

IPST-CNAM  
Programmation JAVA  
NFA 001  
Mercredi 20 Juin 2012

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

**2ème 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<sup>ère</sup> 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 **4 points** négatifs sans pénalité.

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

Le langage JAVA n'est pas un langage orienté objet mais un langage fonctionnel de dernière génération		Q 1
1	OUI	
2	NON	

Soit le code suivant :		Q 2
<pre> public class Exemple {     public static void main(String args[]) {         String str = args[1];         System.out.println(str);     } } </pre>		
Commande d'exécution : java Exemple toto		
1	Ce programme ne se compile pas car il y a une erreur de syntaxe	
2	L'exécution échoue car il y a une erreur d'exécution	
3	L'exécution de ce programme affiche à l'écran la chaîne de caractère passée en argument"	

Soit le code suivant :		Q 3
<pre> Individu tab[] = new Individu[100]; Individu ind = new Individu(); ind.nom = "ROUSSEAU" tab[0] = ind; ind.nom = "VOLTAIRE"; Individu ind2 = tab[0]; Terminal.ecrireStringln(ind2.nom); </pre>		
Ce code :		
1	affiche VOLTAIRE	
2	affiche ROUSSEAU	
3	est incorrect	

Soit le code suivant ::		Q 4
<pre> public class A extends C {     public void faire(C c) {         B b;         b.x = 100;         c.y = 200;     } } </pre>		
Les classes A, B et C sont toutes des classes public.		
1	si y est un attribut protected de C et x est un attribut public de B alors ce code est correct	
2	si x est un attribut privé de B et y est un attribut public 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	

Soit le code suivant :		Q 5
<pre>public class Exemple {     public String[] arr;     private int n;     public Exemple(){         n=0;         arr = new String[10];     }     public add(String e){         arr[n++] = e;     } }</pre> <p>Dans un programme Java:          Exemple ex = new Exemple();          ex.add("EXEMPLE");          Terminal.ecrireStringln(ex.arr[0]);</p>		
Ce code :		
1	affiche "EXEMPLE"	
2	affiche ""	
3	ne fonctionne pas correctement	

Les fichiers .class générés par la compilation javac sont des fichiers intermédiaires qui contiennent un programme exprimé dans un autre langage (P-CODE) et qui est ensuite :		Q 6
1	traduit en fichier binaire directement exécuté par le microprocesseur	
2	interprété par un interpréteur de code P-CODE	

La commande suivante : <b>java MaClasse.main2 AAA 999</b> exécute un programme Java dont la classe main est dans la classe "MaClasse" et dont la méthode main2 est déclarée de la manière suivante :		Q 7
<pre>public static void main2(String arg1, int arg2)</pre>		
1	OUI	
2	NON	

En JAVA, un tableau		Q 8
1	peut contenir des éléments de type primitif	
3	ne peut pas contenir des références d'objet JAVA	

Soit le code suivant :		Q 9
<pre>public class Exemple {     ????? int var_x;     public static void main(String a_args[])     {         System.out.println("X=" + var_x);     } }</pre> <p>Pour que ce code soit correct, il faut remplacer <b>?????</b> par :</p>		
1	private	
2	public	
3	static	

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

Dans la programmation orientée objet, une classe est un concept informatique qui contient la déclaration des informations suivantes :		Q 11
1	des attributs privés ou publics, et des méthodes privés ou publics	
2	des classes internes publics	
3	des méthodes statics et des attributs non statics	

En JAVA, un attribut static est une donnée dont la valeur est toujours constante durant l'exécution du programme JAVA		Q 12
1	OUI	
2	NON	

Le constructeur d'une classe A :		Q 13
1	peut initialiser les attributs privés de la classe A	
2	ne peut pas initialiser les attributs privés de la classe A	
3	peut initialiser les attributs publics d'une autre classe B public	

Quand la classe A hérite de B, cela signifie que :		Q 14
1	A a accès à tous les constructeurs publics de B	
2	A a accès à tous les attributs privés de B	
3	A a accès à toutes les méthodes publics de B	

En programmation objet, dire que la classe A " <u>est composée de</u> " la classe B, signifie que A contient un attribut de classe B dont la valeur est allouée (new B()) dans le constructeur de A		Q 15
1	OUI	
2	NON	

En programmation objet, une collection polymorphe est une classe qui hérite de plusieurs classes		Q 16
1	OUI	
2	NON	

De ces 3 classes quelles sont les classes qui permettent de modifier une chaîne de caractère		Q 17
1	String	
2	StringBuffer	
3	StringTokenizer	

La classe StringToKenizer est une classe qui permet de convertir une chaîne de caractère (String) en une Tokenizer (collection polymorphe dont chaque élément appartient à une classe qui hérite de la classe "Token")		Q 18
1	OUI	
2	NON	

Le "garbage collector" ou ramasse miettes est un traitement de la JVM qui permet automatiquement de détruire les objets dont la référence n'est contenue dans aucun autre objet de la JVM		Q 19
1	OUI	
2	NON	

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

On a une classe A qui contient qu'un seul constructeur :		Q 21
<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>		
1	Ce code est correct	
2	Ce code n'est pas correct	

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 22
1	OUI	
2	NON	

En JAVA, la déclaration d'un tableau comme suit :		Q 23
<pre> Individu[] tab_ind = new Individu[10] </pre>		
contient :		
1	10 éléments dont les valeurs sont égales au résultat de l'exécution du constructeur <i>Individu()</i>	
2	10 éléments dont les valeurs ont toutes à <i>null</i>	

