



CSC4102 : À propos des oraux de validation

Denis Conan

Janvier 2025





Sommaire

1. Agenda
2. Organisation
3. Sujets
4. Thèmes des questions
5. Exemples de questions de niveau 1
6. Exemples de questions de niveau 2
7. Exemples de questions de niveau 3

1 Agenda

- Agenda : lundi matin 29 avril (9h) et mardi matin 30 avril (10h)
 - Cf. planification accessible depuis la page d'accueil du site Web du module
 - « Planif. oraux de valid. »
https:
[//www-inf.telecom-sudparis.eu/COURS/CSC4102/?page=planification_oraux_de_validation](https://www-inf.telecom-sudparis.eu/COURS/CSC4102/?page=planification_oraux_de_validation)
- Les oraux de validation sont individuels
- Chaque étudiant et étudiante se présente à l'heure indiquée
- Les retards ne peuvent pas être permis
 - Tout retard est pris sur la durée de l'oral

2 Organisation

- Les oraux s'effectuent en salle avec un jury de deux personnes
- Chaque oral dure 2+10 minutes dans un créneau de 15 minutes
 - 2 minutes de préparation avec le polycopié (uniquement la version papier)
 - 10 minutes de réponse aux questions (sans le polycopié)
 - Les oraux sont donc cadencés de quart d'heure en quart d'heure
- Un membre du jury partage son écran en vidéoprojection
- Vous répondez à l'oral devant le tableau et utilisez le tableau si nécessaire

3 Sujets

- Les sujets comportent 5 questions à traiter dans l'ordre :
 - 2 questions de niveau 1
 - 2 questions de niveau 2
 - Dont la seconde est construite à partir de votre solution à l'étude de cas (version extraite lors de la livraison)
 - Dans cette question, nous n'évaluons pas votre solution à l'étude de cas, mais l'explicitation de la démarche et la capacité à défendre votre solution
 - 1 question de niveau 3
- Les questions sont présentées et traitées par ordre de niveau
- Une question abandonnée ne peut pas être reprise ensuite
- Rappel : une note inférieure à 5 ne permet pas de valider le module
 - Les 3 premières questions constituent 5 points
 - La quatrième, 3 points
 - La dernière, 2 points

4 Thèmes des questions

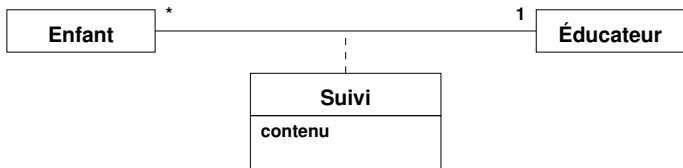
- Pour rappel : grille d'auto-évaluation structurée par séance
https://www-inf.telecom-sudparis.eu/COURS/CSC4102/?page=grilles_auto_evaluation
- Thèmes
 - Gestion de versions
 - Spécification
 - Préparation des tests de validation
 - Conception préliminaire
 - Conception détaillée
 - Préparation des tests unitaires
 - Programmation outillée
 - Programmation des tests
 - Qualité du code et idiomes
 - Qualité du modèle et patrons de conception
- D'une manière générale,
toute diapositive fait potentiellement l'objet d'une question

5 Exemples de questions de niveau 1

- Expliquez le rôle de la spécification dans le développement d'un logiciel et indiquez ce que l'on a fait dans le module pour cette activité
- Indiquez, en argumentant votre réponse, si un message dans un diagramme de séquence est la plupart du temps traduit en une opération privée
- Expliquez le concept de classe abstraite et donnez un exemple (qui peut s'inspirer de votre solution à l'étude de cas)

6 Exemples de questions de niveau 2 I

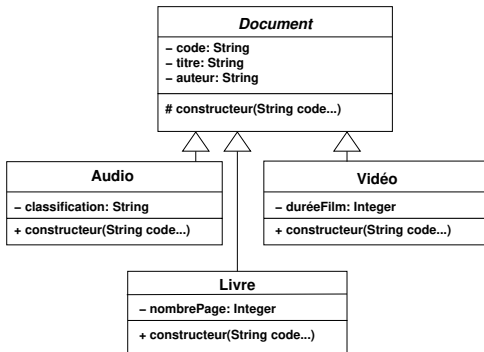
- Pour les besoins de la conception détaillée, traduisez en deux associations la classe d'association du diagramme ci-dessous, en argumentant votre réponse



- A priori, vous utilisez le tableau pour écrire votre réponse

6 Exemples de questions de niveau 2 II

- Expliquez le choix des visibilités montrées dans le diagramme de classes qui suit



6 Exemples de questions de niveau 2 III

- En considérant que le code qui suit compile sans erreur, lisez et commentez le code JAVA des lignes qui suivent en vous focalisant sur les différentes catégories de méthodes :

```
1 public class Copropriete {
2   private String nom; private String adresse;
3   private Integer tantieme;
4   public Copropriete(final String nom, final String adresse,
5                     final Integer tantieme) {
6     if (nom == null || nom.equals("")) {
7       throw new IllegalArgumentException("nom non valide (" + nom + ")");
8     }
9     //...
10    this.nom = nom; this.adresse = adresse; this.tantieme = tantieme;
11    assert invariant(); }
12  @Override
13  public int hashCode() { //...
14  @Override
15  public boolean equals(Object obj) { //...
16  @Override
17  public String toString() { //...
18  public String getNom() { return nom; }
19  public boolean invariant() {
20    return nom != null && !nom.equals("") && adresse != null
    && !adresse.equals("") && tantieme > 0; } }
```

6 Exemples de questions de niveau 2 IV

- En considérant que le code qui suit compile sans erreur, lisez et commentez le code JAVA des lignes qui suivent :

```
1
2 class TestCopropriete {
3     Copropriete c;
4
5     @BeforeEach
6     public void setUp() {
7         c = new Copropriete("Un", "ici", 100);
8     }
9
10    @AfterEach
11    public void tearDown() {
12        c = null;
13    }
14
15    @ParameterizedTest
16    @NullAndEmptySource
```

7 Exemples de questions de niveau 3 I

- Qu'est-ce que le principe de substitution de Liskov & Wing? Vous pouvez vous aider en prenant un exemple.

7 Exemples de questions de niveau 3 II

- En considérant que le code qui suit compile sans erreur, expliquez le résultat de l'exécution

```
1 public final class NomPrenom {
2     private String nom, prenom;
3     public NomPrenom(final String n, final String p) {this.nom=n;this.prenom
        =p;}
4     @Override public boolean equals(final Object obj) { // ...
5         if (obj == null) { return false; }
6         if (this == obj) { return true; }
7         if (!(obj instanceof NomPrenom hc)) { return false; }
8         return (nom.equals(hc.nom) && prenom.equals(hc.prenom)); }
9 }

1 Chercheur a = new Chercheur("nom", "prenom", "adresse");
2 NomPrenom b=new NomPrenom("nom","prenom"), c=new NomPrenom("nom","prenom
    ");
3 Map<NomPrenom, Chercheur> d = new HashMap<NomPrenom, Chercheur>(); d.put
    (c, a);
4 System.out.print(b.hashCode() + ",␣" + c.hashCode() + ",␣" + d.get(c).
    equals(d.get(b)));
```

Résultat de l'exécution : 366712642, 1829164700, false