

IPST-CNAM
Programmation JAVA
NFA 032
Mercredi 24 Juin 2015

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

1^{ère} Session NFA 032

L'examen se déroule en deux parties. Une première partie de 1h15mn, sans document, consacrée à des questions de cours, et une deuxième partie de 1h 15mn, 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)
Durée: 1h15

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:
------	---------

Soit le code suivant ::		Q 1.
<pre> public class A extends C { public void faire() { B b; b.x = 100; y = 200; } } </pre>		
Les classes A, B et C sont toutes des classes public.		
1	si x est un attribut public de B et y est un attribut protected de C alors ce code est correct	
2	si x est un attribut public de B et y est un attribut private de C alors ce code est correct	
3	si x est un attribut public de B et y est un attribut public de C alors ce code est correct	

Quand la classe A hérite de B, alors :		Q 2.
1	A a accès à tous les constructeurs public de B	
2	A a accès à tous les attributs private de B	
3	A a accès à toutes les méthodes private de B	

On a une classe A qui contient qu'un seul constructeur :		Q 3.
<pre> public A(int x){ } </pre>		
On a la classe B qui hérite de A, qui contient le constructeur suivant :		
<pre> public B(){ attr = 100; } </pre>		
Ce code est correct		
1	OUI	
2	NON	

Soit une classe B qui hérite d'une classe A, et B n'a pas de constructeur alors :		Q 4.
1	lors de la création d'un objet de type B, une erreur d'exécution se produit si la classe A n'a pas défini de constructeur	
2	lors de la création d'un objet de type B, les attributs privés de A sont alloués en mémoire et initialisés par le constructeur hérité de A	
3	Il n'est pas possible de créer un objet de B car la classe B n'a pas de constructeur	

Soit la classe C2 qui hérite de la classe C1. Si C1 définit une méthode alors C2 peut définir la même méthode		Q 5.
1	OUI	
2	NON	

En programmation objet, une collection polymorphe est une classe qui hérite de plusieurs classes		Q 6.
1	OUI	
2	NON	

En JAVA, une classe peut hériter de plusieurs autres classes		Q 7.
1	OUI	
2	NON	

Si une classe B hérite d'une classe A alors :		Q 8.
1	B hérite des méthodes privées de A	
2	lors de la création d'un objet de type B, les attributs de A sont alloués en mémoire et initialisés par le constructeur hérité de A	

La caractéristique "protected" d'un attribut de la classe C est utilisée pour :		Q 9.
1	interdire l'écriture mais pas la lecture de l'attribut par les classes extérieures à C	
2	les classes qui héritent de C peuvent accéder, en lecture et écriture, à cet attribut mais pas les classes extérieures au package de C	

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 10.
1	OUI	
2	NON	

Soit la classe C2 qui hérite de la classe C1		Q 11.
1	La classe C2 peut utiliser directement les attributs privés de C1	
2	La classe C2 peut utiliser un des constructeurs de C1	

En Java, il est possible de faire l'héritage entre deux classes abstraites		Q 12.
1	OUI	
2	NON	

L'interface permet de :		Q 13.
1	gérer des collections polymorphes (les éléments sont de type d'une interface)	
2	gérer les fichiers (interface avec le système d'exploitation)	
3	passer en paramètre d'une méthode "un traitement" (traitements génériques)	

En JAVA, une exception est un objet		Q 14.
1	OUI	
2	NON	

Le code suivant est correct :		Q 15.
<pre> public void action(int parametre) { if (parametre==0) throw new MyException("Erreur"); else faireLeTraitement(); } public class MyException extends RuntimeException{} </pre>		
1	OUI	
2	NON	

Le code suivant est correct :		Q 16.
<pre> public void action(int parametre) { if (parametre==0) throw new RuntimeException("Erreur"); else faireLeTraitement(); } </pre>		
1	OUI	
2	NON	

En JAVA, l'instruction suivante permet de déclencher une exception <pre> throw new Exception("Impossible de faire l'action");</pre>		Q 17.
1	OUI	
2	NON	

En JAVA, l'instruction suivante permet de déclencher une exception <pre> throw new RuntimeException("Impossible de faire l'action");</pre>		Q 18.
1	OUI	
2	NON	

