

IPST-CNAM  
Programmation JAVA  
NFA 001  
Mercredi 4 Février 2015

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

**1<sup>ère</sup> 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 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 (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:
------	---------

Dans un langage orienté objet, un principe fort est que les attributs (ou données) non statiques sont alloués dans une instance d'une classe appelée un objet. Ainsi,		Q 1.
1	les données d'une instance peuvent être directement (sans passer par une méthode) modifiées par toutes les autres instances du programme	
2	Une instance peut "protégés" ses données en ne les rendant pas directement accessibles par les autres instances du programme	
3	les données du programme sont réparties dans toutes les instances du programme	

Dans un langage orienté objet, un objet est :		Q 2.
1	une structure de données valuées qui répond à un ensemble de « messages » (ou « méthodes »)	
2	une vision virtuelle d'un paquet d'information qui est extérieur au programme	

Un avantage du langage Java est que le code généré contenu dans les fichiers .class est exécutable par différents Système d'exploitation (PC, Mac, Linux)		Q 3.
1	OUI	
2	NON	

Soit l'arborescence d'un programme java suivante :		Q 4.
<pre> projetNFA031   bin   src       fr/cnam/projet/Projet.java (main)       fr/cnam/ihm/Formulaire.java   compil.bat   run.bat </pre>		
Le fichier de compilation compil.bat est exécuté dans le répertoire projetNFA031.		
Les fichiers compilés sont générés dans le répertoire bin		
Le fichier compil.bat peut être :		
1	javac -d bin -classpath "." fr/cnam/projet/Projet.java	
2	javac -d bin -classpath "src" src/fr/cnam/projet/Projet.java	
3	cd src javac -classpath "." -d ../bin fr/cnam/projet/Projet.java	

Soit l'arborescence d'un programme java suivante :		Q 5.
<pre> projetNFA031   src       fr/cnam/projet/Projet.java (main)       fr/cnam/ihm/Formulaire.java </pre>		
La classe Projet utilise la classe Formulaire.		
Le fichier Formulaire.java contient la ligne : package fr.cnam.ihm;		
Le fichier Projet.java doit contenir la ligne d'import suivante :		
1	import Formulaire;	
2	import fr.cnam.ihm.*;	
3	import fr.cnam.ihm.Formulaire;	

Pour exécuter un programme Java, on exécute la commande suivante :		Q 6.
<b>java Programme</b>		
1	dans ce cas, la JVM exécute le constructeur par défaut de la classe Programme	
2	dans ce cas, la JVM exécute la méthode main de la classe Programme	

En JAVA, un répertoire qui contient des fichiers .java est nécessairement un package		Q 7.
1	OUI	
2	NON	

<div style="text-align: center;"> <pre> classDiagram     package PackageX {         ClassA         ClassB         ClassC         ClassMere {             &lt;del&gt;genre&lt;/del&gt; int x         }         ClassFille     }     package PackageY {         ClassE     }     ClassFille --&gt; ClassMere     ClassFilleIndigne --&gt; ClassMere             </pre> </div> <p>Si &lt;del&gt;genre&lt;/del&gt; est private alors :</p>		Q 8.
1	Seules les classes Fille, FilleIndigne et Mere peuvent accéder à l'attribut x	
2	Seules les classes A, B, Fille et Mere peuvent accéder à l'attribut x	
3	Seule la classe Mere peut accéder à l'attribut x	

Soit la classe C1 et la classe C2 dont tous les attributs sont privés. Les deux classes C1 et C2 appartiennent au même package. Dans ce cas, les méthodes de C1 peuvent accéder directement aux attributs d'une instance de C2		Q 9.
1	OUI	
2	NON	

Une méthode public d'une classe A peut utiliser directement les attributs publics de la classe B		Q 10.
1	OUI	
2	NON	

Soit la classe C1 dont tous les attributs sont <u>publics</u> et <u>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 11.
1	OUI	
2	NON	

Dans une classe JAVA, le constructeur d'une classe permet de :		Q 12.
1	intialiser les attributs de la classe	
2	d'allouer en mémoire les attributs d'objet de la classe	
3	de démarrer le programme dès l'appel du constructeur	

Le code suivant est un exemple correct d'un constructeur :		Q 13.
<pre> public class Individu {     private String nom;     private String prenom;     private int age;      public Individu Individu(String nom, String prenom, int age)     {         this.nom=nom; this.prenom=prenom; this.age = age;         return this;     } } </pre>		
1	OUI	
2	NON	

Soit le code JAVA suivant		Q 14.
<pre> public class Constructeur {     public static void main(String args[])     {         Exemple1 ex = new Exemple1();         ex.tab[10] = 22;     } }  class Exemple1 {     public int[] tab;      public void Exemple1()     {         tab = new int[5];     } } </pre>		
Ce code s'exécute correctement :		
1	OUI	
2	NON	

