

IPST-CNAM  
Programmation JAVA  
NFA 001  
Mercredi 13 Février 2013

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
  - -½ pour chaque réponse fausse
- **les questions à 3 propositions** dont 1 seule proposition est fausse
  - + ½ 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:
------	---------

Un langage de programmation orienté objet est :		Q 1
1	un langage dont les données créées et utilisées sont des paquets d'information décrites par un cadre de description appelé une "classe"	
2	un langage permettant de créer des objets ou instance de classe	
3	un langage spécifique permettant de créer des solutions internet	

Dans un langage orienté objet, comme Java, les traitements informatiques sont portés par les classes et les objets du langage		Q 2
1	OUI	
2	NON	

Le compilateur Java (javac) permet de créer un exécutable qui ne s'exécute que sur le type de machine sur laquelle le programme a été compilé		Q 3
1	OUI	
2	NON	

Pour exécuter un programme Java il est nécessaire, entre autre, d'installer et utiliser une JVM		Q 4
1	OUI	
2	NON	

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 5
1	OUI	
2	NON	

Soit une classe contenant les méthodes m1 et m2. m1 est une méthode static et m2 n'est pas une méthode static :		Q 6
1	la méthode m1 static peut utiliser les attributs non statics de la classe	
2	la méthode m2 non static peut utiliser les attributs statics de la classe	
3	la méthode m1 static peut utiliser les attributs statics de la classe	

Dans la programmation objet, en JAVA, le rôle du constructeur d'une classe est de :		Q 7
1	affecter les valeurs des attributs de la classe	
2	allouer en mémoire du programme les attributs de l'objet	
3	construire la classe (ou .class) qui permet à un autre programme de créer les objets de la classe	

En Java, l'exécution d'un programme peut se faire d'autant de façons différentes qu'il existe de méthode main dans les classes du programme		Q 8
1	OUI	
2	NON	

La signature d'une méthode <b>main</b> de la class <b>C1</b> permettant l'exécution d'un programme JAVA est de la forme <b>public static void main(String... args)</b> . Cette méthode est static parce la commande <b>java C1 a1 a2</b> consiste à demander à la JVM Java d'exécuter l'instruction JAVA suivante : <b>C1.main(a1,a2)</b>		Q 9
1	OUI	
2	NON	

Soit le code suivant :		Q 10
<pre> Livre l = new Livre(); l.nom = "Les cavernes d'acier"; ArrayList&lt;Livres&gt; livres = new ArrayList&lt;Livres&gt;(); livres.add(l); l.nom="Face aux feux du soleil"; System.out.println(livres.get(0).nom); </pre>		
Ce code affiche :		
1	Les cavernes d'acier	
2	Face aux feux du soleil	

Un attribut protected d'une classe C1 appartenant au package P est un attribut qui est visible depuis :		Q 11
1	une classe C2 qui n'hérite pas de C1 et qui appartient à un autre package	
2	une classe C3 qui hérite de C1 et qui appartient à un autre package	

Un attribut privé d'une classe C1 est quand même visible depuis les classes qui héritent de C1		Q 12
1	OUI	
2	NON	

Il est déconseillé de créer des classes dont tous les attributs sont privés et pour laquelle il n'existe aucune méthode permettant de modifier ses attributs car une telle classe n'a pas d'utilité.		Q 13
1	OUI	
2	NON	

Soit le code suivant :		Q 14
<pre> int v=13; boolean premier=true; for(int k=2;k&lt;v;k=k+1)     if (v%k == 0) premier = false; if (premier) System.out.println("PREMIER"); else System.out.println("NON PREMIER"); </pre>		
Ce code :		
1	contient une erreur	
2	affiche "NON PREMIER"	
3	affiche "PREMIER"	

La commande <b>javac</b> C1.java		Q 15
1	crée le fichier C1.class	
2	compile la classe C1 et exécute la méthode main de la classe C1	

En JAVA, un tableau ( tab[] )		Q 16
1	peut contenir des éléments de type primitif	
2	peut contenir des références d'objet JAVA	
3	ne peut pas contenir de références d'objet JAVA	

En JAVA, les méthodes déclarées en dehors d'une classe sont appelées des méthodes statics		Q 17
1	OUI	
2	NON	

Une méthode public d'une classe peut utiliser les attributs privés de la classe		Q 18
1	OUI	
2	NON	

