de la comunitat durant el període que desitgi, i pot abandonar-la a la seva voluntat. En cas que un usuari hagi abandonat una comunitat, pot tornar a unir-se'n de nou posteriorment si així ho desitja.

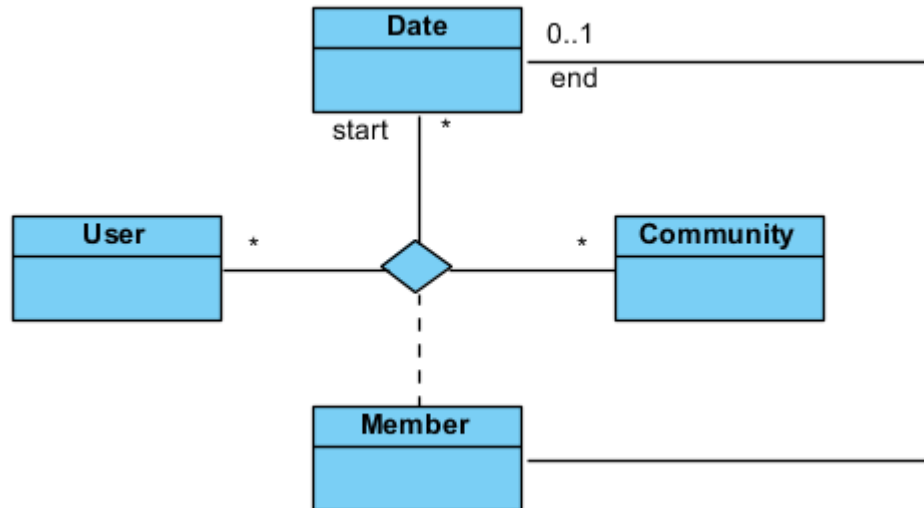
Ens proposem desenvolupar el mòdul que implementa aquesta xarxa social. Abans de fer el diagrama estàtic d'anàlisi, però, revisem els patrons d'anàlisi buscant un que pugui ajudar a resoldre el problema comentat.

Indica quin patró d'anàlisi caldria tenir especialment en compte i justifica breument per què. Proposa un diagrama estàtic d'anàlisi on s'apliqui el patró d'anàlisi identificat pels requisits descrits.

Nota: Si cal, pots fer alguna variació sobre el patró tal com està explicat en els materials per poder satisfer tots els requisits enumerats.

Solució

En aquest cas podem aplicar el patró "Associació històrica" ja que ens permet mantenir un històric de membres dels usuaris a les comunitats, en diferents rangs de dates.



Hem assignat una data de fi a la classe associativa per poder contrastar amb les dates d'inici, de manera que puguem controlar i evitar solapaments entre períodes de membres d'un usuari a una mateixa comunitat.

Pregunta 2 (20%)

Imaginem que estem desenvolupant un programari de *fintech* que permet als usuaris mantenir comptes bancaris en diferents divises. Cada compte bancari disposa d'un saldo especificat en una divisa en concret de les disponibles al sistema.

Específicament, el subconjunt de divises que volem contemplar és el següent:

- Dòlar (USD)
- Euro (EUR)
- Lliura esterlina (GBP)

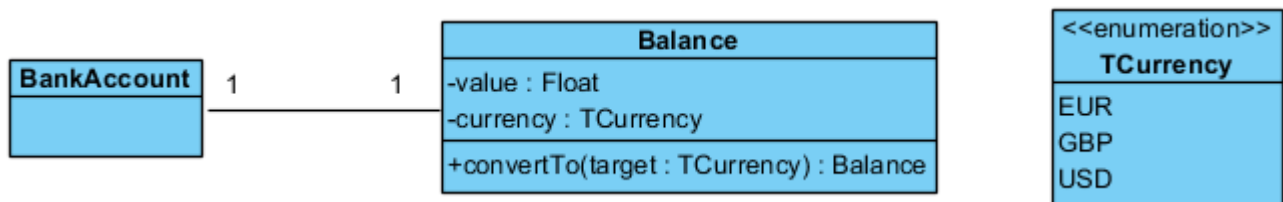
Ens agradaria que, en un futur, es pogués afegir altres divises sense haver de fer grans canvis en el sistema. També ens interessa poder oferir als usuaris la possibilitat de canviar de divisa el seu compte bancari, de manera que necessitem un mètode per convertir el saldo des d'una divisa a una altra.