Le code suivant est correct :		Q 19.
<pre>public void action(int parameter) { if (parametre==0) throw new MonException("action","parametre == 0"); else faireLeTraitement(); } avec public class MonException extends Exception { public MonException(String nomMethode,String erreur) { super("Erreur dans "+nomMethode+" :"+erreur); } }</pre>		
1	OUI	
2	NON	

En Java, la classe <code>Hashtable<K,V></code> permet de créer un tableau dont les éléments sont de type de la classe V qui hérite de K		Q 20.
1	OUI	
2	NON	

En Java, la méthode suivante permet de trier les éléments d'une collection : <code>Collection.sort(List<T> list, Comparator<? super T> c)</code>		Q 21.
1	Comparator est ici une classe prédéfinie du langage Java qui permet de comparer les éléments de la collection	
2	Comparator est ici une classe abstraite du langage Java qu'il faut dériver en une classe qui surcharge la méthode <code>int compare(T o1, T o2)</code> qui est utilisée pour comparer les éléments de la collection	
3	Comparator est ici une interface qui contient la méthode <code>int compare(T o1, T o2)</code> qui est utilisée pour comparer les éléments de la collection	

En Java, on déclare un tableau qui contient des éléments dont la classe d'appartenance implémente une interface I : <pre> I[] tab;</pre>		Q 22.
1	Cela est possible et on ajoute dans le tableau des objets de type I : <code>t[i]=new I();</code>	
2	Cela est possible et on ajoute dans le tableau des objets de type B : <code>t[i]=new B();</code> avec la classe B qui implémente l'interface I	

En Java, pour rechercher un élément dans un tableau de Contact, ArrayList<Contact>, on peut utiliser la méthode prédéfinie public boolean contains . Dans ce cas il faut que :		Q 23.
1	la méthode public int compareTo(Object o) soit implémentée dans la classe Contact	
2	la méthode public boolean comparer(Object o) soit implémentée dans la classe Contact	
3	la méthode public boolean equals(Object o) soit implémentée dans la classe Contact	

La sérialisation est un principe natif du langage JAVA permettant de plier et déplier les attributs des objets afin de pouvoir transporter les objets via un fichier ou un socket		Q 24.
1	OUI	
2	NON	

Soit le code suivant : <pre> public class Individu implements Serializable {String nom; String prenom;} </pre> Puis, <pre> Individu ind = new Individu(); File fic = new File(".", "fic.bin"); FileOutputStream fin = new FileOutputStream(fic); fin.writeObject((Object)ind); fin.close(); </pre> Ce code est correct		Q 25.
1	OUI	
2	NON	

En JAVA, créer un thread consiste à implémenter la méthode void run() dans une classe qui hérite de Thread		Q 26.
1	OUI	
2	NON	

Deux threads peuvent exécuter en même temps la même méthode d'un même objet et donc, par cela, modifier en même temps les mêmes attributs de l'objet		Q 27.
1	OUI	
2	NON	

Soit deux thread t1 et t2 qui utilisent la méthode objet définie ainsi <pre> synchronized public void miseajour(){ . . . } </pre> t1 réalise l'appel suivant : o1.miseajour(); t2 réalise l'appel suivant : o2.miseajour(); o1 et o2 sont deux objets différents. Quand t1 est en train d'exécuter cette méthode de l'objet o1 alors t2 (qui veut aussi l'exécuter sur o1) est en attente que t1 ait fini de l'exécuter		Q 28.
1	t2 est en attente que t1 ait fini d'exécuter la méthode	
2	t1 et t2 exécutent en parallèle la méthode	

La méthode "start" est la méthode qui permet de démarrer un thread		Q 29.
1	Cette méthode doit être implémentée dans la classe qui hérite de Thread	
2	Cette méthode appartient à la classe Thread. On y accède par héritage de la classe Thread	
3	Cette méthode appelle la méthode run() que nous avons écrit dans notre thread	

Le socket est un canal de communication permettant de faire communiquer deux logiciels		Q 30.
1	Ces deux logiciels peuvent être situés sur deux machines différentes connectés en réseau	
2	Ces deux logiciels peuvent être situés sur la même machine	
3	Ces deux logiciels doivent être situés sur des machines différentes	