Une méthode static est une méthode dont le contenu (le code) reste inchangé durant tout le temps d'exécution du programme Java		Q 19
1	OUI	
2	NON	

La classe StringBuffer de Java, comme la classe String, ne permet pas de modifier les caractères de la chaîne de caractère		Q 20
1	OUI	
2	NON	

La classe StringTokenizer est une classe qui permet de créer des chaînes de caractères et contient des méthodes permettant de modifier les caractères de la chaîne de caractères.		Q 21
1	OUI	
2	NON	

En Java, il est possible de modifier le contenu d'un tableau passé en paramètre d'une méthode		Q 22
1	OUI	
2	NON	

En Java, il est possible de modifier le contenu d'un objet passé en paramètre d'une méthode		Q 23
1	OUI	
2	NON	

En JAVA, un objet est un pointeur		Q 24
1	OUI	
2	NON	

Si une classe B qui hérite d'une classe A et si B n'a pas de constructeur alors :		Q 25
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(s) constructeur(s) hérité(s) de A	
3	Il n'est pas possible de créer un objet de B car la classe B n'a pas de constructeur	

La boucle <b>for</b> dite "énumérative" permet :		Q 26
1	d'incrémenter une valeur scalaire de type enum et de réaliser un traitement à chaque valeur	
2	de parcourir tous les éléments d'une instance de ArrayList	
3	de parcourir tous les éléments d'un tableau java ( tab[] )	

Soit la classe suivante :		Q 27
<pre> public class C1{     private int x;     public C1(int x){ this.x = x; } } </pre>		
L'instruction suivante : C1 c1 = new C1(); est valide et la valeur de x est la valeur par défaut de Java 0		
1	OUI	
2	NON	

Soit la classe suivante :		Q 28
<pre>public class C1{     private int nb;     private ArrayList&lt;String&gt; liste;     public C1(int nb){ this.nb = nb; }     public void add(String s){liste.add(s);} }</pre>		
Le code suivant s'exécute sans erreur		
<pre>C1 c1 = new C1(0); c1.add("TOTO");</pre>		
1	OUI	
2	NON	

En JAVA, une exception est un objet à part entière qui est, notamment, une instance de la classe prédéfinie <b>Exception</b>		Q 29
1	OUI	
2	NON	

Le code suivant est correct :		Q 30
<pre>public void action(int parametre) {     if (parametre==0)         throw new Exception("Erreur");     else         faireLeTraitement(); }</pre>		
1	OUI	
2	NON	

Le code suivant est correct :		Q 31
<pre>public void action(int parametre) {     if (parametre==0)         throw new RuntimeException("Erreur");     else         faireLeTraitement(); }</pre>		
1	OUI	
2	NON	

Soit le fichier suivant C:\CodeJava\exercices\cnam\util\Terminal.java. Le fichier Terminal.java contient en 1 <sup>ère</sup> ligne : <b>package exercices.cnam.util;</b> Le répertoire C:\bin est vide. Dans C: on réalise la commande suivante : <b>javac -d C:\bin C:\CodeJava\exercices\cnam\util\Terminal.java</b> On obtient le résultat suivant : C:\bin\exercices\cnam\util\Terminal.class		Q 32
1	OUI	
2	NON	

<p>Soit le fichier suivant C:\CodeJava\exercices\cnam\util\Terminal.java. Le fichier Terminal.java contient en 1<sup>ère</sup> ligne : <b>package exercices.cnam.util;</b> Dans C:\CodeJava\programme se trouve le fichier Prog.java suivant:</p> <pre> import exercices.cnam.util; public class Prog {     public static void main(String... args) {         Terminal.ecrireStringln("Bonjour");     } } </pre> <p>On est dans le répertoire C:\CodeJava\programme, et on veut compiler le programme. Quelle(s) commande(s) est(sont) valide(s) :</p>		Q 33
1	javac -classpath "." Prog.java	
2	javac Prog.java	
3	javac -classpath ".." Prog.java	

<p>En JAVA, l'instruction suivante permet de déclencher une exception</p> <pre> throw new Exception("Impossible de faire l'action"); </pre>		Q 34
1	OUI	
2	NON	

<p>Le code suivant permet d'augmenter la taille du tableau t1 :</p> <pre> int[] t1 = new int[10]; t1 = {1,2,3,4,5,6,7,8,9,10};  augmenterTailleTab(t1,100); </pre> <p>Avec :</p> <pre> public static void augmenterTaille(int[] t,int newTaille) {     int[] tmp = new int[newTaille];     for(int i=0;i&lt;t.length;i++) tmp[i]=t[i];     t = tmp; } </pre>		Q 35
1	OUI	
2	NON	

(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

Expliquer ce qu'est la notion d'héritage dans le cadre des langages de programmation orientée objet.  
Donnez un exemple type et commentez.

