

Numéro étudiant :



Génie logiciel Devoir sur table

Licence 2 informatique 2020-2021

Durée 1h30

Documents de cours autorisés

Les ordinateurs, calculatrices, téléphones portables sont interdits

A. Diagramme de Pert (3 points)

Nous devons mettre en place un projet pour la création d'un site web pour gérer la vente en ligne de produits.

Les tâches sont les suivantes :

- CBD : Création de la base de données (3j)
- CSW : Création de la structure du site web (4j)
- GP : Gestion des produits (10j) la base et la structure du site doivent être réalisées
- GC : Gestion des clients (5j) la base et la structure du site doivent être réalisées
- GA : Gestions des achats (4j) les clients et les produits doivent être gérés
- GB : Gestion des bilans (4j). Cette tâche peut être réalisée après la mise en ligne du site
- MLS : Mise en ligne du site qui permet les achats (5j)

1. Réaliser le **diagramme de PERT**.
2. Combien de temps va durer le projet ? Y-a-t-il un chemin critique ?
3. Combien d'équipes peuvent travailler en parallèle ?

B. Agile & Scrum (2 points)

4. Pourquoi ne pas réaliser un diagramme de Gantt dans le cadre d'une méthode agile ?
5. Lors de la review d'un sprint, la présentation au product owner d'une user story n'a pas été satisfaisante que ce passe-t-il ?

C. Diagrammes UML (7 points)

Nous désirons définir un système d'information pour gérer des équipes de football. A l'aide de classes, nous désirons gérer d'une part les salariés des équipes. Ils ont un nom, un prénom, un pseudo et un salaire. Les salariés sont soit des joueurs, des entraîneurs, des administratifs et des soigneurs. Pour les joueurs, nous désirons connaître leur poste sur le terrain (gardien, milieu de terrain, attaquant, etc.) et gérer leurs transferts entre équipes. Les soigneurs ont une fonction : médecin, kinésithérapeute, etc. De même, les administratifs ont un rôle. Pour les entraîneurs nous désirons connaître leur diplôme d'entraîneur : DE1, DE2, etc. Chaque équipe regroupe des salariés et a un budget. Lors d'un transfert du joueur J de l'équipe A vers l'équipe B. L'équipe B retire de son budget une somme qui est affectée à l'équipe A et le joueur J fait alors partie des salariés de l'équipe B.

6. Définir le **diagramme de classes** détaillé (classes, associations, attributs, méthodes avec paramètres, cardinalités, héritages). Vous devez aussi spécifier si les classes et les méthodes sont abstraites.
7. Définir un **diagramme de séquences** pour la méthode transfert. L'ensemble des méthodes appelées doivent être définies dans le diagramme de classes.

D. Documentation et Git (3 points)

8. Ecrire sous la forme d'une FAQ de quelques lignes un document qui explique :
 - Comment identifier une version
 - Comment récupérer un projet git distant pour la première fois ?
 - Comment supprimer les modifications réalisées depuis le dernier commit sur un seul fichier sans supprimer les mises à jour des autres fichiers ?

E. Test (5 points)

Nous désirons réaliser des tests sur la gestion du passage classe supérieure. Pour rappel, voici la procédure de passage : Pour passer directement un étudiant doit avoir au moins 10 de moyenne sur les deux semestres, sinon il peut aller en rattrapage s'il a au moins 10 à un semestre ou au moins 8 de moyenne sur l'année. Il peut aussi être ABI à l'un des deux semestres dans ce cas, il redouble.

La méthode passage prend en entrée 2 nombres réels entre 0 et 20 ou -1 si ABI et produit en sortie une valeur entre 0 et 2 (0 redoublement, 1 rattrapage, 2 passage)

```
/**
 * permet de savoir si un étudiant passe, redouble ou a le rattrapage
 * en fonction de ses notes
 * @param s1 moyenne semestre 1
 * @param s2 moyenne semestre 2
 * @return 0: redoublement, 1: rattrapage, 2 : passage
 */

public static int passage(float s1, float s2) {
    int resu = -1;
    if (s1 == -1)           // ABI en S1
        resu = 0 ;          // redoublement
    else
        if (s2 == -1)       // ABI en S2
            resu = 0;        // redoublement
        else                 // pas d'ABI
            if (((s1+s2)/2)>10)
                resu = 2;     // passage si la moyenne générale
            else
                if (s1>10)    // pas la moyenne générale mais un s1 > 10
                    resu = 1; // rattrapage
                else
                    if (s2>10) // pas la moyenne générale mais un s2 > 10
                        resu = 1; // rattrapage
                    else
                        if (((s1+s2)/2)>=8) // moyenne générale >= 8
                            resu = 1; // rattrapage
                        else // moyenne générale < 8
                            resu = 0; // redoublement
    return resu;
}
```

9. Dessinez le **graphe de flot** et exprimez les chemins de contrôle sous **forme algébrique**. Vous pouvez ajouter des lettres à côté des instructions de cette feuille puis utiliser ces lettres pour votre diagramme. Merci de rendre la feuille dans ce cas.
10. A l'aide de ce graphe, proposez des données de test (DT) et leur oracle. Avec ces données allez-vous trouver des erreurs dans cette fonction ?
11. Ajouter de nouvelles données de tests pour **tester aux limites** cette fonction. Avec ces données allez-vous trouver des erreurs dans cette fonction ?
12. Pour la JavaDoc, quel commentaire ajouter pour connaître l'auteur de cette méthode et dire que c'est la version 1.1