

IPST-CNAM
Programmation JAVA
NFA 002
Mercredi 27 Juin 2012

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

2^{ème} Session NFA 002

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 (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
 - -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 **4 points** négatifs sans pénalité.

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

En Java, la classe <code>Hashtable<K,V></code> est une classe de définition d'une collection d'élément, dont les éléments sont de type V et l'indice d'accès est un entier K qui va de 0 à <code>size()-1</code>		Q 1
1	OUI	
2	NON	

En Java, la classe <code>Collections</code> permet de trier les éléments de n'importe quelle collection (classe qui implémente l'interface <code>List<T></code>) grâce à la méthode <code>Collections.sort(List<T>)</code> et si la classe d'appartenance des éléments de la collection implémente l'interface <code>Comparable</code>		Q 2
1	OUI	
2	NON	

JAVA est un langage orientée objet dont les fichiers sources sont directement interprétables		Q 3
1	OUI	
2	NON	

Une classe abstraite est une classe comme les autres et qui peut contenir des méthodes abstraites (sans code). Dans ce cas il est à la charge des classes dérivées d'implémenter ces méthodes abstraites		Q 4
1	OUI	
2	NON	

Soit le code correct suivant :		Q 5
<pre> public class A extends B { private int attr_A; public A() { attr_A = 10; attr_B = "TOTO"; C.attr_C = 100; } } </pre>		
1	attr_B peut être un attribut privé de B	
2	attr_C peut être un attribut statique et publique de C	
3	attr_C peut être un attribut statique et privé de C	

Soit deux classes B et C qui héritent d'une classe abstraite A. Les classes B et C peuvent utiliser par héritage les méthodes publiques non abstraites de la classe A.		Q 6
1	OUI	
2	NON	

Une classe qui contient au moins une méthode abstraite doit être déclarée abstraite.		Q 7
1	OUI	
2	NON	

En Java, on déclare un tableau de la manière suivante <code>T[] tab = new T[100]</code> .		Q 8
1	T peut être une classe abstraite	
2	T peut être une interface	
3	T ne peut pas être de type primitif	

En JAVA, si une classe C implémente plusieurs interfaces I1, I2, I3		Q 9
1	La classe C ou les classes dérivées de C doivent implémenter toutes les méthodes de toutes les interfaces I1, I2, I3	
2	La classe C ou les classes dérivées de C doivent implémenter toutes les méthodes de I1 mais	

	pas de I2 et I3	
--	-----------------	--

En JAVA, une interface peut contenir des attributs qui sont statiques et publiques		Q 10
1	OUI	
2	NON	

En JAVA, une interface permet de :		Q 11
1	créer une méthode générique dont le traitement utilise des méthodes de l'interface	
2	créer des classes abstraites	

La déclaration d'une méthode suivante : <pre>public void traitement(String s) throws MyException</pre> précise que la méthode propage l'exception MyException.		Q 12
1	OUI	
2	NON	

Soit le code suivant qui ajoute un Individu dans un tableau : <pre>public void ajouter(String nom) throws Exception { Individu ind=null; try { ind = rechercher(nom); } catch(NonTrouveException ex) { tab[n++] = ind; } }</pre> La méthode rechercher retourne l'exception NonTrouveException si le nom de l'individu n'est pas trouvé.		Q 13
1	si l'individu que l'on veut ajouter n'est pas trouvé alors la méthode retourne l'exception NonTrouveException	
2	si l'individu que l'on veut ajouter n'est pas trouvé et le tableau n'est pas plein alors l'individu est ajouté au tableau	
3	Si l'individu que l'on veut ajouter n'est pas trouvé et le tableau est plein alors la méthode retourne une exception prédéfinie Java (IndexOutOfBoundsException)	

Soit le code suivant : <pre>try{ System.out.println("AAA"); call(); System.out.println("BBB"); } catch(MyException ex) { System.out.println("DDD"); } catch(Exception ex) { System.out.println("CCC"); }</pre> avec la méthode <i>call</i> qui déclenche l'exception <i>UneAutreException</i> .		Q 14
Ce code affiche :		
1	AAA CCC	
2	AAA	
3	AAA DDD	

En JAVA, un package est une classe qui hérite de Package et permet de créer une collection à travers les méthode pack() et unpack()		Q 15
1	OUI	
2	NON	

En JAVA le "package" est un design pattern qui modélise les principes de compression et dedécompression des objets dans un socket (pliage/dépliage ou pack/unpack)		Q 16
1	OUI	
2	NON	

Le design pattern Factory est un modèle de conception de la mise en œuvre d'une communication client serveur		Q 17
1	OUI	
2	NON	

Le Singleton est un design pattern qui est une classe contenant la méthode unique public T getInstance() . A chaque appel cette méthode retourne un objet unique de type T		Q 18
1	OUI	
2	NON	

Le code suivant crée un descripteur de fichier dont le nom est "exemple.txt" permettant par exemple de tester si le fichier existe <pre>File fichier; fichier = new File("exemple.txt");</pre>		Q 19
1	OUI	
2	NON	

La classe File permet de parcourir les fichiers d'un répertoire		Q 20
1	OUI	
2	NON	

On a le code suivant : <pre>File fichier = new File("ListeDouble.bin"); FileOutputStream fos = new FileOutputStream(fichier); DataOutputStream dos = new DataOutputStream(fos); dos.writeInt(tab.length); for(int i=0;i< tab.length;i++) dos.writeDouble(tab[i]); dos.close();</pre>		Q 21
1	Ce code crée un fichier de nom 'ListeDouble.bin' contenant la liste de doubles	
2	Ce code crée un fichier dont les informations sont dans un format texte	
3	Ce code crée un fichier dont les informations sont dans un format binaire	

