

Contrôle continu - 2h

Cet énoncé comporte 4 pages.

Aucun document autorisé. Les téléphones portables et autres smartphones doivent être éteints et rangés. Le barème est donné à titre indicatif.

1. Questions de cours (6 pts)

Question 1. Sélectionnez les affirmations vraies

1. Une interface peut étendre (mot-clé extends) une classe abstraite.
2. Une interface peut étendre (mot-clé extends) plusieurs interfaces.
3. Une classe abstraite peut étendre (mot-clé extends) une classe concrète.
4. Super est une référence vers la superclasse.

Question 2. Parfois une méthode d'instance a besoin d'accéder à l'objet auquel elle appartient. Quel mot-clé permet de faire cela ?

1. Le nom de sa classe ;
2. super ;
3. this ;
4. self.

Question 3. Quelles sont les deux plus importantes méthodes héritées de la classe `Object` que toute classe métier doit redéfinir ? Pourquoi ?

Question 4. Dans quelles situations est-on obligé de déclarer une classe abstraite ?

1. Quand la classe déclare une méthode abstraite.
2. Quand la classe hérite une méthode abstraite sans l'implanter.
3. Quand la classe hérite une méthode abstraite et l'implante.
4. Quand la classe ne déclare pas d'attributs.

Question 5. Quel modificateur d'accès limite la visibilité d'un membre aux autres classes du même paquetage, et à ses sous-classes, quelque soit leur paquetage ?

1. public
2. (le modificateur par défaut)
3. protected
4. private

2. Analyse de code (6 pts)

Question 6. Qu'affiche le programme suivant ?

Listing 1 – Classe AA

```
1 public class AA {
2     static String [] text = {"A", "B", "C"};
3     public static void loop() {
4         for (int i = 2; i<=3; i++)
5             System.out.println(text[i]);
6     }
7     public static void main(String [] args) { loop(); }
8 }
```

Question 7. Dire si le programme suivant génère ou non une erreur à la compilation ou à l'exécution et la raison de cette erreur. Sinon, indiquer ce qu'il affichera.

Listing 2 – Classe CC

```
1 public class CC {
2     int i;
3     public CC(int i) {
4         i = i;
5         System.out.println(i);
6     }
7     public static void main(String [] args) {
8         CC cc = new CC(10);
9         System.out.println(cc.i);
10    }
11 }
```

Question 8. Considérer les programmes suivants. Puis, pour chacun des items numérotés listés après, dire si les instructions de l'item génèrent une erreur à la compilation ou à l'exécution et la raison de cette erreur. Sinon indiquer ce qui sera affiché à leur exécution.

Listing 3 – Classes R, S et T

```
1 public interface T {
2     void a();
3 }
4 public class S implements T {
5     public void a() { System.out.println("S.a"); }
6     public void b() { System.out.println("S.b"); }
7 }
8 public class R extends S {
9     public void a(){ System.out.println("R.a"); }
10    public void c() { System.out.println("R.c"); }
11 }
```

1. T t1 = new R(); S s2 = (S) t1; s2.a();
2. R r1 = new R(); S s3 = r1; s3.b();
3. T t2 = new S(); R r2 = (R) t2; r2.b();
4. S s4 = new S(); T t3 = (T) s4; R r3 = (R) t3; r3.c();
5. S s5 = new T(); R r4 = (R) s5; r4.a();

Question 9. Que se passera-t-il lorsque vous essayerez de compiler et d'exécuter le programme suivant, sachant que le fichier `Hello.txt` n'existe pas dans le répertoire courant ?

Listing 4 – Classe Worker

```
1 import java.io.*; //c'est pour gagner de la place :-)
2 public class Worker {
3     public static void main(String argv[]) {
4         Worker w = new Worker();
5         System.out.println(w.work());
6     }
7     public int work() {
8         try {
9             FileInputStream dis = new FileInputStream("Hello.txt");
10        } catch (FileNotFoundException fne) {
11            System.out.println("No such file found");
12            return -1;
13        } catch (IOException ioe) {
14        } finally {
15            System.out.println("Doing finally");
16        }
17        return 0;
18    }
19 }
```

3. Gestion de la billetterie d'un parc d'attraction (8 pts)

On souhaite développer une application de gestion de la billetterie d'un parc d'attraction. Les billets sont vendus selon les catégories de tarifs suivantes :

1. tarif normal individuel journalier avec accès illimité à toutes les attractions du parc ;
2. tarif normal individuel journalier avec accès illimité à 5 attractions seulement du parc ;
3. tarif réduit (familles nombreuses, étudiants,...) avec accès illimité à toutes les attractions, correspondant à 70% du premier tarif normal ;
4. tarif réduit avec accès illimité à 5 attractions du parc, correspondant à 70% du deuxième tarif normal.

3.1. Classe ParcAttraction

La classe devant servir à gérer la billetterie s'appellera `ParcAttraction`. Elle est caractérisée par :

- son nom, contenu dans une chaîne de caractères,
- sa capacité d'accueil, donnée par la valeur d'un entier,
- les prix unitaires des billets à tarif normal (réels),
- les nombres de billets vendus de chaque catégorie de tarif (entiers).

Le nom, la capacité d'accueil ainsi que les prix unitaires des billets sont fixés à la création d'un nouvel objet de type `ParcAttraction`. Il n'est plus possible de les modifier après la création de l'objet.

La classe `ParcAttraction` expose les opérations suivantes :

- ▶ `int nbPlacesDispo()` - renvoie la capacité d'accueil restante du parc.
- ▶ `void vendreBillets(int nb, Tarif tarif)` - permet de vendre des billets, où `nb` exprime la demande en nombre de places et `tarif` le type de tarif demandé. Le type `Tarif` de ce paramètre est une énumération.

La vente n'est pas effectuée si la capacité d'accueil restante ne permet pas de satisfaire la demande (exprimée dans `nb`). Un message doit afficher cette impossibilité.

- ▶ `void reinitialiser()` - remet à zéro tous les compteurs de billets.

- `double chiffreAffaires()` - calcule et renvoie le chiffre d'affaire produit par le parc depuis la mise en service de l'objet correspondant ou la dernière réinitialisation.
- `double tauxRemplissage()` - renvoie le taux (en pourcentage) de remplissage du parc.
- `String toString()` - renvoie une représentation sous forme de chaîne de caractères de l'objet courant de type `ParcAttraction`. Un exemple :

```
Parc MiageWonderland,  
Capacité d'accueil : 180,  
Tarif normal illimité : 38.50 euros,  
Tarif normal illimité 5 : 28.50 euros,  
24 billets vendues au tarif illimité,  
19 billets vendues au tarif illimité 5,  
32 billets vendues au tarif réduit,  
23 billets vendues au tarif réduit 5.
```

3.2. Classe Tarif

Tarif est une énumération des différentes catégories de tarif : illimité, illimité 5, réduit, réduit 5.

Question 10. Définissez les classes <code>ParcAttraction</code> et <code>Tarif</code> .
--