Indica quin patró d'anàlisi caldria tenir especialment en compte i justifica breument per què. Proposa un diagrama estàtic d'anàlisi on s'apliqui el patró d'anàlisi identificat en els requisits descrits.

Nota: Si cal, pots fer alguna variació sobre el patró tal com està explicat en els materials per poder satisfer tots els requisits enumerats.

Solució

En aquest cas podem aplicar el patró "Quantitat" ja que volem que la unitat de mesura sigui explícita i poder utilitzar diferents unitats de mesura, així com poder fer conversions fàcilment (és a dir, ens trobem davant el problema que resol aquest patró).



Hem aplicat una variació respecte als exemples dels materials on es té en compte el fet que volem poder convertir entre divises. Per a això hem inclòs un mètode a la classe *Balance* per a, donada una divisa destí, obtenir-ne el nou saldo. A més, hem creat una enumeració *TCurrency* per tal de concretar els valors possibles de les divises acceptades. Una solució amb l'atribut *currency* amb tipus de dades *String* també seria vàlida.

Una altra solució possible seria especificar *Currency* com a classe a banda, incloent-hi el mètode de conversió de divises, de la següent manera:



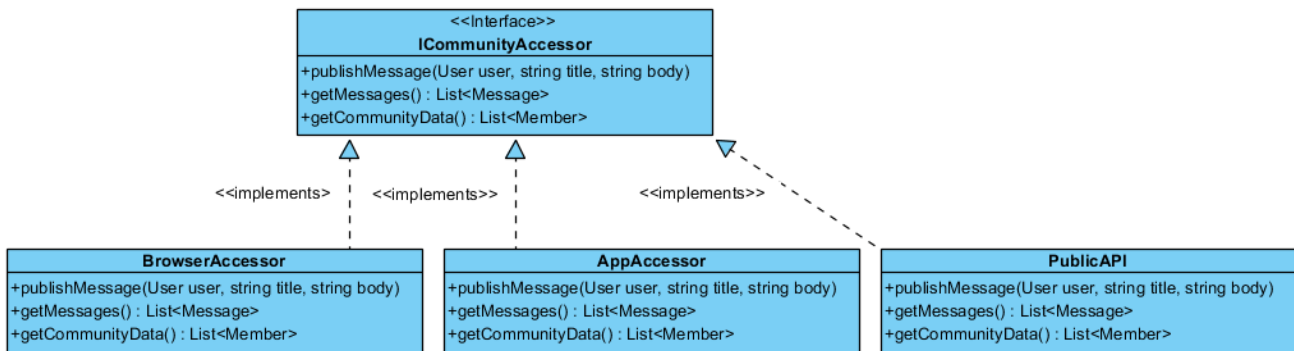
Amb això, concretam més la responsabilitat del tipus de canvi a la divisa, en comptes de tenir-la agregada al saldo.

Pregunta 3 (30%)

Un dels principis de disseny que ha de complir un sistema per obtenir un nivell alt de qualitat és el principi de "Segregació d'Interfícies" que diu que "Els clients no haurien de dependre d'operacions que no utilitzen".

Suposem ara que estem dissenyant el sistema de xarxa social de la Pregunta 1. En concret, hem de contemplar les típiques operacions de publicació de missatge a una comunitat i la consulta de missatges publicats a les mateixes. Al mètode de publicació de missatge necessitem indicar l'usuari que realitza la publicació, el títol de la mateixa i el cos del missatge. Tenim un mètode de consulta de missatges publicats a una comunitat que retorna el llistat de missatges (inclou data de publicació, nom i avatar de l'autor, títol i cos). A més, es pot sol·licitar el llistat complet de membres de la comunitat (inclou nom, avatar i última data d'alta de membre).

En un diagrama de classes, podríem representar-ho com:



La xarxa social està disponible per al seu accés tant des de navegadors de dispositius desktop (Windows i Mac) com des d'apps per a dispositius mòbils Android i iOS. Als navegadors per a Windows i Mac es mostrarà la informació completa retornada pel mètode de consulta de missatges i pel mètode de consulta de membres de la comunitat. En canvi, a les apps tindrem una visió més reduïda per no malbaratar dades i per això el llistat de missatges d'una comunitat només mostra les dades d'aquests missatges, obviant el mètode que proporciona el llistat adicional amb tots els membres de la comunitat.

