

JIPST-CNAM
Programmation JAVA
NFA 031
Mercredi 8 Février 2017

Avec document
Durée : **2 h30**
Enseignant : LAFORGUE Jacques

1^{ère} Session NFA 031

L'examen se déroule en deux parties. Une première partie de 1h15mn, sans document, consacrée à des questions de cours, et une deuxième partie, avec document, consacrée à la réalisation de programmes Java.

Au bout de 1h15mn, les copies de la première partie seront ramassées avant de commencer la deuxième partie.

Pour la première partie, vous devez rendre le QCM rempli et les réponses aux questions libres écrites sur des copies vierges.

Pour la deuxième partie, vous écrivez vos programmes sur des copies vierges. Vous devez écrire le code en Java.

1^{ère} PARTIE : COURS (sans document)

1h15mn

1. QCM (35 points)

Mode d'emploi :

Ce sujet est un QCM dont les questions sont de 3 natures :

- **les questions à 2 propositions**: dans ce cas une seule des 2 propositions est bonne.
 - +1 pour la réponse bonne
 - -1 pour la réponse fausse
- **les questions à 3 propositions** dont 1 seule proposition est bonne
 - + 1 pour la réponse bonne
 - -½ pour chaque réponse fausse
- **les questions à 3 propositions** dont 1 seule proposition est fausse
 - + ½ pour chaque réponse bonne
 - -1 pour la réponse fausse

Il s'agit de faire une croix dans les cases de droite en face des propositions.

On peut remarquer que cocher toutes les propositions d'une question revient à ne rien cocher du tout (égal à 0).

Si vous devez raturer une croix, faites-le correctement afin qu'il n'y ait aucune ambiguïté.

N'oubliez pas d'inscrire en en-tête du QCM, votre nom et prénom.

Vous avez droit à **4 points** négatifs sans pénalité.

NOM:	PRENOM:
------	---------

Dans un Langage Orienté Objet, une classe est la déclaration d'un type de donnée structurée qui contient également des traitements informatiques qui permettent de modifier ce type de donnée		Q 1.
1	OUI	
2	NON	

Dans un langage orienté objet, un objet représente un état d'information. Cet état est constitué de :		Q 2.
1	l'ensemble des attributs de l'objet	
2	uniquement l'ensemble des attributs publiques de l'objet	
3	uniquement l'ensemble des attributs privés de l'objet	

Dans un langage orienté objet, un objet peut être composé d'autres objets qui peuvent être eux-mêmes composés de sous-objets.		Q 3.
1	OUI	
2	NON	

En JAVA, on doit soi-même s'occuper de la destruction des objets qui ne sont plus utilisés.		Q 4.
1	OUI	
2	NON	

En JAVA, pour compiler un programme (ensemble de plusieurs fichiers .java et donc constitué de plusieurs classes), il suffit de compiler la classe principale (contenant la méthode main)		Q 5.
1	OUI	
2	NON	

Soit le fichier suivant le fichier CodeJava\exercices\cnam\util\Terminal.java. Le fichier Terminal.java contient en 1 ^{ère} ligne : package exercices.cnam.util; Dans CodeJava\programme se trouve le fichier Prog.java suivant: <pre> import exercices.cnam.util.*; public class Prog { public static void main(String... args) { Terminal.ecrireStringln("Bonjour"); } } </pre> On est dans le répertoire CodeJava/programme, et on veut compiler le programme. Quelle(s) commande(s) est(sont) valide(s) :		Q 6.
1	javac -classpath "." Prog.java	
2	javac Prog.java	
3	javac -classpath ".." Prog.java	

Le compilateur Java (javac) permet de créer un ensemble de fichier .class qui sont ensuite interprétés par une JVM		Q 7.
1	OUI	
2	NON	

En Java, l'exécution d'un programme peut se faire d'autant de façons différentes qu'il existe de méthode main dans les classes du programme		Q 8.
1	OUI	
2	NON	