Soit le code suivant :		Q 15.
<pre> ArrayList&lt;Individu&gt; tab = new ArrayList&lt;Individu&gt;(); Individu ind = new Individu("LAFONT", "Pierre");  tab.add(ind); tab.add(ind); System.out.println("Nombre : ", tab.size()); </pre>		
L'exécution de ce code :		
1	déclenche une erreur car deux objets identiques (de même adresse mémoire) ne peuvent être ajoutés dans un même tableau	
2	affiche: 1	
3	affiche: 2	

Soit une classe C qui contient qu'un seul constructeur public C(int x)		Q 16.
Il est possible d'écrire C c = new C() qui crée un objet		
1	OUI	
2	NON	

Soit le code suivant :		Q 17.
<pre> Livre l = new Livre(); l.nom = "Les cavernes d'acier"; ArrayList&lt;Livre&gt; livres = new ArrayList&lt;Livre&gt;(); livres.add(l); l.nom="Face aux feux du soleil"; livres.add(l); System.out.println(livres.get(0).nom); System.out.println(livres.get(1).nom); </pre>		
Ce code affiche :		
1	Les cavernes d'acier Face aux feux du soleil	
2	Les cavernes d'acier Les cavernes d'acier	
3	Face aux feux du soleil Face aux feux du soleil	

Soit le code suivant qui teste si v est premier ou non :		Q 18.
<pre> int v=4; boolean premier=true; for(int k=2;k&lt;v;k=k+1)     if (v%k == 0) premier = false;     else premier = true; if (premier) System.out.println("PREMIER"); else System.out.println("NON PREMIER"); </pre>		
Ce code :		
2	affiche "NON PREMIER"	
3	affiche "PREMIER"	

Soit le code suivant :		Q 19.
<pre> int i=0;; int[] tab = new int[3]; for(int v : tab) {     tab[i] = v*10;     i = i+1; } for(int j=0;j&lt;tab.length;j++)     System.out.print(tab[j]+" "); </pre>		
Ce code affiche :		
1	0 10 100	
2	0 0 0	

En JAVA, le type de retour d'une méthode est toujours la référence d'un objet ou void		Q 20.
1	OUI	
2	NON	

En JAVA, pour qu'un objet puisse être passé en paramètre d'une méthode, il faut qu'il soit différent de null		Q 21.
1	OUI	
2	NON	

Le code suivant permet d'augmenter la taille du tableau t1 :		Q 22.
<pre>int[] t1 = new int[10]; t1 = {1,2,3,4,5,6,7,8,9,10};  t1 = augmenterTailleTab(t1,100);</pre>		
Avec :		
<pre>public static int[] augmenterTaille(int[] t,int newTaille) {     int[] tmp = new int[newTaille];     for(int i=0;i&lt;t.length;i++) tmp[i]=t[i];     return tmp; }</pre>		
1	OUI	
2	NON	

Il est possible de modifier les caractères de la chaîne de caractère contenu dans un objet de classe String		Q 23.
1	OUI	
2	NON	

Soit le code suivant :		Q 24.
<pre>String str1[] = new StringTokenizer("AA;BB;CC"), ";"); for(String s: str1)     System.out.println(s);</pre>		
Ce code ::		
1	affiche : AA BB CC	
2	n'affiche rien	
3	ne se compile pas correctement	

La classe StringTokenizer :		Q 25.
1	a un constructeur dont la signature est : StringTokenizer(String str, String tokens)	
2	a un constructeur dont la signature est : StringTokenizer()	
3	a un constructeur dont la signature est : StringTokenizer(String str)	

Soit le code JAVA suivant :		Q 26.
<pre>String slue = "un deux trois quatre cinq six sept huit"; StringTokenizer str = new StringTokenizer(slue); while (str.hasMoreTokens()) {     String s = str.nextToken();     s=s+"_"; } Terminal.ecrireString (s);</pre>		
Ce code :affiche :		
un_deux_trois_quatre_cinq_six_sept_huit_		
1	OUI	
2	NON	

Dans la classe String la méthode d'objet <i>void set(int i, char c)</i> change le i-ième caractère d'une chaîne de caractère avec la valeur c.		Q 27.
1	OUI	
2	NON	

Entre autre, une différence entre un tableau java [] et un ArrayList est :		Q 28.
1	il n'y a pas de limite de capacité à mettre des éléments dans un ArrayList	
2	il n'y a pas de limite de capacité à mettre des éléments dans un tableau	

Soit le code JAVA suivant :		Q 29.
<pre> ArrayList&lt;String&gt; tab = new ArrayList&lt;String&gt;(4); tab.add("UN"); tab.add("DEUX"); for(String s : tab){System.out.println(s);} </pre>		
1	Ce code n'est pas correcte (erreur de compilation)	
2	Ce code affiche : UN DEUX	
3	Ce code affiche : UN DEUX null null	

En JAVA, la déclaration d'un tableau comme suit :		Q 30.
<pre> Individu[] tab_ind = new Individu[10] </pre> contient :		
1	10 éléments dont les valeurs sont toutes à <i>null</i>	
2	10 éléments dont les valeurs sont égales au résultat de l'exécution du constructeur <i>Individu()</i>	