### Q 2

En JAVA, il existe la classe Exception et la classe RuntimeException.  
Expliquez précisément la différence qu'il existe dans l'utilisation de ces deux classes.  
Donner une définition d'une exception.

### Q 3

En JAVA, quels sont les rôles du constructeur ?  
Dans le langage Java expliquez la contrainte qui existe dans l'utilisation du constructeur par défaut et du constructeur sans paramètre.

---

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

---

### Exercice [15 points]

En utilisant la classe Terminal, faire le programme complet JAVA qui permet de saisir des objets de classe Cercle et de calculer le périmètre ( $2 \cdot \text{PI} \cdot \text{rayon}$ ) de chacun des cercles saisis. Un Cercle est défini par les coordonnées du centre (x,y), et un rayon. Tous les attributs de la classe Cercle sont privés.

Le programme principal est la méthode main d'une autre classe : Exercice.

La classe Terminal et Cercle sont dans le package : cnam.util  
La classe Exercice est dans le package : cnam.exos

1/ Ecrire les classes Cercle et Exercice.

2/ Donnez les commandes de compilation et d'exécution de ce programme.

La commande de compilation est lancée dans le répertoire "cnam".

Les fichiers .class sont générés dans le répertoire « bin » qui se trouve à côté du répertoire « cnam ».

La commande d'exécution est lancée depuis le répertoire « bin ».

Nous avons donc l'arborescence suivante :

```
cnam/exos/Exercice.java
cnam/util/Cercle.java
cnam/util/Terminal.java
cnam/
  <commande de compilation>
bin/
  <commande d'exécution>
```

(Tourner la page)

**Exercice [25 points]**

Soit un fichier "compte.txt" contenant des lignes d'écriture de débits et crédits d'un compte bancaire.

La méthode suivante de la classe Terminal vous est donnée et lit le contenu du fichier et retourne le contenu du texte sous la forme d'un tableau dont chaque élément est une ligne du fichier texte :

```
public static String[] lireFichierTexte(String nomFichier) ;
```

Le fichier est structuré de la manière suivante.

La première ligne qui est le solde initial en débit ou crédit du compte (« valeur » est positif), est de la forme :

"SOLDE CREDIT *valeur*" ou

"SOLDE DEBIT *valeur*"

Les autres lignes sont de la forme

"DEBIT *valeur nature*" ou "CREDIT *valeur nature*"

Chaque ligne est une "écriture" bancaire (« valeur » est positif)

Exemple :

```
SOLDE      CREDIT      2345.89
DEBIT      423.56      loyer
DEBIT       63.5       essence
CREDIT    1234.56      salaire
DEBIT       56.7       restaurant
CREDIT     300.0       pension
DEBIT       53.5       essence
```

On se propose de réaliser les traitements suivants :

- le calcul du solde du compte
- le calcul de la somme de toutes les valeurs d'une écriture en particulier.

1/ Ecrire en JAVA la classe **CompteBancaire** (qui peut contenir une classe interne **Ecriture**) contenant les méthodes :

- public double calculerSolde();
- public double calculerSommeEcriture(String nature);

Le constructeur de la classe CompteBancaire prend en entrée le nom du fichier.

Le contenu du compte bancaire est chargé en mémoire de la classe dans le constructeur.

Si le fichier contient une erreur de syntaxe le constructeur retourne une exception.

2/ Une famille gère 3 comptes bancaires :

COMPTE\_JOINT.txt

COMPTE\_EPARGNE.txt

LIVRET\_A.txt

Ecrire la méthode main qui utilise la classe Comptebancaire et qui affiche la synthèse des comptes de la famille : solde de chaque compte et solde total cumulé des 3 comptes.

Si un des fichiers a un problème alors le programme affiche un texte d'erreur et fait le traitement sur les autres comptes.

**(Fin du sujet)**