

# Licence d'informatique 2003-2004

## Langage Java

### Contrôle des connaissances - avril 2004

#### Durée de l'épreuve : 1 h 30

L'usage des photocopiés du cours est autorisé. Tous les autres documents, sous quelque forme que ce soit (papier, électronique ou autre) sont interdits, y compris les énoncés et corrigés des exercices des TD.

Le barème est donné entre crochets au début de chaque question.

1 [1 pts] Soit une classe A dont le code est :

```
public abstract class A {  
    public abstract int m();  
    public int n() {  
        return 2 * m();  
    }  
}
```

Répondez par oui ou par non pour chacune des questions suivantes et donnez une courte explication de votre réponse :

- (a) Est-ce que la méthode n doit être déclarée abstraite? Sinon, dites pourquoi.
- (b) Est-ce que le code suivant compile? Sinon, dites pourquoi.

```
A a = new A();  
int i = a.n();
```

- (c) Soit p une méthode définie dans une classe B, avec l'en-tête suivante :  
`public A p(int i)`. Soit b une instance de la classe B.

Est-ce que le code suivant compile? Sinon, dites pourquoi.

```
A a = b.p(3);  
int i = a.m();
```

2 [1 pt] Pour que vous puissiez utiliser la classe `java.util.ArrayList` dans le code d'une de vos classes, cette classe doit nécessairement comporter le code :

- (a) `import java.util;`
- (b) `import java.util.*;`
- (c) rien n'est nécessaire pour utiliser `java.util.ArrayList`.

3 [2 pts] Soient A une classe Java et B une classe fille de A, et les 2 déclarations suivantes :

```
A a;  
B b;
```

Entourez les lignes qui ne compilent pas ; pour chaque ligne qui ne compile pas, donnez la raison.

- (a) `a = new B();`
- (b) `b = new A();`
- (c) `a = (A)new B();`
- (d) `b = (B)new A();`

Entourez les lignes qui compilent mais qui provoquent une erreur à l'exécution ; pour chaque ligne que vous aurez donnée, donnez la raison de l'erreur :

- (a) `a = new B();`
- (b) `b = new A();`
- (c) `a = (A)new B();`
- (d) `b = (B)new A();`

4 [1 pts] Voici 2 classes (placées dans 2 fichiers différents).

```
01 :public class A {
02 :   public int m() {
03 :     return 1;
04 :   }
05 :
06 :   public String toString() {
07 :     return "A";
08 :   }
09 :}
10 :
11 :import java.util.*;
12 :public class TestA {
13 :   public static void main(String[] args) {
14 :     List liste = new ArrayList();
15 :     liste.add(new A());
16 :     liste.add(new A());
17 :     for (int i = 0; i < liste.size(); i++) {
18 :       System.out.println(liste.get(i).m());
19 :     }
20 :   }
21 :}
```

Entourez la bonne réponse, et répondez sous les questions à la question correspondante (répondez par exemple “ligne 03; Correction : return 3; Affiche 3”):

- (a) Ce code ne compile pas. Donnez le numéro de la ligne qui empêche la compilation. Corrigez la ligne. Dites ce qui sera affiché par la méthode `main`.
- (b) Ce code compile mais provoque une erreur à l’exécution. Donnez le numéro de la ligne qui provoque l’erreur. Corrigez l’erreur. Dites ce qui sera affiché par la méthode `main`.
- (c) Ce code s’exécute sans erreur. Dites ce qui sera affiché par la méthode `main`.

5 [1 pts] Voici 2 classes (placées dans 2 fichiers différents).

```
01 :public class A {
02 :   public int m() {
03 :     return 1;
04 :   }
05 :
06 :   public String toString() {
07 :     return "A";
08 :   }
09 :}
10 :
11 :public class TestA {
12 :   public static void main(String[] args) {
13 :     java.util.List liste = new java.util.ArrayList();
14 :     liste.add(new A());
15 :     liste.add(new A());
16 :     String s = "";
17 :     for (int i = 0; i < liste.size(); i++) {
18 :       s += liste.get(i).toString();
19 :     }
20 :     System.out.println(s);
21 :   }
22 :}
```

Entourez la bonne réponse, et répondez à la question correspondante :

- (a) Ce code ne compile pas. Donnez le numéro de la ligne qui empêche la compilation. Corrigez la ligne. Dites ce qui sera affiché par la méthode `main`.
- (b) Ce code compile mais provoque une erreur à l'exécution. Donnez le numéro de la ligne qui provoque l'erreur. Corrigez l'erreur. Dites ce qui sera affiché par la méthode `main`.
- (c) Ce code s'exécute sans erreur. Dites ce qui sera affiché par la méthode `main`.

