

Examen de Génie Logiciel

2nde session - L3 informatique

Responsables : C. Roudet et B. Bordeaux

Documents autorisés : feuille A4 RV avec notes manuscrites + double page des DP

Juin 2025. Durée : 1h30

Exercice 1 : Questions de réflexion portant sur le cours (6 pts/20)

ATTENTION : pour chaque question, **justifiez vos réponses et faites des phrases** : toute réponse non justifiée sera considérée comme nulle et notée comme telle. 8-10 lignes maximum pour chaque réponse.

1. Comment peut-on rendre une application informatique (écrite en Java) conforme au principe O de SOLID, en permettant l'extension des classes tout en les rendant fermées aux modifications ? Vous pouvez vous servir d'un exemple pour illustrer votre explication.
2. Qu'est-ce que la couverture dans le contexte des tests logiciels ? Comment est-elle concrètement exploitée ? Vous pouvez vous servir d'un exemple pour illustrer votre explication.
3. Selon vous, comment peut-on réduire l'écart entre les besoins réels d'un client et les besoins exprimés dans le cahier des charges ?

Exercice 2 : Design Pattern et diagramme de classe (7 pts/20)

L'entreprise Héber propose **trois services d'hébergement web** : perso, pro et premium. Chaque offre peut inclure plusieurs **services supplémentaires** tels que SSL, support 24/24, sauvegardes, Google Ads (programme de publicité en ligne de Google), CloudFlare (pour améliorer la sécurité et intensifier les performances des sites et services Web), MySQL.

On s'intéresse à la conception d'une application de calcul du **coût mensuel du service, en fonction de l'offre choisie et des services supplémentaires sélectionnés**. L'application doit également être flexible pour **ajouter de nouvelles offres** lorsque celles-ci deviennent disponibles, tout en évitant une explosion du nombre de classes.

Par exemple, une combinaison choisie par un client pourrait être : Hébergement perso, SSL, Google Ads, MySQL.

1. Quel design pattern vous semble le plus approprié à ce contexte. **Justifiez** en précisant pourquoi ce pattern répond bien au problème mentionné.
2. Donnez le diagramme de classe spécifique au cas présenté ci-dessus (en respectant bien les conventions des diagrammes de classe UML). Même si aucun détail n'est précisé concernant les coûts mensuels précis, vous ferez bien apparaître la méthode **cout_mensuel** (dans les classes adéquates), puisque c'est le service minimum demandé par l'application.

Exercice 3 : Tests fonctionnels, structurels et unitaires (7 pts/20)

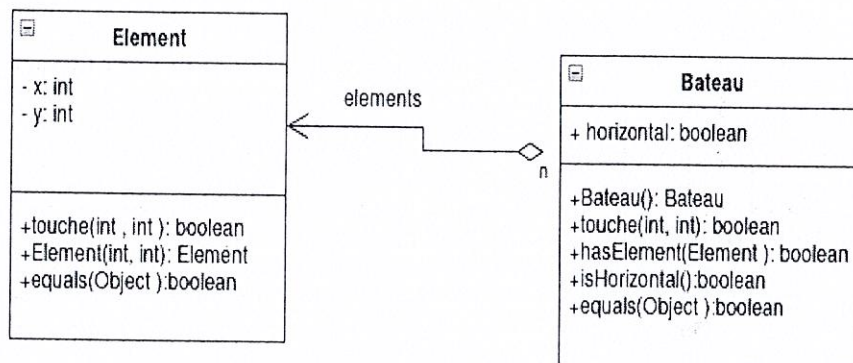


Figure 1 : diagramme de classes pour la gestion de bateaux

Considérons une application de **gestion de Bateaux** dans une grille 2D (permettant de créer une instance de **Bateau**, de comparer deux **Bateaux**, de vérifier qu'ils ne se touchent pas ...). Cette application sert de base pour la création du jeu de bataille navale.

On suppose qu'un **Bateau** est constitué d'un ensemble d'**Éléments unitaires**. Pour parler de **Bateau** il faut que les **Éléments** unitaires le constituant soient alignés horizontalement ou verticalement. Chaque **Élément** a une position (x, y) dans la grille, car il pourra être bombardé par un adversaire à la bataille navale.

Les valeurs extrêmes possibles pour un **Bateau**, relativement à son nombre d'**Éléments** sont : min = 1 et max = la largeur/hauteur de la grille sur laquelle on le place). Le diagramme de classes est présenté dans la figure 1, les méthodes de chaque classe ne sont pas exhaustives. Nous précisons aussi que la méthode **touche** permet de savoir si un **Élément** d'un bateau touche un **Élément** d'un autre bateau.

Vous êtes chargé de mettre en place des **tests** pour assurer la qualité du logiciel. **Répondre aux questions suivantes à l'aide de phrases argumentées :**

1. Qu'est-ce qu'un **test fonctionnel** et quel est son objectif principal dans le contexte de l'application de gestion de Bateaux ?
2. Donnez un exemple concret de test fonctionnel pour cette application, en précisant une fonctionnalité à tester et les résultats attendus.
3. Qu'est-ce qu'un **test structurel** et quel est son objectif principal dans le contexte de l'application de gestion de Bateaux ?
4. Donnez un exemple concret de test structurel pour cette application, en précisant les éléments de structure à tester et les critères d'évaluation.
5. Qu'est-ce qu'un **test unitaire** et quel est son objectif principal dans le contexte de l'application de gestion de Bateaux ?
6. Donnez un exemple concret de test unitaire (en *JUnit*) pour cette application, en précisant la méthode à tester. Écrire des **cas de test** couvrant **deux scénarii différents** (que vous préciserez) pour cette méthode.