La commande java		Q 9.
1	prend en entrée un fichier .java afin de l'interpréter	
2	prend en entrée un fichier .class afin de l'interpréter	
3	exécute la méthode main de la classe java contenue dans le fichier .class qui est en entrée de la commande	

En JAVA, les méthodes déclarées en dehors d'une classe sont appelées des méthodes statiques		Q 10.
1	OUI	
2	NON	

En JAVA, une méthode statique d'une classe peut utiliser les attributs non statiques de la même classe:		Q 11.
1	OUI	
2	NON	

Soit le code JAVA suivant :		Q 12.
<pre> public class C { private int attribut; public C(){attribut=10;} public static void main(String[] args) { C objet = new C(); objet.attribut = Integer.parseInt(args[0]); // LIGNE 7 } } </pre>		
Le code de la LIGNE 7 est correct		
1	OUI	
2	NON	

Les méthodes public de la classe A		Q 13.
1	peuvent utiliser directement les attributs privés de la classe A	
2	peuvent utiliser directement les attributs privés d'une autre classe B	
3	peuvent utiliser directement les attributs publiques de la classe A	

Soit la classe C1 dont tous les attributs sont <u>privés et statiques</u> . Soit la classe C2 appartenant à un autre package que C1. Dans ce cas, les méthodes de C2 peuvent accéder aux attributs de C1		Q 14.
1	OUI	
2	NON	

Soit le code JAVA suivant :		Q 15.
<pre> public class Exemple { ????? int var_x; public static void main(String a_args[]) { var_x = Integer.parseInt(a_args[0]); System.out.println("X=" + var_x); } } </pre>		
On exécute ce programme avec la commande : java Exemple 123		
Pour que ce code soit correct, il faut remplacer ????? par :		
1	private	
2	public	
3	static	

Soit le code JAVA suivant :		Q 16.
<pre> public class Bidon { public int valeur; public Bidon(int v){valeur=v;} } public class Exemple{ public static void main(String[] args){ Bidon obj = new Bidon(12); traitement(obj,100) System.out.println(obj.valeur); } static void traitement(Bidon bid,int v){ bid.valeur=v; } } </pre>		
L'exécution de ce programme est :		
1	affiche la valeur 100	
2	affiche la valeur 12	

En JAVA, une méthode privée (private) d'une classe a accès aux attributs publics d'une autre classe		Q 17.
1	OUI	
2	NON	

En JAVA, pour créer un package, il suffit de :		Q 18.
1	préfixer le nom d'un répertoire par le mot "package" et de mettre dans ce répertoire les fichiers java contenant les classes devant appartenir au package	
2	créer un répertoire de nom quelconque et de mettre dans ce répertoire les fichiers java contenant les classes devant appartenir au package	
	créer un répertoire de nom quelconque, de mettre dans ce répertoire les fichiers java contenant les classes devant appartenir au package et de déclarer le nom du package en entête de chacun des fichiers java appartenant à ce répertoire	

En JAVA, dans une classe, il faut toujours créer un constructeur sauf si la classe contient que des méthodes statiques.		Q 19.
1	OUI	
2	NON	

Dans la programmation objet, en JAVA, le rôle du constructeur d'une classe est de :		Q 20.
1	créer un objet en mémoire de la JVM et d'initialiser les attributs	
2	créer une classe en mémoire de la JVM	

En JAVA, dans un constructeur il est possible d'initialiser les attributs privés de la classe mais pas les attributs publics de la classe		Q 21.
1	OUI	
2	NON	

Soit la classe suivante :		Q 22.
<pre>public class Truc{ private int x; public Truc(int x){ this.x = x; } public Truc(){ this.x = -1; } }</pre>		
L'instruction suivante :		
<pre>Truc chose = new Truc();</pre> est valide et la valeur de x de chose a la valeur -1		
1	OUI	
2	NON	

