

IPST-CNAM
Programmation JAVA
NFA 001
Mercredi 29 Février 2012

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

1^{ère} Session NFA 001

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 en 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 les codes commentés en Java.

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

1. QCM (30 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
 - -1/2 pour chaque réponse fausse
- **les questions à 3 propositions** dont 1 seule proposition est fausse
 - + 1/2 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 **3 points** négatifs sans pénalité.

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

Le langage JAVA est-il un langage orienté objet ?		Q 1
1	OUI	
2	NON	

L'en-tête de déclaration d'une méthode main d'une classe JAVA est:		Q 2
1	static public void main(String args[])	
2	static public void main(int nb_args, String args[])	
3	static private void main(String args[])	

Soit le code suivant :		Q 3
<pre> Individu tab[] = new Individu[100]; Individu ind = new Individu(); ind.nom = "DUPONT" tab[0] = ind; tab[0].nom = "LAFONT" Terminal.ecrireStringln(ind.nom); </pre>		
Ce code affiche:		
1	DUPONT	
2	LAFONT	

<div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;"> <p>Package X</p> <div style="display: flex; flex-direction: column; align-items: center;"> <div style="display: flex; flex-direction: column; gap: 10px;"> <div style="border: 1px solid black; padding: 5px; width: 60px; text-align: center;">Class A</div> <div style="border: 1px solid black; padding: 5px; width: 60px; text-align: center;">Class B</div> <div style="border: 1px solid black; padding: 5px; width: 60px; text-align: center;">Class C</div> </div> <div style="border: 1px solid black; padding: 5px; width: 100px; text-align: center; margin-top: 20px;">Class Fille</div> </div> </div> <div style="text-align: center;"> <p>Package Y</p> <div style="border: 1px solid black; padding: 5px; width: 60px; text-align: center; margin-top: 20px;">Class E</div> <div style="border: 1px solid black; padding: 5px; width: 100px; text-align: center; margin-top: 20px;">Class FilleIndigne</div> </div> </div> <div style="margin: 20px 0;"> <div style="border: 1px solid black; padding: 5px; width: 150px; margin: 0 auto;"> <p style="margin: 0;">Class Mere</p> <pre style="margin: 0;"> { <genre> int x; } </pre> </div> <div style="display: flex; justify-content: space-around; margin-top: 20px;"> <div style="text-align: center;"> </div> <div style="text-align: center;"> </div> </div> </div>		Q 4
Compléter le tableau de visibilité suivant (0,25 point par réponse juste)		
si <genre> a la valeur suivante :	alors x est visible dans les classes :	
private		
public		
protected		
<i>indéfini (vide)</i>		

Soit le code suivant :		Q 5
<pre>public class Exemple { public ArrayList<String> arr; public Exemple(){ public add(String e) { arr.addElement(e); } }</pre>		
<p>Dans un programme Java:</p> <pre>Exemple ex = new Exemple(); ex.add("EXEMPLE"); Terminal.ecrireStringln(ex.arr.get(0));</pre>		
Ce code :		
1	affiche "EXEMPLE"	
2	affiche ""	
3	ne fonctionne pas correctement	

La commande javac		Q 6
1	crée un fichier .exe exécutable par le système d'exploitation via la JVM Java	
2	crée un fichier .class qui sera ensuite interprété par la machine virtuelle JAVA	

La commande java		Q 7
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, un tableau		Q 8
1	peut contenir des éléments de type primitif	
2	peut contenir des références d'objet JAVA	
3	ne peut pas contenir des références d'objet JAVA	

Le langage JAVA est portable sur la plupart des plateformes (windows,unix,linux,...)		Q 9
1	OUI	
2	NON	

La compilation d'un fichier .java (usage de la commande javac)		Q 10
1	crée un et un seul fichier .class	
2	crée un fichier .jar	
3	crée autant de fichier .class que de classes définies dans le fichier .java	

Dans la programmation orientée objet, un objet est :		Q 11
1	une instance d'une classe créée par exemple dans le programme principal	
2	une donnée structurée en mémoire de l'ordinateur	
3	un processus informatique qui s'exécute indépendamment du programme principal	

Une méthode statique est une méthode dont le contenu (le code) reste inchangé durant tout le temps d'exécution du programme Java		Q 12
1	OUI	
2	NON	

Le constructeur dans une classe d'objet permet :		Q 13
1	de construire la classe de définition de l'objet (attributs et méthodes)	
2	d'initialiser les attributs de l'objet avec des valeurs bien particulières	
3	d'initialiser les attributs en fonction des paramètres du constructeur	

