

Ecole Supérieure Privée Technologies & Ingénierie

Type d'épreuve : ☐ Devoir ☒ Examen SESSION PRINCIPALE
Enseignant : Mr Slah BOUHARI
Matière : Programmation JAVA
Année Universitaire : 2023-2024 Semestre : 2
Classe : TIC-1-O OLAB
Documents : ☐ Autorisés ☒ Non autorisés
Date : 19/01/2024 Durée : 1h30mn
Nombre de pages : 6

Questions QCM (8 pts):

1. Soit le code suivant :

```
1. public class A {  
2. int add(int i, int j){  
3. return i+j;  
4. }  
5. }  
6. public class B extends A{  
7. public static void main(String argv[]){  
8. short s = 9;  
9. System.out.println(add(s,6));  
10. }  
11. }
```

Trouver la bonne réponse :

- a) Compile fail due to error on line no 2
- b) Compile fail due to error on line no 9
- c) Compile fail due to error on line no 8
- d) 15

2. Quel est le résultat d'affichage du code suivant :

```
public class A {  
    int k;  
    boolean istrue;  
    static int p;  
    public void printValue() {  
        System.out.print(k);  
        System.out.print(istrue);  
        System.out.print(p);  
    }  
}  
  
public class Test{  
    public static void main(String argv[]){  
        A a = new A();  
        a.printValue();  
    }  
}
```

Trouver la bonne réponse :

- a) 0 false 0
- b) 0 true 0
- c) 0 0 0
- d) Compile error - static variable must be initialized before use.

3. On considère le code suivant :

```
class F {
    void f() {
        System.out.print("F.f() "); this.g();
    }
    void g() {
        System.out.print("F.g() ");
    }
}

class Fbis extends F{
    void f() {
        System.out.print("Fbis.f() ");this.g();
    }
    void g() {
        System.out.print("Fbis.g() "); super.f();
    }
}

class Main{
    public static void main(String[] args){
        F f = new Fbis();
        ((F)f).f();
    }
}
```

Trouver la bonne réponse :

- a) il ne peut pas être compilé
- b) il boucle à l'exécution (boucle infinie)
- c) sa sortie sera: F.f() F.g()
- d) sa sortie sera: Fbis.f() Fbis.g() F.f() F.g()

4. On considère le code suivant :

```
class A{
    String s="A";
    void f(String s1){
        System.out.println(s+" "+s1);
    }
}

class B extends A{
    String s="B";
    void f(String s1){
        super.f(s+" "+s1);
    }
}

class C extends B{
    String s="C";
    void f(String s1){
        super.f(s+" "+s1);
    }
}
```

```
} }
```

Que réalise le code suivant : `A a = new C(); a.f("X");`

- a) Provoque une erreur à la compilation
- b) Affiche C C C X
- c) Affiche A B C X
- d) Affiche A X

5. Parmi les méthodes suivantes de l'API Stream, laquelle qui n'est pas une opération finale :

- a) `forEach`
- b) `map`
- c) `findFirst`
- d) `collect`

6. On considère le code suivant :

```
List<String> words = Arrays.asList("API", "Stream", "for",  
                                "Programmation Java", "JAVA Quiz").  
  
Optional<String> MyString = words.stream()  
                                .reduce((word1, word2)  
                                -> word1.length() > word2.length() ? word1 : word2);  
  
MyString.ifPresent(System.out::println);
```

Trouver le résultat de son exécution :

- a) Stream
- b) Programmation Java
- c) JAVA Quiz
- d) API
- e) [API, for]

7. Quel sera le résultat d'exécution de ce code :

```
Stream<String> stream = Stream.of("a", "b", "c");  
  
Optional<String> result = stream.filter(s -> s.equals("d")).findFirst();
```

- a) "d"
- b) "a"
- c) `Optional.Empty`
- d) `null`

8. Quel sera le résultat d'exécution de ce code :

```
int Number = IntStream.range(2, 6)
    .reduce((num1, num2) -> num1 * num2)
    .orElse(-1);

System.out.println("The number is : " + Number);
```

Trouver la bonne réponse :

- a) 48
- b) 20
- c) 120
- d) 720

Problème 1 (12 pts):

On cherche à représenter des albums de musique caractérisés par les pistes audio (audio tracks) qui les composent.

La classe `Track` permet de définir les données sur les pistes audio. Un objet `Track` est défini par son titre (`title`) et sa durée (`duration`) en nombre de secondes.

La classe `Track` (appartient au paquetage `music.data.`) est définie par le diagramme :

music::data::Track
- title : String - duration : int
+ Track(title : String, duration : int) + getDuration() : int + getTitle(): String

1. Donnez le code complet de la classe `Track` conforme au diagramme ci-dessus et en utilisant les données suivantes :
 - Une méthode `equals` sachant que deux pistes sont considérées égales si elles ont *même titre* et *même durée*.
 - Une méthode `toString` dont le résultat reprendra le titre et la durée de la piste séparée par un espace.

La classe `Album` permet de représenter les albums de musique.

Un album de musique est défini par **son titre** (`title`), représenté par une chaîne de caractères, **son genre musical** (`category`) représenté par une chaîne de caractères, et les pistes audios qui le composent.

Les albums sont classés par catégorie de genres musicaux. On suppose qu'il n'existe que 4 genres musicaux `rock`, `rap`, `pop`, `classique` (par exemple).

Le titre et la catégorie et **le nombre de pistes** composant un album sont fournis à la création de cet album. On utilise un tableau dynamique pour mémoriser les pistes d'un album. Le nombre de pistes et une constante qui sera défini à la création de l'album.

Les pistes sont ajoutées une par une à un album.

Lorsque toutes les pistes ont été ajoutées, l'album est dit terminé (`finished`) (et il ne l'est pas dans le cas contraire).

2. Donnez le code de la classe sachant qu'elle possède :

- Une méthode `isFinished()` dont le résultat est vrai si et seulement si l'album est terminé.
- Une méthode `isTrueTrack(Track T)` dont le résultat est vrai si et seulement si la piste donnée en paramètre est égale à l'une des pistes de l'album.
- Une méthode `addTrack(Track T)` permet d'ajouter une piste à un album.

On supposera que l'on n'ajoute jamais deux pistes égales dans un même album. Si un tel cas se présente cette méthode déclenche une exception `AlbumException`.

Si l'album est terminé au moment de l'ajout, cette méthode déclenche une exception `AlbumException`.

Les messages de l'exception doivent être personnalisés à chaque cas.

- Une méthode `totalDuration()` qui retourne la durée totale d'un album terminé, cette méthode retourne -1 si l'album n'est pas terminé.
- Une méthode `equals()` : On dit que deux albums sont égaux s'ils contiennent les mêmes pistes, mais celles-ci peuvent être dans des ordres différents dans les deux albums.

3. Ecrire un programme principal pour tester les classes implémentées.

Créer deux pistes `p1` et `p2`, respectivement de titre "`t1`" et "`t2`" et de durée 230 et 145.

Initialiser une référence `d` avec un objet `Album` de titre "`td`", dont le genre musical est `rap` et composé de 2 pistes,

Ajouter à l'album, dans l'ordre, les pistes `p1` et `p2` puis affiche la durée totale de l'album.