Soit la classe suivante :		Q 23.
<pre>public class Truc{ public int x; public static Truc Truc(){ Truc c=new Truc();return c; } public Truc(){ this.x = -1; } }</pre>		
L'instruction suivante :		
<pre>Truc chose = Truc.Truc();</pre> est valide		
1	OUI	
2	NON	

Quand une classe définit un constructeur avec des paramètres alors le constructeur par défaut de Java (sans paramètre) est accessible		Q 24.
1	OUI	
2	NON	

Soit la classe suivante :		Q 25.
<pre>public class A{ private ArrayList<String> liste; public A(){ ArrayList<String> liste = new ArrayList<String>(); } public void add(String s){liste.add(s);} }</pre>		
Un programme réalise le code suivant:		
<pre>A exemple = new A(); exemple.add("LAFONT");</pre>		
Le code du programme s'exécute normalement :		
1	OUI	
2	NON	

En JAVA, le type de retour d'une méthode est soit void, soit un type primitif, soit la référence d'un objet		Q 26.
1	OUI	
2	NON	

Soit le code suivant : <pre>int tab_int[] = new int[10]; [...]</pre> <pre>for(int i=0; i< A ;i++) Terminal.ecrireIntln(B);</pre> Ce code affiche tous les éléments du tableau tab_int. A et B peuvent être remplacés par :		Q 27.
1	A → tab_int.length B → tab_int[i]	
2	A → tab_int.size() B → tab_int.get(i)	

En JAVA, pour qu'un objet puisse être passé en paramètre d'une méthode, il faut qu'il soit différent de null		Q 28.
1	OUI	
2	NON	

En JAVA, il est possible de créer des tableaux à 4 dimensions.		Q 29.
1	OUI	
2	NON	

Soit le code suivant : <pre>String str1[] = new StringTokenizer("AA;BB;CC", ";"); for(String s: str1) System.out.println(s);</pre> Ce code ::		Q 30.
1	affiche : AA BB CC	
2	n'affiche rien	
3	ne se compile pas correctement	

En Java, il est possible de modifier le contenu d'un tableau passé en paramètre d'une méthode		Q 31.
1	OUI	
2	NON	

En Java, un tableau [] ne peut contenir que des données de type primitif (int, double, ...) alors qu'un ArrayList peut contenir aussi des objets		Q 32.
1	OUI	
2	NON	

Soit le code JAVA suivant :		Q 33.
<pre>String[] tab = new String[2]; tab[0]="UN"; tab[1]="DEUX"; for(String s : tab){System.out.println(s);}</pre>		
1	Ce code affiche : UN DEUX	
2	Ce code affiche : UN DEUX null null	

En Java augmenter la taille physique d'un attribut de type tableau est possible dans la mesure où il est toujours possible changer la valeur de l'attribut avec un nouveau tableau d'une taille supérieure		Q 34.
1	OUI	
2	NON	

Ce code est correct :		Q 35.
<pre>ArrayList<String> tab = new ArrayList<String>(); tab.array[0] = "LAFONT"; tab.array[1] = "DUPONT"; tab.size = 2;</pre>		
1	OUI	
2	NON	

2. Questions libres (15 points)

Chaque question est notée sur 5 points.

Vous répondez à ces questions sur une **copie vierge** en mettant bien le numéro de la question, sans oublier votre nom et prénom.

Q 1

Faire la comparaison de l'utilisation des tableaux Java [] et des ArrayList() : déclaration, type des éléments, création, utilisation,...

Q 2

- Expliquez pourquoi, il est fortement conseillé de déclarer les attributs privés.
- Expliquez pourquoi, il ne faut pas abuser des attributs et des méthodes statiques.
- Expliquez pourquoi, on a besoin de créer des méthodes privées.

Q 3

Décrivez précisément ce qu'est une méthode en Java (de quoi elle est composée, de son rôle, de ses natures, ...)

