

TD 15 - Projet Zeldiablo - Conception et première réalisation (2h)

1 Conception (50 min)

1.1 Répartition des fonctionnalités (10 min)



Question 1

Répartissez-vous le travail en vous attribuant entre vous les responsabilités des fonctionnalités. Précisez cette responsabilité à côté des fonctionnalités (chaque fonctionnalité doit avoir un responsable).

Le fichier de fonctionnalités doit donc avoir cette forme:

```
* fonctionnalités

- fonctionnalite 1 : [...]
- responsable: login1

- fonctionnalite 2 : [...]
- responsable: login2

[....]
```

Attention :

L'ensemble des membres du projet doit participer au projet. Le travail en équipe est l'une des difficultés de cet exercice.

1.2 Diagramme de séquence et de classes (40 min)



Question 2

Pour chaque fonctionnalité, le responsable doit dessiner sur une feuille un diagramme de séquence expliquant comment votre application est censée fonctionner.



Question 3

Mettez progressivement et collectivement à jour votre diagramme de classe pour qu'il soit compatible avec vos diagrammes de séquence. Ne présentez dans ce diagramme que les classes utiles à la version 1.



Question 4

Photographier et déposer l'ensemble de vos diagrammes sur le dépôt (dans le répertoire documents de la version1). Si le temps manque, passez cette étape, mais faites le pour la prochaine séance.



Question 5

Appelez votre enseignant avant de commencer à écrire le code source de votre application.

2 Réalisation (1h)

2.1 Ecriture du code (40 min)

Vous pouvez ensuite commencer l'écriture collective du code selon ces contraintes ci-dessous

Démarche

- à l'issue des deux heures, vous devez avoir une application qui fonctionne et qui a été testée;
- écrivez les tests unitaires à partir de vos fonctionnalités et avant d'écrire le code de l'application;
- écrivez des commentaires dans les méthodes avant d'écrire le code java;
- écrivez la javadoc au fur et à mesure;
- pensez à faire des pull et des push régulièrement pour limiter les conflits.

Qualité de code

- ne faites pas de copier-coller, s'il y a un copier-coller, votre code doit être restructuré (appelez votre enseignant si besoin);
- respectez les conventions de nommage (cf fichier);
- écrivez la javadoc et les commentaires au fur et à mesure du code;
- écrivez des tests unitaires qui vérifient l'ensemble de votre code.

Question 6

Répartissez-vous le travail en fonction des responsabilités précédentes et des diagrammes de classe et de séquence (quel étudiant code quelle classe ou quelle méthode ?). N'oubliez pas qu'il faudra aussi coder les tests.

Question 7

Dans le dépôt que vous avez créé, un des membres du groupe doit créer un projet eclipse "*projet_zeldiablo*". Les autres membres du groupe doivent cloner le repository.

Question 8

Écrire collectivement le code de la première version de l'application.

2.2 Finalisation de la version (20 min)

Une fois le code écrit,

- vérifiez d'abord que vos tests sont valides et corrigez tant que ce n'est pas le cas;
- ajoutez un tag avec le numéro de version v1.0 et envoyez le au serveur;
- générez un diagramme de classe avec objectAid et sauvez le fichier image dans le répertoire *version_1*;
- analysez les différences avec votre diagramme initial si besoin. Ce diagramme sera votre point de départ pour la version suivante.

A l'issue de votre travail, vous devez disposer d'un fichier structuré de la manière suivante

```
* fonctionnalités

- fonctionnalite 1 : [....]
- responsable: login1
- diag de sequence: nom fichier image
- explication textuelle fonctionnement

- fonctionnalite 2 : [....]
- responsable: login2
- diag de sequence: nom fichier image
- explication textuelle fonctionnement

[....]

* réalisation

- classe ou méthode ou test
- responsable : login

- classe ou méthode ou test
- responsable : login

[....]
```



Question 9

I Finaliser la première version de l'application.