

Examen de Langage Java

Durée : 2h

Aucun document autorisé

Mobiles interdits Note : la qualité des commentaires, avec notamment la présence d'affirmations significatives, ainsi que les noms donnés aux variables, et la bonne indentation rentreront pour une part importante dans l'appréciation du travail.

Soient les déclarations de classes et interface Java suivantes :

```
abstract class ClassA {
    public void methode(){System.out.println(1);}
}
class ClassB extends ClassA {
}
abstract class ClassC extends ClassA implements Interface1 {
    public void methode(){System.out.println(2);}
    public void methode(int x){System.out.println(3);}
}

class ClassD extends ClassC {
    private void methode(){System.out.println(4);}
}
class ClassE extends ClassC {
    public void methode(String s){System.out.println(5);}
    public void methode(int x){System.out.println(6);}
}
interface Interface1 {
    public void methode(int x);
}
```

- 1. Dessinez le graphe d'héritage des classes et de l'interface précédentes.
- 2. Pour chacune des lignes du code suivant, indiquez s'il y a une erreur, d'où vient l'erreur et, si la ligne est censée afficher quelque chose, indiquez ce qu'elle affiche.

```
1 : ClassA c1 = new ClassB();
2 : ClassA c2 = new ClassC();
3 : ClassD c3 = new ClassD();
4 : ClassB c4 = new ClassD();
5 : ClassC c5 = new ClassE();
6 : c1.methode();
7 : c2.methode();
8 : c3.methode();
9 : c3.methode(3);
10 : c4.methode();
11 : c5.methode();
12 : c5.methode(3);
```

Tout d'abord, la déclaration de la méthode dans la classe D ne peut être **private** (redéfinition avec un accès plus faible interdite).

```
1 : ClassA c1 = new ClassB();
2 : ClassA c2 = new ClassC(); //classe abstraite
```

```
3 : ClassD c3 = new ClassD();
4 : ClassB c4 = new ClassD(); //types incompatibles
5 : ClassC c5 = new ClassE();
6 : c1.methode(); //1
7 : c2.methode(); //erreur à cause la déclaration erronée de c2
8 : c3.methode(); //4
9 : c3.methode(3); //3
10 : c4.methode(); //erreur à cause de la déclaration erronée de c4
11 : c5.methode(); //2
12 : c5.methode(3); //6
```

Un fichier est composé d'une suite de trames. Le nombre de trames est quelconque et peut-être égal à 0. Chaque trame est formée d'un entier (type **int** ≥ 0), suivi de réels (type **double**). Le nombre de réels dans la trame est égal à la valeur de l'entier, et on appellera longueur d'une trame ce nombre.

- 3. Écrivez la fonction `nbTrames` qui renvoie le nombre de trames de longueur `lg` contenues dans un fichier de trames. Le nom du fichier de trames, et la longueur des trames recherchées sont donnés en paramètre. On considérera que les trames sont correctement formées. Vous pourrez utiliser les classes `DataInputStream`, `FileInputStream`, `EOFException`, `FileNotFoundException` et `IOException`. L'en-tête de la méthode est le suivant :

```
static int nbTrames(String f, int lg)

/**
 * Antécédent : f nom du fichier de trames
 *              lg>=0 longueur des trames recherchées
 *
 * Rôle : renvoie le nombre de trames de longueur lg
 *        contenues dans le fichier
 */
static int nbTrames(String f, int lg) {
    int n = 0;
    try (DataInputStream dis = new DataInputStream(new FileInputStream(f)))
    {
        while (true) {
            //lire l'entier
            int x = dis.readInt();
            if (x==lg)
                //on a une nouvelle trame de longueur lg
                n++;
            //lire les x réels
            for (int i=0; i<x; i++) dis.readDouble();
        }
    }
    catch (EOFException e) {}
    catch (FileNotFoundException e) { /* traiter l'erreur */ }
    catch (IOException e) { /* traiter l'erreur */ }
    finally { return n; }
}
```

- 4. Écrivez la classe `BoutonRB` qui superpose dans une `JFrame`, un `JButton` et un `JPanel`. La première fois qu'on clique sur le bouton, la couleur de fond du panel devient bleu, puis les clics suivants, on alterne entre la couleur rouge et la couleur bleu.

```

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class BoutonRB extends JFrame implements ActionListener
{
    private JButton bt = new JButton("press");
    private JPanel p = new JPanel();
    private boolean bleu = false;

    //Rôle : construit une fenêtre avec les 2 boutons bleu et rouge
    public BoutonRB() {
        //placer le label et le bouton
        getContentPane().setLayout
            (new BorderLayout(getContentPane(), BorderLayout.PAGE_AXIS));
        getContentPane().add(bt);
        getContentPane().add(p);
        //ajouter les auditeurs
        bt.addActionListener(this);
        pack();
        setVisible(true);
    }
    //Rôle : inverse la couleur de label
    public void actionPerformed(ActionEvent e) {
        p.setBackground((this.bleu ? Color.red : Color.blue);
        this.bleu = !this.bleu;
    }
}

```

.....