Dans la conception objet d'une classe, l' <u>agrégation</u> et la <u>composition</u> 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>Collections</i> la méthode <i>sort</i> permet de rechercher un élément dans un collection dont les éléments sont de type T. Pour cela, il est faut que la classe T implémente l'interface <b>Comparable</b> :		Q 25
1	OUI	
2	NON	

En JAVA, pour qu'un objet puisse être passé en paramètre d'une méthode, il faut que la classe d'appartenance de l'objet soit une classe public		Q 26
1	OUI	
2	NON	

Si on ne crée pas de constructeur à une classe alors il n'est pas possible de créer une instance de cette classe		Q 27
1	OUI	
2	NON	

Le code suivant se compile-t-il correctement :		Q 28
<pre> public class Individu {     private String nom;     private String prenom;     public void Individu(){         nom=" "; prenom=" ";     } } </pre>		
1	OUI	
2	NON	

Soit la classe A définie ainsi :		Q 29
<pre>public class A {     ArrayList&lt;String&gt; liste;     public void ajouter(String s) {         liste.add(s);     } }</pre>		
Le code suivant s'exécute-t-il correctement :		
<pre>a = new A(); a.ajouter("TOTO");</pre>		
1	OUI	
2	NON	

La classe String contient la méthode :		Q 30
<pre>public void replace(String str, String str_replace)</pre>		
qui remplace dans l'objet de type String toutes les sous-chaine str par str_replace		
1	OUI	
2	NON	

La classe StringBuffer est une classe prédéfinie JAVA qui gère une chaîne de caractère dans laquelle il est possible d'ajouter de nouveaux caractères		Q 31
1	OUI	
2	NON	

En JAVA, il n'existe pas d'ordre dans la déclaration des attributs et des méthodes		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	

En JAVA, il est possible de déclarer des attributs en dehors de la classe. Ils sont alors déclarés en static.		Q 34
1	OUI	
2	NON	

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

## 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.

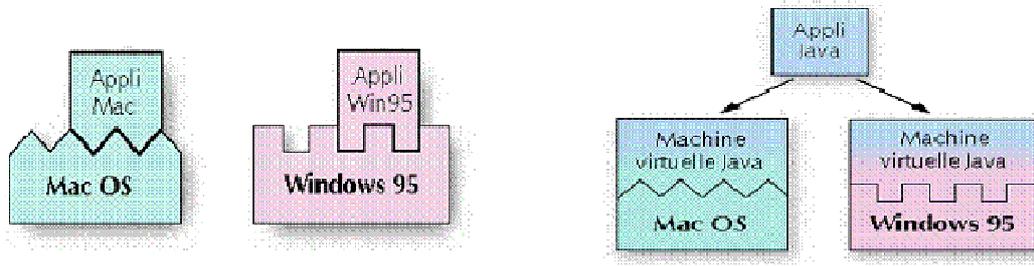
### Q 1

Soit une classe A qui contient une méthode m quelconque avec des paramètres.

Enumérez et précisez tous les éléments du langage Java que la méthode peut utiliser dans son corps de méthode. Expliquez.

### Q 2

Veillez commenter le schéma suivant :



### Q 3

Donner une définition assez complète pour chacun des concepts suivants :

- héritage
- polymorphisme
- encapsulation

(Tourner la page)

---

**2<sup>ème</sup> PARTIE : PROGRAMMATION (avec document)**

---

**Exercice [20 points]**

Faire le programme JAVA complet qui saisie une chaîne de caractère composée de mots et qui affiche par ligne à l'écran les mots de même longueur en commençant par les plus petits. Dans chaque ligne, les mots sont ordonnés par ordre alphabétique croissant.

**Probleme [30 points]**

On se propose de gérer un répertoire téléphonique.

Le répertoire téléphonique contient des contacts. Un contact peut avoir plusieurs numéros de téléphone.

Un numéro de téléphone est composé de deux informations : un intitulé (optionnel) qui caractérise le numéro (ex: "portable"; "home", "petit ami", "père", ...), et le numéro (ex: 0561722000).

Nous faisons le choix qu'un numéro de téléphone est une chaîne de la forme:  
**"intitulé / numéro"** ou **"numéro"**

Un contact est défini par son nom (texte libre) et une liste de ses numéros de téléphone.

La classe **Repertoire** gère une collection de contact et contient les méthodes permettant de réaliser les traitements suivants :

- ajouter un nouveau contact :  
**void ajouterContact(String nom, String intitulé, String numéro)**
- ajouter un numéro à un contact existant :  
**void ajouterNumero(String nom, String intitulé, String numéro)**  
Si le nom n'existe pas alors afficher un message d'erreur
- modifier le numéro d'un des numéros d'un contact :  
**void modifier(String nom, String intitulé, String numéro)**  
Si le nom ou l'intitulé n'existe pas alors afficher un message d'erreur
- retourne tous les numéros d'un contact :  
**String[] getNumeros(String nom)**  
Si le nom n'existe pas alors afficher un message d'erreur
- affiche tout le contenu de l'agenda par ordre alphabétique sur le nom des contacts  
**void afficher()**

La classe **Contact** contient une collection de chaîne de la forme " ***intitulé / numéro*** ".

Ecrire les classes complètes **Repertoire** et **Contact**.

**(Fin du sujet)**