Les tableaux java [] ne peut contenir que des données de type primitif (int, double, ...)		Q 31.
1	OUI	
2	NON	

En JAVA, les tableaux de dimension [N] sont indicés de :		Q 32.
1	1 à N	
2	0 à N-1	

Dans la classe String la méthode d'objet <b><i>void replace(String replaced, String new)</i></b> permet de remplacer dans l'objet String passé en paramètre toutes les occurrences de la chaîne <b><i>replaced</i></b> par la chaîne <b><i>new</i></b> .		Q 33.
1	OUI	
2	NON	

Soit le code suivant :		Q 34.
<pre> int tab_int[] = 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 peuvent être remplacés par :		
1	A → 10 B → tab_int[i]	
2	A → tab_int.length B → tab_int[i]	
3	A → tab_int.size() B → tab_int.get(i)	

Soit le code suivant :		Q 35.
<pre>public class Exemple {     public ArrayList&lt;String&gt; arr;     public Exemple(){         arr = new ArrayList&lt;String&gt;();     }     public add(String e)     {         arr.add(e);     } }</pre> <p><u>Dans un programme Java:</u> Exemple ex = new Exemple(); ex.add("EXEMPLE"); Terminal.ecrireStringln(ex.arr.get(0));</p>		
Ce code :		
1	affiche "EXEMPLE"	
2	affiche ""	
3	ne compile pas correctement	

(Tourner la page)

## 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

Soit une classe A qui contient une méthode m (public ou privée) quelconque qui a des paramètres. Enumérez et précisez les éléments du langage Java que la méthode peut utiliser dans son corps de méthode.

### Q 3

Citez les 3 différences importantes qu'il existe entre l'utilisation d'un tableau Java [] et d'un ArrayList<>

**Fin de la 1<sup>ère</sup> partie**

---

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

---

### Exercice [10 points]

Ecrire le code **complet** de la méthode statique qui prend en entrée un `ArrayList<RendezVous>`, un intervalle d'heure (heure de début au format "HH:MM", heure de fin au format "HH:MM"); et qui retourne les rendez-vous qui sont en intersection avec cet intervalle.

La classe `RendezVous` vous est donnée et contient les méthodes :

```
public String getHeureDebut()  
public String getHeureFin()
```

L'heure retournée par ces deux méthodes est au format "HH:MM".

### Probleme [25 points]

On se propose de créer une classe `Agenda` qui contient des rendez-vous.

La classe `RendezVous` vous est donnée. Elle est définie par les méthodes publiques suivantes:

```
public RendezVous(String date,           // "JJ/MM/AAAA"  
                  String heureDebut,    // "HH:MM"  
                  String heureFin,     // "HH:MM"  
                  String libelle); // Le constructeur  
public String getDate(); //Date du rendez-vous  
public String toString(); //Le rdv en chaine
```

NB: La date est toujours renseignée au format JJ/MM/AAAA

On vous donne la méthode suivante de la classe `Agenda` :

```
public static Calendar dateStringToCalendar(String date)
```

Cette méthode convertit une date qui est au format "JJ/MM/AAAA", en une date `Calendar` (classe Java du package `java.util`)

On vous donne la méthode suivante de la classe `Agenda` :

```
public static ArrayList<RendezVous> lireFichierAgenda(String  
fichier);
```

Cette méthode retourne tous les rendez-vous qui sont lus depuis un fichier texte.

---

On veut dorénavant (par rapport au projet de cette année) que, dans la classe `Agenda`, les rendez-vous soient stockés par numéro de semaine.

On crée donc l'attribut suivant dans la classe `Agenda` contenant tous les rendez-vous.

```
private RendezVousSemaine[] lesRendezVous;
```

Ce tableau est dimensionné à 54. L'indice 1 correspond à la semaine 1. L'indice 0 n'est donc pas utilisé. Ainsi chaque case du tableau contient un objet de type `RendezVousSemaine` qui contient dans une liste, les rendez-vous d'une semaine.

**1/** Ecrire la classe `RendezVousSemaine` et le **constructeur** de la classe `Agenda` qui initialise le contenu de l'attribut `lesRendezVous` à partir du fichier "data/Agenda.txt".

**2/** Ecrire la méthode de la classe `Agenda` qui retourne les rendez-vous qui sont dans une date donnée :

```
public String getDateRendezVous(String date)
```

Cette méthode retourne sous la forme d'une chaîne de caractère la concaténation de tous ces rendez-vous. (afin de les afficher à l'écran par exemple).

**3/** Ecrire la méthode de la classe Agenda qui retourne les rendez-vous qui sont dans un numéro de semaine donnée :

```
public String getDateRendezVous(int numeroSemaine)
```

Cette méthode retourne sous la forme d'une chaîne de caractère la concaténation de tous ces rendez-vous. (afin de les afficher à l'écran par exemple).

NB:

La classe Calendar contient la méthode `public int get(Calendar.WEEK_OF_YEAR)` qui retourne le numéro de la semaine (de 1 à 53) d'une date Calendar.

**(Fin du sujet)**