Dans la classe String la méthode d'objet <i>void replace(String replaced, String new)</i> permet de remplacer dans l'objet String toutes les occurrences de la chaîne <i>replaced</i> par la chaîne <i>new</i> .		Q 14
1	OUI	
2	NON	

En JAVA, les méthodes déclarées en dehors d'une classe sont appelées des méthodes statiques		Q 15
1	OUI	
2	NON	

Une méthode publique d'une classe peut utiliser les attributs privés de la classe		Q 16
1	OUI	
2	NON	

La classe StringBuffer de Java permet de modifier une chaîne de caractère directement (exemple : méthode <i>insert</i> pour insérer une sous-chaîne)		Q 17
1	OUI	
2	NON	

La classe StringTokenizer :		Q 18
1	a un constructeur dont la signature est : <code>void StringTokenizer(String str)</code>	
2	permet de parcourir une chaîne de caractère afin d'y extraire les sous-chaînes qui sont séparées par un ou plusieurs caractères spécifiques	
3	permet de rechercher la position des tokens (ou caractères séparateurs) contenus dans une chaîne de caractère	

Le code suivant :		Q 19
<pre>String slue = "un::deux trois,:quatre cinq////six:sept huit neuf"; StringTokenizer str = new StringTokenizer(slue, " ,:/"); while (str.hasMoreTokens()) { String s = str.nextToken(); Terminal.ecrireString (s+"_"); } affiche :</pre>		
1	un_deux_trois_quatre_cinq_six_sept_huit_neuf_	
2	un_deux_trois_ quatre_ _ cinq_ _ _ six_sept_huit_neuf_	
3	un:deux_trois,quatre_cinq/six:sept_huit_neuf_	

En Java, il est possible de modifier le contenu d'un tableau passé en paramètre d'une méthode		Q 20
1	OUI	
2	NON	

Dans les langages orientés objet, le polymorphisme est la capacité pour un objet de se transformer en n'importe quel autre type d'objet		Q 21
1	OUI	
2	NON	

En JAVA, une classe peut hériter de plusieurs autres classes		Q 22
1	OUI	
2	NON	

En JAVA, l'interface est :		Q 23
1	une méthode permettant de transformer et échanger des données entre deux classes JAVA	
2	un mécanisme JAVA permettant d'écrire des traitements génériques	
3	un moyen de décrire l'interface graphique d'un programme JAVA	

Dans la conception objet d'une classe, l'agrégation et la composition d'objet sont deux notions bien distinctes. Soit, la classe A contenant par agrégation un objet de classe B et contenant par composition un objet de classe C		Q 24
1	Quand on détruit un objet de classe A alors les objets de classe B et C sont également détruits	
2	Quand on détruit un objet de classe A alors seul l'objet de classe B est détruit	
3	Quand on détruit un objet de classe A alors seul l'objet de classe C est détruit	

Dans la classe <i>ArrayList<Individu></i> la méthode <i>indexOf</i> permet de rechercher un individu dans la collection. Pour cela, il est indispensable que le traitement de comparaison entre l'élément recherché et les éléments de la collection soit implémenté dans :		Q 25
1	la méthode <i>equals</i> de la classe Individu : boolean equals(Object obj)	
2	la méthode <i>equals</i> de la classe Individu : boolean equals(Individu ind)	

En JAVA, un objet est un pointeur		Q 26
1	OUI	
2	NON	

Quelque soit le cas de figure, il est possible d'utiliser le constructeur par défaut d'une classe (Le constructeur par défaut est exécuté par l'instruction <i>Classe obj = new Classe ();</i>)		Q 27
1	OUI	
2	NON	

La difficulté dans la programmation objet est la maîtrise de la destruction des objets qui doit être faite dans l'ordre inverse de leurs constructions.		Q 28
1	Le langage JAVA ne fait pas exception à cette difficulté.	
2	Dans le langage JAVA cette difficulté est facilitée par l'implémentation systématique de la méthode <i>dispose</i> dans chacune des classes créées.	
3	Dans le langage JAVA cette difficulté n'existe plus grâce à un mécanisme interne appelé le garbage-collector (ou ramasse-mièges) qui détruit automatiquement les objets plus utilisés.	