Fin de la 1^{ère} partie

2^{ème} PARTIE : PROGRAMMATION (avec document)**1h15mn**

Problème 1 [15 points]

Soit les classes **Site** et **Produit** vues dans le cadre du projet de cette année (voir l'annexe ci-après).

Ecrire le code de la méthode de la classe **Site** suivante :

```
public Produit produitsMoinsChere(double minPrix, double  
maxPrix, String recherche )
```

Cette méthode retourne le Produit le moins cher parmi les produits dont le prix est compris entre [*minPrix*, *maxPrix*] et dont le *nom* contient *recherche*; retourne null si le produit n'a pas été trouvé.

Problème 2 [35 points]

On se propose de créer un site (classe **Site**) qui gère des réservations (classe **Reservation**) de chambre d'hôtel.

Le fichier texte, data/Reservations.txt, suivant contient des réservations permettant d'initialiser le site (exemple) :

```
172;22/12/2016;3/1/2017;2  
238;1/3/2017;5/3/2017;1  
172;6/1/2017;13/1/2017;1
```

Chaque ligne de ce fichier correspond à une **Reservation** et contient :

- le numéro de la chambre. Numéro de 1 à 121.
- la date de début de la réservation
- la date incluse de fin de la réservation
- nombre de personne

Ecrire les classes **Site** et **Reservation** (ne pas écrire les getteur, setteur) afin de :

- lire le fichier data/Reservations.txt (dans le constructeur de la classe Site)
- écrire la méthode de la classe Site qui permet d'ajouter une demande de réservation :

```
public String ajouterReservation(int numeroChambre,  
String dateDebut, String dateFin, int nbPersonne)
```

La méthode retourne toutes les réservations sous la forme d'une chaîne de caractère.

- écrire la méthode de la classe Site qui retourne sous la forme d'une chaîne de caractère tous les créneaux occupés d'une chambre donnée :

```
public String reservationsChambre(int numeroChambre)
```

Si le **numeroChambre** n'existe pas retourne "La chambre n'est pas réservée"

- écrire la méthode de la classe Site qui valide si une demande de réservation est valide.

```
public boolean valide(int numeroChambre, String  
dateDebut, String dateFin)
```

La méthode retourne vrai si l'intervalle [dateDebut, dateFin] n'est pas en intersection avec une des réservations de la chambre de numeroChambre.

Pour cela vous utiliser, la méthode **DateString int jours(String date)** qui retourne le nombre de jours entre 1/1/1970 et date.

Annexe

Pour rappel (le code de ma correction) :

```
public class Site
{
    private ArrayList<Produit> stock;          // Les produits du stock
    private ArrayList<Commande> commandes;    // Les bons de commande

    public Site()
    {
        // Initialisation des attributs par default
        //
        stock = new ArrayList<Produit>();
        commandes = new ArrayList<Commande>();

        // lecture du fichier data/Produits.txt
        // pour chaque ligne on cree un Produit que l'on ajoute au stock
        initialiserStock("data/Produits.txt");

        // lecture du fichier data/Commandes.txt
        // pour chaque ligne on cree une commande ou on ajoute une reference
        // d'un produit (ref_produit) a une commande existante.
        //
        initialiserCommandes("data/Commandes.txt");
    }

    [...]
}
```

```
public class Produit
{
    // Les caracteristiques d'un Produit
    //
    private String reference;          // reference du produit
    private String nom;               // nom du produit
    private double prix;              // prix du produit
    private int quantite;             // quantité du produit

    // Constructeur
    //
    public Produit(String reference,
                  String nom,
                  double prix,
                  int quantite)
    {
        this.reference = reference;
        this.nom = nom;
        this.prix = prix;
        this.quantite = quantite;
    }

    public String getReference(){return reference;}
    public double getPrix(){ return prix;}
    public String getNom(){return nom;}
    public int getQuantite(){return quantite;}
}
```

(Fin du sujet)