En JAVA, la classe ServerSocket permet de créer un serveur de socket. Le code peut être :		Q 31.
1	<code>ServerSocket ssoc = new ServerSocket("localhost", 9100)</code>	
2	<code>ServerSocket ssoc = new ServerSocket("localhost")</code>	
3	<code>ServerSocket ssoc = new ServerSocket(9100)</code>	

En JAVA, les données échangées, via un socket, entre un client et un serveur, sont des données qui sont écrites et lues dans un flot séquentiel d'information		Q 32.
1	OUI	
2	NON	

Une communication synchrone entre un producteur et un consommateur est que		Q 33.
1	la production des évènements est indépendante de leurs consommations	
2	la production des évènements est synchrone à leurs consommations	

Le "mode push synchrone" est : le producteur envoie une donnée à chacun des consommateurs sans se mettre en attente de la consommation de la donnée par les consommateurs. Les consommateurs reçoivent alors la donnée, réalisent leur traitement et n'envoient pas un accusé au producteur.		Q 34.
1	OUI	
2	NON	

<pre> sequenceDiagram participant S as :Serveur Producteur participant C1 as :Client1 Consommateur participant C2 as :Client2 Consommateur S->>C1: push(objet) S->>C2: push(objet) S->>C1: push(objet) S->>C2: push(objet) C1->>C1: faireTr(objet) C2->>C2: faireTr(objet) C1->>C1: faireTr(objet) C2->>C2: faireTr(objet) </pre>		Q 35.
Ce diagramme de séquence correspond à un mode de communication :		
1	push-synchrone	
2	push-asynchrone	

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 ce qu'est une classe abstraite et dans quel cas on l'utilise.

Q 2

Citez au moins 4 grands principes qu'il est nécessaire de respecter pour réaliser une communication client/serveur en utilisant un socket en JAVA.

Q 3

Expliquez pourquoi il est avantageux d'utiliser un intermédiaire pour réaliser la communication avec des producteurs et des consommateurs d'évènements.

(Tourner la page)

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

Durée: 1h15

Problème [50 points]

On se propose de créer un serveur qui gère des contacts. Les contacts sont en mémoire du serveur qui gère un annuaire qui contient les contacts.

Un contact est défini par la classe **Contact** dont les attributs sont :

- un nom
- un prénom
- un numéro de téléphone.

Ce serveur accepte les requêtes suivantes :

Requete1 : "RECHERCHER_CONTACT" " *rech*"

Requete 2 : "AJOUTER_CONTACT" "*nom*" "*prenom*" "*telephone*"

La requête 1 consiste à rechercher les contacts dont le nom ou le prénom contiennent la chaîne "*rech*". Si la chaîne est égale à "" alors la requête retourne tous les contacts.

En retour la requête écrit sur le socket les contacts trouvés :

"*nb*" "*nom1*" "*prenom1*" "*telephone1*" "*nom2*" "*prenom2*" "*telephone2*"

avec "*nb*" = le nombre de contacts trouvés

La requête 2 consiste à ajouter dans le serveur un nouveau contact s'il n'existe pas (même nom et même prénom).

En retour la requête retourne OK ou NOK.

OK : l'ajout a été fait (le contact n'existait pas déjà).

NOK : l'ajout n' a pas été fait (le contact existe déjà).

1/ Ecrire, en JAVA, le programme complet de ce serveur.

Vous écrivez au moins les classes suivantes :

- la classe **Serveur** qui est le programme principal (main)
- la classe **Annuaire** qui contient les contacts
- la classe **Contact** qui définit un contact.

Remarque: les éléments des requêtes: "*rech*" "*nom*" "*prenom*" "*telephone*"; peuvent contenir des caractères blancs.

2/ Ecrire un programme client (pas d'IHM) qui fait que le traitement suivant :

- il se connecte au serveur
- il envoie la requête : "RECHERCHER_CONTACT" ""
- il récupère tous les contacts retournés par la requête
- il affiche à l'écran (System.out.println) les contacts par ordre alphabétique sur le nom et le prénom.

(Fin du sujet)