La nostra xarxa social exposa públicament una API mitjançant la qual qualsevol sistema extern pot obtenir els missatges publicats a una comunitat, sense la informació adicional dels seus membres, com fan les nostres apps. Però a aquesta API pública no volem exposar el mètode de publicació de missatge.

- a) Aquest exemple constitueix o pot constituir una violació del principi de segregació d'interfícies. Per què?
- b) Quines conseqüències tindria aquesta violació?

Solució

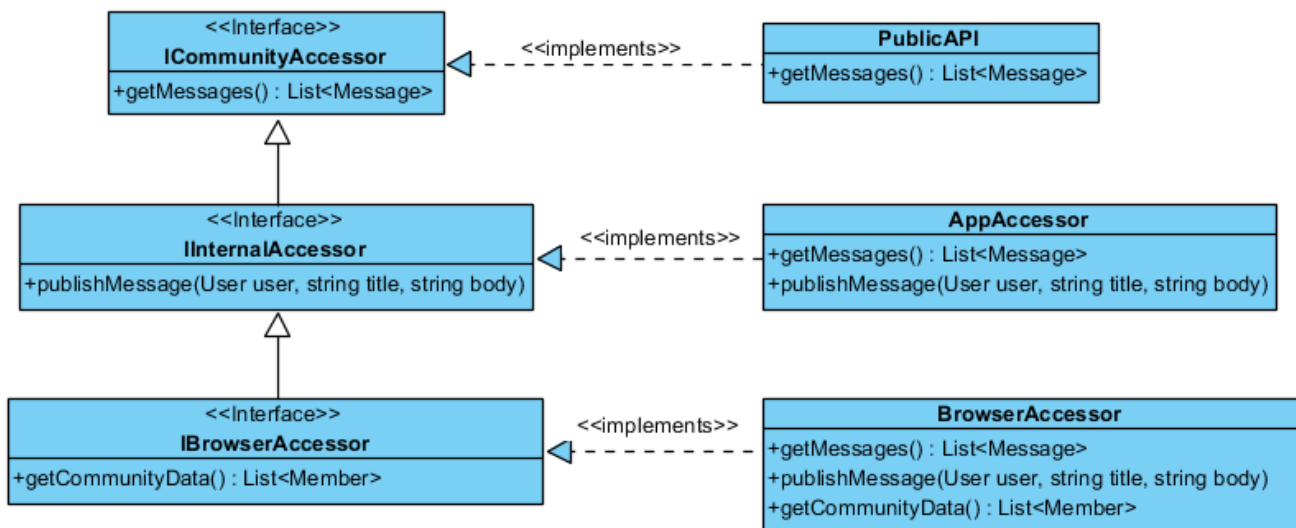
- a) Es tracta d'una violació del principi de segregació d'interfícies ja que per mostrar el llistat de missatges des de qualsevol de les apps estem depenent d'informació que no fem servir, com és el llistat complet de membres de la comunitat consultada. Aquest mètode, per tant, no ha de ser implementat per les apps. D'altra banda, l'API necessita implementar una excepció per al mètode de publicació de missatge, quan en realitat no es permet a sistemes externs realitzar aquesta acció.
- b) La principal conseqüència és que serà difícil fer evolucionar el sistema ja que cada vegada que vulguem afegir dades a tornar als navegadors de Windows i Mac podrem afectar la implementació de les apps (per què el seu UI ignori aquestes noves dades). A més, qualsevol mètode que vulguem afegir al nostre sistema tindrà la seva corresponent implementació a l'API, de manera que (1) qualsevol canvi de la nostra funcionalitat afectarà la integració de sistemes externs; i (2) potencialment estarem exposant mètodes que no volem que es coneguin públicament, tot i que la implementació es limiti al llançament d'una excepció.

Pregunta 4 (30%)

Proposa com es podria dissenyar el sistema anterior per què se satisfaci el principi de segregació d'interfícies.

Utilitza diagrames estàtics d'anàlisi, diagrames de seqüència i/o pseudocodi d'operacions segons creguis més rellevant per explicar la solució proposada. Raona, també, com la solució supera les conseqüències negatives de l'apartat *b* de la pregunta anterior.

Solució



Com podem veure en aquesta solució, i gràcies a la segregació d'interfícies, cadascuna de les classes que proporcionen accés a les dades d'una comunitat implementa únicament i exclusivament els mètodes apropiats, són independents, de manera que les desacoplem. Com a resultat, a més, cap de les classes ha d'implementar mètodes buits.