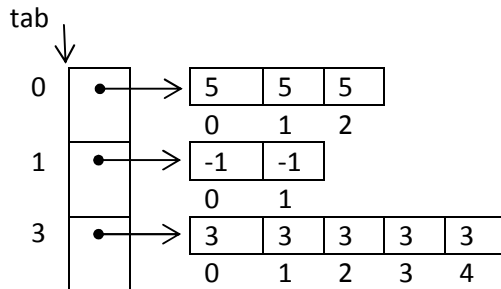


Examen POO

Durée : 1h30 min

Questions de cours

1. Quelle est la différence entre la visibilité **protected** et la visibilité par défaut (sans utiliser aucun modificateur de visibilité).
2. Ecrire les instructions java permettant de remplir le tableau **tab** en utilisant la méthode **Arrays.fill(..)**



3. Qu'affiche le programme suivant :

```

class A{
    int a;
    A(){ System.out.println("Aa="+a);}
}

class B extends A{
    B(int b) { a=b; System.out.println("a="+a);}
}

public class testH {
    public static void main(String[] args){
        B obj=new B(2);
    }
}
  
```

Remarque

Dans l'exercice suivant, on suppose que la classe **String** (chaîne de caractère) dispose de la méthode **equals** définie comme suit : **boolean equals(String s)** qui détermine si deux chaînes de caractère ont la même valeur. Exemple d'utilisation de **equals** :

```

String str= "Mohamed";      boolean test1=str.equals("Mohamed");      //test1<-- true
                             boolean test2=str.equals("Omar");      //test2<-- false
  
```

Exercice

On veut créer un petit système de suivi des surveillances des examens. Pour cela, on crée trois classes : **Enseignant**, **Surveillance**, **PlanningSurveillance**

1. Réaliser la classe **Enseignant** définie par les membres suivants :
 - l'attribut **privé dep** de type entier représentant le département
 - l'attribut **privé nom** de type String représentant le nom et prénom de l'enseignant
 - un constructeur avec deux paramètres pour initialiser **dep** et **nom**
 - les méthodes qui permettent de modifier et lire les attributs **dep** et **nom** à l'extérieur de la classe **Enseignant**

2. réaliser la classe **Surveillance** définie par les membres suivants :

- l'attribut privé **numSeance** de type entier représentant le n°de séance dans la journée
- l'attribut privé **date** de type String représentant la date de la surveillance.
- l'attribut privé **module** de type String représentant le module concerné par la surveillance.
- l'attribut privé **tabEnseig** est un tableau de **Enseignant** représentant la liste des enseignants surveillants. Une surveillance est assurée par 04 enseignants au maximum.
- l'attribut privé **nbreEnseig** de type entier représentant le nombre d'enseignants dans une surveillance
- Le constructeur **public Surveillance(int numSeance, String date, String module)**
- les méthodes qui permettent de modifier et lire les attributs **tabEnseig** et **module** à l'extérieur de la classe **Surveillance**
- La méthode **public void ajouterEnseignant(Enseignant e)** permettant d'ajouter un enseignant **e** dans la liste des enseignants surveillants

3. réaliser la classe **PlanningSurveillance** définie par les membres suivants :

- l'attribut privé **nbreSurv** de type entier représentant le nombre de surveillances dans le planning
- l'attribut privé **tabSurveillance** est un tableau de **Surveillance** représentant la liste des surveillances dans le planning. Un planning contient 10surveillances au maximum.
- Un constructeur sans paramètres pour initialiser **tabSurveillance**
- La méthode **public void ajouterSurveillance(int numSeance,String date, String module)** permettant d'ajouter une surveillance dans la liste des surveillances du planning. On suppose qu'un module ne se répète pas dans un planning.
- La méthode **public int nbreSurvParEnseig(String e)** permettant de retourner le nombre de surveillances assurées par un enseignant **e**
- La méthode **public boolean existeDansSurv(String ens, String module)** permettant de vérifier si un enseignant **ens** assure une surveillance.