

Génie logiciel

Devoir sur table

Licence 2 informatique 2019-2020

Durée 1h30

Documents de cours autorisés

Les ordinateurs, calculatrices, téléphones portables sont interdits

A. Diagramme de Gantt (4 points)

Soit les cinq tâches suivantes qui doivent être réalisées par les équipes A, B et C

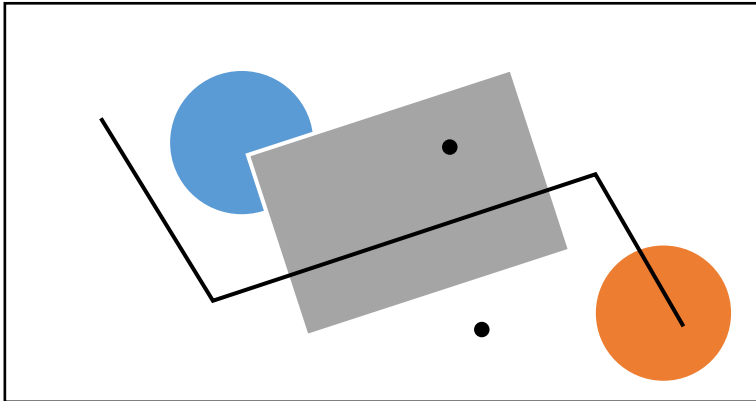
- Tâche 0 : une semaine par les équipes A et B
 - Tâche 1 : deux semaines par l'équipe C.
 - Tâche 2 : deux semaines par l'équipe A, la tâche 0 doit être finie pour que la tâche 2 commence.
 - Tâche 3 : trois semaines par l'équipe B, la tâche 0 doit être réalisée au préalable.
 - Tâche 4 : deux semaines par les équipes A, C. les tâches 1 et 2 doivent être finies au préalable ?
1. Réaliser le **diagramme de Gantt** pour ces tâches avec les contraintes de précédences et le nom des équipes spécifiés pour chaque tâche.
 2. Combien de temps va durer le projet ?
 3. Y a-t-il des semaines ou une des équipes ne travaille pas ?

B. Agile & Scrum (3 points)

4. Pourquoi ne pas réaliser un diagramme de Gantt dans le cadre d'une méthode agile ?
5. Pourquoi définir des sprints et des releases ?
6. 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 définissons un logiciel de dessin géométrique. Un utilisateur peut dessiner différentes figures de couleur : des points, des lignes, des rectangles et des ronds. Un point a deux coordonnées x et y . Une ligne est une suite ordonnée de points qui lui sont propres. Un rectangle est défini par un point de base, les longueurs de ses deux côtés et l'angle entre son premier côté et l'axe des abscisses. Un rond a un point centre et un rayon. Un dessin est défini par un nom, une feuille qui est un rectangle et un ensemble ordonné de figures. Une figure ne peut être qu'à l'intérieur du dessin. Si elle dépasse, le dessin est agrandi.



Pour les figures des méthodes communes existent :

- aire. Les points et les lignes ont une aire (surface) de valeur 0.
- intersection qui renvoie vrai si les deux figures ont une intersection.
- aireIntersection qui renvoie l'aire de l'intersection

Le dessin a aussi des méthodes spécifiques :

- aire qui renvoie l'aire de la feuille,
 - pourcentage qui renvoie le pourcentage de la feuille ayant des figures dessinées.
 - ajout qui permet d'ajouter une figure au dessin. La figure est obligatoirement ajoutée à la fin de l'ensemble et dessinée au-dessus des autres figures.
 - agrandir qui permet d'agrandir la feuille du dessin si besoin lors de l'ajout d'une figure
7. 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.
 8. Définir un **diagramme de séquences** pour la méthode pourcentage d'un dessin ayant deux figures f1 un rond et f2 un rectangle. Les figures f1 et f2 ont une intersection.

D. Test unitaire et d'intégration (6 points)

Soit la fonction `divisionPositif` suivante :

```
public static int divisionPositif(int[] tab, int k)
{
    int resu = 0;
    for(int i=0; i<tab.length ; i++)
    {
        resu += tab[i];
        if (tab[i]>0)
            tab[i] = tab[i]/k;
    }
    return resu;
}
```

Cette fonction renvoie la somme des entiers du tableau passé en paramètre et divise les nombres positifs du tableau par `k`.

9. Dessinez le **graphe de flot** et exprimez les chemins de contrôle sous **forme algébrique**.
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. A partir du digramme de séquences de la question 8, proposez un ordre de **tests** des méthodes impliquées dans ce diagramme pour une approche **d'intégration ascendante**.