Il faut initialiser les attributs d'un objet dans le(s) constructeur(s) car le langage JAVA n'initialise pas par défaut les attributs d'un objet		Q 29
1	OUI	
2	NON	

Si une classe B hérite d'une classe A alors :		Q 30
1	lors de la création d'un objet de type B, les attributs privés et les attributs publics de A sont alloués en mémoire et initialisés par le(s) constructeur(s) hérité(s) de A	
2	B hérite des méthodes privées de A	

La boucle for dite "énumérative" permet :		Q 31
1	d'incrémenter une valeur scalaire de type enum et de réaliser un traitement à chaque valeur	
2	créer une boucle permettant de faire évoluer un indice qui sert d'accès à une collection	
3	de parcourir tous les éléments de n'importe quelle collection	

Un constructeur d'une classe C est une méthode de la classe dont la forme de déclaration est : <code>public C C(paramètres).</code> Exemple d'utilisation : <code>C x = new C(12,"xx")</code>		Q 32
1	OUI	
2	NON	

Quand une classe définit un constructeur avec des paramètres alors le constructeur par défaut (sans paramètre) n'est plus accessible		Q 33
1	OUI	
2	NON	

Soit le code suivant : <code>int[] tab_int; tab_int[0]=12; tab_int[1]=3;</code>		Q 34
Ce code s'exécute correctement.		
1	OUI	
2	NON	

Soit le code suivant : <code>int tab_int[] = new int[10]; [...] for(int i=0; i< A ;i++) Terminal.ecrireIntln(B);</code>		Q 35
Ce code affiche tous les éléments du tableau tab_int. A et B doivent être remplacés par :		
1	A → <code>tab_int.size()</code> B → <code>tab_int.get(i)</code>	
2	A → 10 B → i	
3	A → <code>tab_int.length</code> B → <code>tab_int[i]</code>	

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

Donner une définition de ce qu'est un **objet** dans un langage orienté objet. Vous pouvez prendre en exemple le langage JAVA. (Soyez théorique et précis dans vos explications et vos affirmations).

Q 2

Expliquer pourquoi, en JAVA, alors que le passage de paramètre se fait par valeur, un objet passé en paramètre peut être modifié (modification de ses attributs).

Q 3

Décrivez toutes les étapes indispensables qu'il est nécessaire de faire pour écrire un programme Java puis de l'exécuter (moyens, commandes, code).

(Tourner la page)

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

Exercice [15 points]

Faire le programme complet JAVA qui détermine si un nombre entier strictement positif est un nombre premier.

Nous rappelons qu'un nombre est premier s'il n'est divisible par aucun nombre entier positif qui le précède autre que 1. En java un entier x est divisible par y si le reste de la division de x par y est égal à 0 ($x\%y$).

Le programme doit pouvoir saisir autant de valeurs que l'opérateur le désire.

Vous devez écrire deux classes :

- la classe **Math** qui contiendra la méthode statique: **static public boolean siPremier(int n)**; qui détermine si un nombre est premier.
- la classe **Exercice** qui appelle autant de fois que nécessaire la méthode **siPremier**.

Probleme [35 points]

1/

Faire le programme JAVA complet qui permet de créer un agenda dans lequel il est possible d'ajouter un rendez-vous (date de début, date de fin, lieu, objet). Les dates sont des chaînes de caractère de la forme "JJ/MM/AAAA HH:MM".

Pour cela vous devez créer la classe **Agenda**, **RendezVous** et **Probleme1**

La classe *Agenda* gère une collection de rendez-vous et contient la méthode **void add(RendezVous rdv)** permettant d'ajouter un rdv. Les rendez-vous sont rangés dans l'ordre d'arrivée.

La classe *RendezVous* définit un élément de rendez-vous.

La classe *Probleme1* contient la méthode main et teste la classe *Agenda*. Elle ajoute des rendez-vous et affiche tout le contenu de l'agenda.

2/

Modifier la classe *RendezVous* et la méthode *add* de *Agenda* afin que les rendez-vous soient classés par ordre chronologique de la date de début du rendez-vous. La méthode **long getTime()** de la classe *RendezVous*, vous est donnée, et retourne en millisecondes la date de début du rendez-vous.

3/ Ecrire la méthode **long getTime()** de la classe *RendezVous*.

Vous pouvez vous aider du code suivant :

```
Calendar cal = Calendar.getInstance();  
cal.set(int year, int month, int day, int hour, int minute, int second);  
long l = cal.getTimeInMillis();
```

NB: Vous pouvez utiliser la classe Terminal que nous avons vue tout au long de l'année.

(Fin du sujet)