

Matière	Programmation orientée objets
Enseignant	M. NAJIB
Date	31-10-2023
Durée	1h50

Documents non-autorisés

Course Outcomes (Syllabus) avec le pourcentage				
Course Outcomes	Evaluer	Exercice	Pourcentage	
CO1: Master the basic concepts of OOP	X	1	50%	
CO2: Master the visibility of attributes, methods, inheritance principles, polymorphism, abstract classes, and interfaces	X	2, 3	33%	
CO3: Develop ergonomic graphic interfaces and events programming using JAVA	-	-	-	

Exercice 1 : (3 pts)
Questions de cours :

1. Dans une relation d'héritage, une classe fille peut hériter les attributs protégés de la classe mère. (Vrai ou faux) (1pt)
2. Une classe Java peut hériter d'une interface. (Vrai ou faux) (1pt)
3. Expliquez le rôle de la machine virtuelle JAVA (JVM). (1pt)

Exercice 2 : (9 pts)

Pour le développement d'une application de gestion des projets, nous souhaitons utiliser les classes suivantes :

- **ProjetDevLogiciel** (*id* (entier auto-inc), *nom*, *budget* (double), *langageProg* (*string privé*), *tblDev* (tableau de 4 développeurs max (length = 4)))
- **Projet** (*id* (entier auto-inc), *nom*, *budget* (double))
- **ProjetMarketing** (*id* (entier auto-inc), *nom*, *budget* (double), *publicCible* (string), *responsable* (type Développeur))

Contraintes :

- **Id** du projet est un entier **auto-incrémente**
- On suppose que la classe **Développeur** caractérisée par (**idDev**, **nomDev**) est **donnée** avec les **constructeurs et littérateur** (**toString**).
- Vous pouvez utiliser les **listes** à la place des tableaux.
- Toutes les classes sont déclarées dans le même package

Questions :

1. Donnez un schéma qui illustre la relation d'héritage entre les trois classes projets. (Précisez les attributs de chaque classe). (2pts)

2. Donnez le code de la classe « **Projet** » avec **constructeur** full-paramètres, **getters** et **setters** (en cas de besoins) et la méthode **toString**. (2pts)
3. Donnez le code de la classe « **ProjetDevLogiciel** » avec **constructeur** full-paramètres, **getters** et **setters** (en cas de besoins) et la méthode **toString**. (3pts)
4. Dans la méthode Main, donnez le code nécessaire pour la création d'un objet de type « **ProjetDevLogiciel** ». Initialiser le tableau des développeurs de ce projet (**tblDev**) avec deux développeurs (hard-coded, sans l'utilisation des Scanners) (2pts)

Exercice 3 : (8 pts)

Nous supposons que tous les projets de développement logiciels sont enregistrés dans le tableau « **tProjDev** » déclarée dans la méthode main comme suite :

```
public class Test {
    public static void main (String[] args) {
        ProjetDevLogiciel[] tProjDev = new ProjetDevLogiciel[20];
    }
}
```

Questions : (les méthodes demandées sont déclarées dans la classe « **ProjetDevLogiciel.java** »)

1. Donnez le code de la méthode qui permet de récupérer un tableau (**tProjDev**) trié par ordre croissant du budget du projet. (2pt)

public ProjetDevLogiciel[] trierProjets(ProjetDevLogiciel[] tProjDev)

2. Donnez le code de la méthode qui permet de récupérer le nombre de développeurs affectés à un projet de développement logiciel. (2pts)

public int countDev()

3. Donnez le code de la méthode qui permet de supprimer les projets de développement logiciel sans développeurs affectés (0 développeur) et de récupérer le tableau des projets après suppression (2pts)

public ProjetDevLogiciel[] suppProjetVide(ProjetDevLogiciel[] tProjDev)

4. Donnez le code de la méthode qui permet d'afficher le nombre de projets auxquels participe chaque développeur, comme suit (2pt) : **public void bilanAffectation(ProjetDevLogiciel[] tProjDev)**

Id développeur	Nom	Nbr de projets
1	- Rachidi	- 2 projet(s)
2	- Taha	- 5 projet(s)

Bonne chance