Université Abou Beckr Belkaid
Master 2 Génie Logiciel
Année Universitaire 2021/2022

Examen de rattrapage

—

Les technique de construction d'architectures logicielles avancées

—

Remarques :

— Les documents ne sont pas autorisés ainsi que les appareils électroniques (PC, Tablette, téléphone..).
— La lisibilité et la clarté de vos réponses et de votre code sont très importantes. Une réponse pas claire ne sera pas prise en compte lors de la correction.

Questions de cours : (3 points)

1 Donner la définition des designs pattern.
2 Donner les différences entre les classes sessions bean et entités bean.

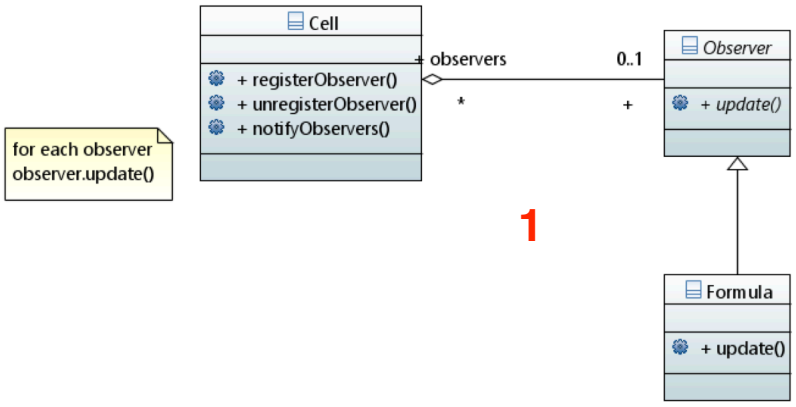
Exercice 1 :(5 points)

Pour chaqu'un des exemples suivants donner :

1 le design pattern approprié, en justifiant.
2 le diagramme UML de la solution en utilisant le design pattern choisi.

— **L'exemple 1 :** Vous développez un tableur qui permet de calculer automatiquement les cellules en fonction des formules, en sachant que les formules peuvent changer a tout moment.

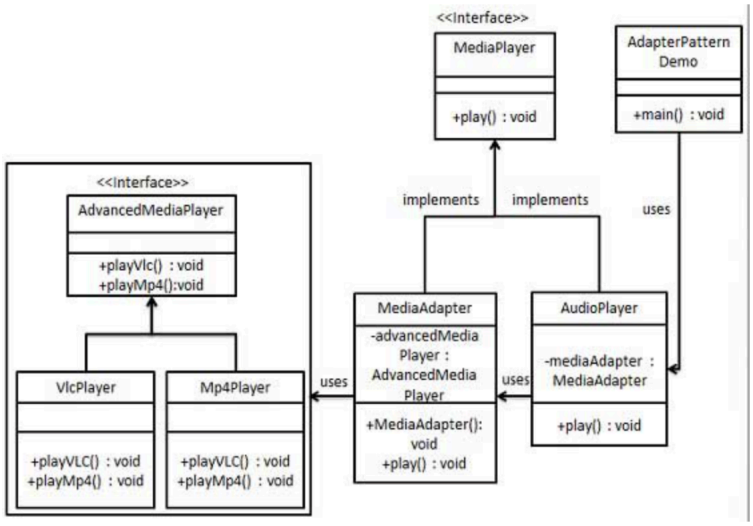
1,5 le design pattern Observer, car les cellules observent le changement de la formule pour recalculer les valeurs.



— **L'exemple 2 :** Une application de lecteur multimédia permet de lire des fichiers MP3. Nous souhaitons réutiliser l'implémentation de cette version pour obtenir une version avancée qui permet de lire des fichiers VLC et MP4.

1,5 Le design pattern Adapter, la version avancée utilise l'implémentation des fichier MP3 pour implémenter les fichiers VLC et MP4.

1



Exercice 2 :(5 points)

Des utilisateurs souhaitent avoir une application qui permet d'encoder un String afin de l'enregistrer dans un bufferWriter. La solution proposée est la suivante :

```
+ public class StringCodec implements SimpleCodec<String> {
+
+     @Override
+     public String decode(Reader reader) {
+         throw new UnsupportedOperationException("Should never have to decode a String object");
+     }
+
+     @Override
+     public void encode(final String value, Writer writer) {
+         writer.writeString(value);
+     }
+ }
```

1,5 1 Le principe SOLID non respecté est ISP (interface segrégation) car la méthode *decoder* n'est pas utilisée par les utilisateurs de la classe *StringCodec*..

2 2 Donner une définition simple de ce principe.(voir le cours)

1,5 3 La solution pour résoudre le problème : diviser *StringCodec* en deux interfaces une pour encoder et l'autre pour décoder.

Exercice 3 : (7 points)

Dans un système d'achat en ligne de livres. Le client à le choix entre :

— acheter un livre en format électronique (en .pdf).

— acheter un livre en format papier, qui va être envoyer via une adresse postale.

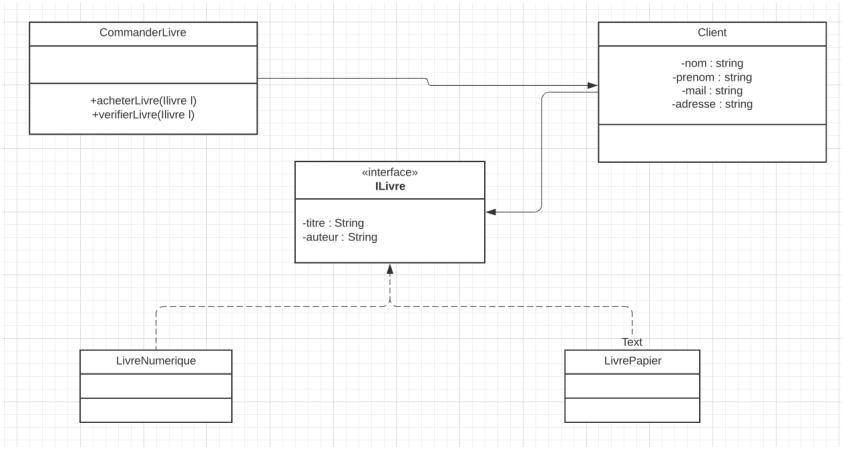
Le client est connu par son nom/prénom, son email et son adresse postale. Le livre est définit par un titre et un auteur.

Avant de passer commande, un client devra s'assurer que le livre est disponible dans la boutique en ligne.

Questions :

1 donnez une conception évolutive à ce système, en utilisant un digramme UML de classes.

3



2 2 En utilisant les EJB, préciser les classes sessions beans et entités beans, en justifiant votre choix.

2 3 Donnez une implémentation simple de la classe session bean.