La class File ne gère que les fichiers. Pour gérer des répertoires on utilise la classe Directory		Q 22
1	OUI	
2	NON	

La sérialisation est un service du langage Java qui permet d'écrire et de lire n'importe quel objet dans un fichier binaire (Serializable) ou un fichier texte (XML)		Q 23
1	OUI	
2	NON	

Le code suivant est correct :		Q 24
<pre>File f = new File("/etc/passwd"); System.out.println(f.exists()); System.out.println(f.canRead()); System.out.println(f.canWrite()); System.out.println(f.getLength()); File d = new File("/etc/"); System.out.println(d.isDirectory()); String[] files = d.list(); for(int i=0; i < files.length; i++) System.out.println(files[i]);</pre>		
1	OUI	
2	NON	

Le package est une unité de programmation permettant de regrouper et architecturer les classes du langage Java (prédéfinies ou développées) dans des répertoires et accessibles aux autres unités de programmation (programme Java, Applet, ...).		Q 25
1	OUI	
2	NON	

Un package est un répertoire qui ne peut pas contenir d'autres répertoires.		Q 26
1	OUI	
2	NON	

L'option -d de la commande javac permet de générer les fichiers .class dans un autre répertoire que celui contenant les fichiers .java.		Q 27
1	OUI	
2	NON	

En JAVA, un thread est une JVM qui s'exécute en parallèle de la JVM dans laquelle le thread a été créé.		Q 28
1	OUI	
2	NON	

Soit le thread définit de la manière suivante :		Q 29
<pre>public class MonThread extends Thread { . . . } et MonThread t = new MonThread();</pre>		
On démarre le thread t de la manière suivante :		
1	t.run()	
2	t.start()	
3	new Thread(t).start()	

La classe A hérite de B qui hérite de C. C est une classe abstraite qui implémente une interface I. A et B ne sont pas des classes abstraites		Q 30
1	C peut implémenter une partie des méthodes de l'interface I	
2	A et B doivent à elles deux implémenter toutes les méthodes de I qui n'ont pas été implémentées par C	
3	A ne peut pas implémenter des méthodes de l'interface I	

Soit le code suivant :		Q 31
<pre>class MonRunnable extends Toto implements Runnable { public void run() { // traitement } }</pre>		
Pour créer et démarrer le thread :		
1	<code>new Thread(new MonRunnable ()).start();</code>	
2	<code>MonRunnable p = (MonRunnable)(new Thread());</code> <code>p.start();</code>	
3	<code>MonRunnable p = new MonRunnable ();</code> <code>Thread q = new Thread(p);</code> <code>q.start();</code>	

Un tableau Java peut contenir des type primitifs.		Q 32
1	OUI	
2	NON	

Une collection polymorphe est une instance d'une classe P<T> contenant des objets dont les classes d'appartenance peuvent être différentes et toutes ces classes héritent de la classe abstraite T:		Q 33
1	OUI	
2	NON	

En JAVA, la classe RuntimeException qui hérite de la classe Exception permet de déclencher une exception pour laquelle il n'est pas utile de déclarer comme devant être propagé		Q 34
1	OUI	
2	NON	

La sérialisation est une interface prédéfinie de JAVA qui permet d'écrire et lire les objets dans un fichier		Q 35
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

Expliquez à quoi sert l'interface.

Q 2

Quels sont les deux moyens permettant de créer un thread ?
Pourquoi utiliser l'un ou l'autre ?

Q 3

A quoi est utilisé le design paterne factory ?
Quel est son principe fondamental ?

(Tourner la page)

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

Probleme [30 points]

On se propose de gérer un agenda de rendez-vous.

La classe Agenda gère une collection de RendezVous et permet de :

- ajouter un nouveau rendez-vous dans la collection
- afficher tous les rendez-vous de la collection.

La classe RendezVous est une classe abstraite qui contient la méthode abstraite **afficher()**.
Les classes RdvJournalier et RdvPeriodique héritent de la classe RendezVous.

La classe RdvJournalier permet de créer un rendez-vous unique dans le temps alors que la classe RdvPeriodique permet de définir un rendez-vous qui est soit hebdomadaire, soit mensuel entre une date de début (1^{ère} occurrence) et une date de fin.

1/

Faites une conception des classes RendezVous, RdvJournalier et RdvPeriodique :

- identifier les attributs de chaque classe
- définir les paramètres des constructeurs de chaque classe
- définir les méthodes de chaque classe

Commentez vos choix de conception.

Les dates sont des chaînes de caractère de la forme AAAA-MM-JJ (ex 20012-09-23)

Les heures sont des chaînes de caractères de la forme : HHhMM (ex : 10h20)

2/

Ecrire en Java la classe Agenda :

- les attributs
- le constructeur
- la méthode **public void ajouter(RendezVous rdv)** qui permet d'ajouter un rendez-vous
- la méthode **public void afficher()** qui affiche à l'écran le contenu de l'agenda

Ecrire en Java les classes RendezVous, RdvJournalier et RdvPeriodique.

Ecrire la classe Probleme qui contient le programme **main** qui crée un agenda, ajoute à titre d'exemple un rendez-vous journalier et un rendez-vous périodique et qui affiche l'agenda.

2/ Ecrire dans la classe Agenda la méthode **trier()** qui permet de trier par ordre croissant dans le temps des rendez-vous de l'agenda et qui est appelée dans la méthode afficher.

(Fin du sujet)