6 [2 pts] 2 classes dans 2 fichiers séparés :

```
01 :public class A {
02 :   public A(int i) {
03 :     System.out.println(i);
04 :   }
05 :   public void m() {
06 :     System.out.println("A");
07 :   }
08 :}
09 :
10 :public class B extends A {
11 :   public void m() {
12 :     System.out.println("B");
13 :   }
14 :
15 :   public static void main(String[] args) {
16 :     A a = new B();
17 :     a.m();
18 :   }
19 :}
```

On veut lancer l'exécution de la classe B.

Entourez la bonne réponse et répondez à la question correspondante.

- (a) Ce code ne compile pas; dire quelles lignes provoquent une erreur et la raison de l'erreur.
- (b) Ce code ne compile mais il y a une erreur à l'exécution; dire sur quelle ligne et la raison de l'erreur.
- (c) Il s'affiche quelque chose (A, B, par exemple) : dans ce cas, dire exactement ce qui s'affiche.
- (d) Rien ne s'affiche à l'exécution.

7 [2 pts] Même question qu'à l'exercice précédent mais on remplace à présent la classe B de l'exercice précédent par ce code :

```
10 :public class B extends A {
11 :   public B (int i) {
12 :       super(i);
13 :       System.out.println(i*2);
14 :   }
15 :
16 :   public void m() {
17 :       System.out.println("B");
18 :   }
19 :
20 :   public static void main(String[] args) {
21 :       A a = (A)new B(2);
22 :       a.m();
23 :   }
24 :}
```

8 [2 pts] 3 classes dans 3 fichiers séparés :

```
01 :public class A {
02 :   public String m() {
03 :       return "A";
04 :   }
05 :}
06 :
07 :public class B extends A {
08 :   public String m() {
09 :       return "B";
10 :   }
11 :}
12 :
13 :public class Test {
14 :   public static void main(String[] args) {
15 :       A a = new A();
16 :       B b = new B();
17 :       A c = new B();
18 :       System.out.println(a.m() + b.m() + c.m() + ((A)b).m());
19 :   }
20 :}
```

On veut lancer l'exécution de la classe `Test`.

Choisissez le numéro de la bonne réponse et répondez à la question correspondante.

- (a) Ce code ne compile pas; dire quelles lignes provoquent une erreur et la raison de l'erreur.
- (b) Il y a une erreur à l'exécution; dire sur quelle ligne et la raison de l'erreur.
- (c) Il s'affiche quelque chose; dans ce cas, dire exactement ce qui s'affiche.
- (d) Rien ne s'affiche à l'exécution.

9 [8 pts] Vous allez compléter la classe `Etagere` étudiée pendant les TPs. On suppose que les classes sont dans le paquetage `fr.unice.bibliotheque`.

Rappels :

– La classe `Etagere` a 2 variables :

```
private Livre[] livres; // contient les livres de l'étagère
private int nbLivres; // nombre de livres de l'étagère
```

– La classe `Livre` a 2 méthodes

```
public String getAuteur()
public String getTitre()
```

Ajoutez à la classe `Etagere` une méthode `rechercherLivresDesAuteurs` dans la classe `Etagere`. Cette méthode recherche les livres dont les auteurs sont passés en paramètres dans un tableau complètement rempli (aucune case du tableau n'est vide). La méthode renvoie le résultat de sa recherche dans un tableau complètement rempli. **IMPORTANT : vous n'utiliserez aucune méthode déjà écrite dans la méthode `rechercherLivresDesAuteurs`.**

La méthode devra lancer une exception contrôlée par le compilateur, de type `PasDeLivreTrouveException`, dans le cas où aucun titre n'est trouvé. Vous donnerez le code de cette classe qui sera dans le même paquetage que `Etagere`. Cette classe sera très simple et ne comportera que 2 constructeurs : un sans paramètre et un avec un paramètre de type `String`.

Écrivez une classe `Test` dont la méthode `main` teste la méthode `rechercherLivresDesAuteurs` en affichant les titres et auteurs de tous les livres trouvés. La méthode `main` affiche "Pas de livre trouvé" si aucun titre n'est trouvé. Cette classe ne doit pas être dans le paquetage `fr.unice.bibliotheque`. Vous pourrez utiliser un constructeur pour la classe `Etagere` ainsi qu'une méthode `ajouter` pour ajouter un livre dans une étagère, sans avoir besoin d'écrire leur code.