

I. Compléter la classe Couleur, ci-dessous, en répondant aux questions suivantes : (11 points)

- 1) Compléter la méthode modifierCouleur((Couleur coul)) (ligne 12) pour qu'elle :
 - Affecte aux attributs codeCoul, nomCoul de l'instance courante, ceux de l'instance coul
 - Crée et retourne un objet ob de type Couleur avec les attributs de l'instance courante modifié.
- 2) Compléter la méthode saisie (ligne 18) permettant de donner des valeurs à a et b en utilisant la classe Scanner. Faire les importations nécessaires (ligne 2)
- 3) Définir le principe d'encapsulation en JAVA
.....
.....
.....
- 4) Ecrire les méthodes nécessaires permettant d'accéder et de modifier les attributs codeCoul et nomCoul dans tous les packages
- 5) Ecrire la méthode main dans la classe Couleur permettant de :
 - Appeler la méthode saisie
 - créer un objet co en donnant les valeurs a et b à ses attributs.
 - Afficher l'objet co
- 6) Qu'affiche l'instruction d'affichage dans la question 5?
.....
.....

Quelle est la méthode qui a permis d'obtenir ce résultat ? elle appartient à quelle classe ?
.....

- 7) Quelle est la méthode qu'il faut ajouter à la classe Couleur pour que System.out.println(co) affiche les attributs de co ? Ecrire cette méthode dans la classe.
.....

1. package packCoul ;
2.
3. **public class Couleur{**
4. private int codeCoul ;
5. private String nomCoul ;
6. public static int a;
7. public static String b;
8. **public Couleur(int c, String n){**
9. codeCoul=c;
10. nomCoul=n;
11. }

```

12. public Couleur modifierCouleur(Couleur coul){
13. ....
14. ....
15. ....
16. ....
17. }
18. public static void saisie() {
19. ....
20. ....
21. ....
22. }
23. ....
24. ....
25. ....
26. ....
27. ....
28. ....
29. ....
30. ....
31. ....
32. ....
33. ....
34. ....
35. ....
36. ....
37. ....
38. ....
39. ....
40. ....
41. ....
42. }

```

II. Compléter la classe Sac ci-dessous en répondant aux questions suivantes : (5.5 points)

- 1) Compléter la méthode colorerSac (ligne 7) en :
 - Affichant la couleur c de l'instance courante
 - Appeler la méthode saisie de la classe Couleur
 - Créer un la couleur c avec les valeurs de a et b utilisées dans la méthode saisie.
 - Appeler modifierCouleur pour modifier la couleur c

2) Est-ce que la méthode colorerSac peut être static ? justifier.

.....
.....

3) Quel est l'effet du modificateur de nonaccès final pour la méthode qualiteSac ?

.....
.....

4) Compléter la méthode qualiteSac pour qu'elle :

- affiche l'attribut nomCoul de l'attribut c de l'instance courante (ligne 15)
- modifie l'attribut codeCoul de l'attribut c de l'instance courante par 122 (ligne 16)

5) Déclarer une méthode public abstract float prixSac(float prixv) de type float (ligne 21).

6) Citer les caractéristiques d'une classe abstract

.....
.....

1. Package packSac ;

2. import packCoul.Couleur ;

3. **public abstract class Sac{**

4. Couleur c ;

5. **protected** int volume ;

6. **public Sac(){ }**

7. **protected void colorerSac(){**

8.

9.

10.

11.

12. **}**

13. **final void** qualiteSac(String qualite){

14. System.out.println("la qualité est :"+qualite) ;

15.

16.

17. **}**

18. **public static void** afficher(){

19. System.out.println("je suis une méthode static") ;

20. **}**

21.

22. **}**

III. (3.5 points)

1) Ecrire une classe concrète SacAMain dans un package fillesSac qui hérite de la classe Sac et faire le nécessaire pour que la classe SacAmain compile. Sachant que Le prix d'un sac=prix*volume.

.....

.....

.....

.....

.....

.....

2) Ecrire une méthode main qui permet de :

- créer un objet sM de type SacAMain
- afficher les attributs de sM qui lui sont accessibles
- Invoquer les méthodes accessibles par sM
- Invoquer les méthodes static accessibles de la classe Sac

.....

.....

.....

.....

.....

.....

.....

.....