

1) Quelle est la différence entre une classe abstraite et une interface en java ?

une classe abstraite est une classe non instanciable, qui contient déjà du code, et tout particulièrement des méthodes non implémentées (que la classe qui en héritera devra implémenter), ou des méthodes déjà implémentées (factorisation de code, facilité de maintenance).

L'interface est assez proche de la classe abstraite, mais aucune méthode n'est implémentée dedans. Il y a juste la déclaration (la signature) des méthodes. Ainsi, une classe qui implémente (on ne dit pas hériter) une interface, devra obligatoirement implémenter les méthodes.

2) Quelle est la différence entre une collection de type **ArrayList** et une collection de type **LinkedList** ?

ArrayList stocke les éléments dans un tableau dynamique, ce qui signifie que les éléments sont stockés dans une structure de tableau de taille variable qui peut être agrandie ou réduite dynamiquement. En revanche, LinkedList stocke les éléments sous forme de nœuds, où chaque nœud contient une référence à l'élément suivant dans la liste.

- L'accès aux éléments d'un ArrayList est plus rapide que celui d'une LinkedList.
- l'insertion et la suppression d'éléments sont plus rapides dans une LinkedList que dans un ArrayList.

3) Définir l'**upcasting** en java.

A une référence déclarée d'une classe mère, il est possible d'affecter une référence vers un objet de sa sous classe.

4) Quelle est la différence entre la **Surcharge** et la **Redéfinition** des methodes.

- Surcharge : possibilité de définir des méthodes possédant le même nom mais dont les arguments différents.
- Redéfinition : lorsque la sous-classe redéfinit une méthode de la classe mère dont le nom, les paramètres et le type de retour sont identiques.

5) Dans la gestion des Exceptions, peut on ajouter plusieurs blocs **catch** à un bloc **try** ? Justifier votre réponse.

Oui, on peut ajouter plusieurs blocs catch à un bloc try si les blocs catch capture des exceptions différents (des exceptions personnalisé ou des exceptions qui hérite de la classe mère Exception).

6) Soit deux interfaces java I1 et I2 qui contiennent la même méthode : `int compter (String)`. Une classe C implémente les deux interfaces. De quelle interface la méthode `compter` sera défini en C ? justifier votre réponse.

La méthode "compter(String)" de la classe C sera définie en C par rapport à la première interface que C implémente.

L'ordre dans lequel les interfaces sont déclarées dans la liste d'implémentation de la classe est important car il détermine la priorité de l'interface.

7) Quelle est la différence entre les mots de java : throw et throws ?

- "throw" est un mot-clé qui est utilisé pour générer une exception explicite dans un bloc de code.
- "throws" est un mot-clé utilisé dans la signature de méthode pour déclarer les exceptions potentielles que la méthode peut lancer.

8) Peut-on avoir un bloc **try** sans le bloc **catch** en java ? Justifier votre réponse.

Oui, il est tout à fait possible d'avoir un bloc try sans bloc catch en Java, si on inclut un bloc finally.

9) Définir l'ergonomie.

L'ergonomie vise à concevoir des outils et machines pour une utilisation confortable, sûre et efficace.

10) Citer les principales règles d'ergonomie des interfaces graphiques.

- Les formulaires doivent être brefs.
- Pas de désordre.
- Les saisies doivent être assistées autant que possible.
- Les couleurs doivent avoir un sens et ne pas être utilisées de manière excessive.

11) Définir une API.

Une API en Java est un ensemble de classes regroupées par leur proximité fonctionnelle dans un paquetage ou un ensemble de paquetages.

12) Quelle est la différence entre JFrame et JWindow ?

La classe Jwindow permet de créer des fenêtres sans bordure ni bouton, utilisées généralement pour des écrans d'attente.

Par contre, JFrame représente une fenêtre avec une bordure et des boutons de fermeture, d'agrandissement et de réduction.

13) Les boîtes de dialogues :

- Simple boîte :
`JOptionPane.showMessageDialog(frame, "Un message");`

- Boite d'avertissement :
`JOptionPane.showMessageDialog(frame, "Titre", "message",
JOptionPane.WARNING_MESSAGE);`
- Boite de question/confirmation :
`JOptionPane.showConfirmDialog(null, "Do you like bacon?");`
// par défaut, `JOptionPane.YES_NO_CANCEL_OPTION`
// `YES_NO_OPTION`, `OK_CANCEL_OPTION`
// type de retour est un entier : 0 pour `JOptionPane.NO_OPTION`
- Boite de saisie :
`JOptionPane.showInputDialog(frame, "Entrez votre username");`
//type de retour est un `String`

14) Quel est le rôle des adaptateurs dans **swing** ?

Les adaptateurs d'interfaces implémentent toutes les méthodes sans accomplir d'action pour éviter de les citer dans le programme.

15) Quelle est la différence entre `JWindowListener` et `JActionListener` ?

- `JWindowListener` est une interface pour gérer les événements de fenêtres.
- L'interface `JActionListener` permet de gérer les événements liés aux actions utilisateur.

16) Expliquer l'utilité de la classe `ResultSetMetaData`

La classe `ResultSetMetaData` fournit des informations sur les colonnes d'une table, telles que le nombre de colonnes et les noms des colonnes.