

IFT1025 E10 — Examen Intra

Miklós Csűrös

15 juin 2010, 15 :30–17 :30

Aucune documentation n'est permise. L'examen vaut 100 points et comprend 8 exercices.

Exercice	0 (nom)	1 (vrai ou faux)	2 (polymorphisme)	3 (différence)	4 (lignes)	5 (héritage)	6 (exceptions)	7 (horloge)
Points	1	20	9	15	15	10	10	20

Répondez à toutes les questions dans les cahiers d'examen.

BONNE CHANCE !

0 Votre nom (1 point)

Écrivez votre nom et code permanent sur tous les cahiers soumis.

1 Interfaces : vrai ou faux (20 points)

1. Une interface peut déclarer des méthodes `finales` .
2. Une interface peut contenir des variables `finales` .
3. Si classe `A` est la sous-classe de `B` et `B` implante l'interface `C`, alors `A` implante l'interface `C` aussi.
4. Il est permis de déclarer une variable du type d'une interface (p.e., `Comparable c`).
5. Une classe ne peut implanter qu'une interface à la fois.
6. Il est permis de déclarer des constructeurs dans une interface.
7. Une interface peut être la sous-interface (`extends`) de plus qu'une interface à la fois.
8. Il est permis de déclarer des méthodes statiques dans une interface.
9. Une interface peut avoir une variable statique de type tableau (disons, `int []`).
10. Toutes les exceptions implament l'interface `java.lang.Exception`.

2 Polymorphisme (9 points)

A. (3 points) Considérez la déclaration de classe suivante.

```
public class Panneau extends JPanel implements MouseListener
```

Quels types peuvent remplacer T dans la déclaration suivante sans erreur de syntaxe? (Énumérez au moins 3 possibilités.)

```
T panneau = new Panneau();
```

B. (6 points) Supposons que la classe `Sandwich` implante l'interface `Edible`, et on a les déclarations.

```
Sandwich jambonetbrie = new Sandwich();  
Rectangle canape = new Rectangle(0,0,10,10);  
Edible gouter = null;
```

Lesquels des affectations suivantes sont légales?

- a. `gouter = jambonetbrie;`
- b. `jambonetbrie = gouter;`
- c. `jambonetbrie = (Sandwich) gouter;`
- d. `gouter = canape;`
- e. `gouter = (Edible) canape;`
- f. `gouter = (Rectangle) canape;`

3 Différence (15 points)

Expliquez la notion de *différance* chez Derrida, **ou** écrivez le code d'une méthode

```
static double difference(ArrayList<Integer> A)
```

qui trouve la plus grande différence entre des paires dans la liste A : $\max_{p,q \in A} |p - q|$. Lancez une exception `IllegalArgumentException` (sous-classe de `RuntimeException`) si A est `null` ou si elle contient moins que deux éléments.

4 Lignes (15 points)

Écrivez le code d'une méthode

```
public static int nbLignes(String fichier)
```

pour compter le nombre de lignes dans un fichier texte nommé `fichier`. Votre méthode ne doit pas lancer des exceptions `IOException` : s'il y a un problème, il faut retourner la valeur `-1`.

5 Héritage (10 points)

A. (5 points) Considérez la classe suivante.

```
class Demeter
{
    public Demeter(String s){this.d=s;}
    private static int a=1;
    private int b=1;
    protected double c=1.0;
    public String d="1";
    public static String e="1";
    private int getA(){return a;}
    protected int getB(){return b;}
    public String getD(){return d;}
}
```

Quels variables/méthodes/constructeurs sont hérités dans une sous-classe de Demeter parmi les suivants ?

- ★ constructeur Demeter (String)
- ★ variable c
- ★ variable d
- ★ méthode getA()
- ★ méthode getD()

B. (5 points) Qu'est-ce qui est affiché lors de l'exécution du code suivant ?

```
class Persephone extends Demeter
{
    public Persephone(){super("2");}
    private int b=3;
    public String d=e;
    protected int getB(){return super.getB();}
    public String getD(){return d;}
    public static void main(String[] args)
    {
        Persephone P = new Persephone();
        System.out.println(P.getB()+" "+P.getD());
    }
}
```

6 Exceptions (10 points)

A. (5 points) Considérez le code suivant.

```
static String nEme(ArrayList<String> S, ArrayList<Integer> I, int n)
{
    try
    {
        int idx = I.get(n);
        return S.get(idx);
    } catch (ArrayIndexOutOfBoundsException E)
    {
        throw new IllegalArgumentException("Argument n erronné");
    }
}
```

Il y a au moins un problème avec cette implantation : identifiez lequel ou lesquels.

1. `ArrayIndexOutOfBoundsException` ne peut pas être capturée par `catch` parce qu'elle est une `RuntimeException`.
2. Il manque la branche `finally`.
3. La logique devrait être implantée par la comparaison directe de l'argument `n` avec les bornes de `S` et `I` au lieu de capturer une exception.
4. Il n'est pas permis de lancer une exception dans un bloc `catch`.
5. La construction `try ... catch` n'est pas permise dans un contexte statique.

B. (5 points) Considérez le code suivant où `E` est une sous-classe de `Exception`.

```
try {f();} catch (Exception e){if (e instanceof E){g();}}
```

Quelle version ci-dessous est équivalente ?

1. `try{f();} catch (Exception e){} catch (E e){g();}`
2. `try{f();} catch (E e){g();} catch (Exception e){}`
3. `try{f();} catch (E e){g();}`
4. `try{f();} catch (Exception e){} finally {g();}`
5. Aucune des quatre versions n'est équivalente à l'originale.

7 Horloge (20 points)

Écrivez le code d'une classe `Montre` pour afficher une montre à aiguilles (heure, minute et seconde). Définissez une sous-classe de `JPanel` et servez-vous d'un `Timer` pour faire tourner les aiguilles (stocker le temps dans des variables locales). Constructeur : `Montre(int h, int m, int s)` initialise le temps à h heures m minutes et s secondes. L'interface graphique doit afficher la silhouette de la montre ainsi que les trois aiguilles. **Indice** : syntaxe de quelques méthodes qui peuvent être utiles.

Classe	Signature de méthode ou de constructeur
<code>java.awt.Graphics</code>	<code>void drawLine(int x1, int y1, int x2, int y2)</code>
<code>java.awt.Graphics</code>	<code>void drawOval(int x1, int y1, int largeur, int hauteur)</code>
<code>javax.swing.Timer</code>	<code>Timer(int delai, ActionListener listener); void start()</code>
<code>java.awt.ActionListener</code>	<code>void actionPerformed(ActionEvent e)</code>
<code>java.lang.Math</code>	<code>static double sin(); static double cos();</code>