

Durée 1 heure 30 minutes
Aucun Document Autorisé

Les parties I, II sont indépendantes.
Le barème est donné à titre indicatif.

Partie I, 8 points

Cette partie est composée de questions de cours. Pour chaque question vous devez argumenter et proposer un petit programme exemple.

Question 1 (2 points) : Expliquez les deux utilités majeures (vu en cours) du mécanisme de paquets de Java.

Question 2 (2 points) : Il existe 4 types de droits d'accès en Java : **private**, **public**, **protected** et **rien**. Expliquez le rôle de chacun d'eux.

Question 3 (2 points) : En Java il existe des types primitifs et non primitifs. Expliquez leurs différences de fonctionnement.

Question 4 (2 points) : Que fait le programme suivant ?

```
1 class A {  
2     public String f () {  
3         return "A";  
4     }  
5 }  
6 class B extends A {  
7     public B (A a) {  
8         -a = a;  
9     }  
10    public String f () {  
11        return super.f () + "B" + a.f ();  
12    }  
13    private A -a;  
14 }  
15 public class C extends B {  
16     private static final A quesako = new B (new C (new A ( )));  
17     public C (A a) {  
18         super (a);  
19     }  
20    public String f () {  
21        return "C" + super.f ();  
22    }  
23    public static void main (String argv[]) {  
24        System.out.println (C.quesako.f ());  
25    }  
26 }
```

Partie II Calcul numérique en Java, 12 points

L'objectif de cette partie est de modéliser des fonctions de type $y = f(x)$ à l'aide de la programmation orientée objet. Les fonctions que nous étudierons sont des fonctions de \mathcal{R} dans \mathcal{R} . Les services qu'une de nos fonctions doit offrir sont les suivants :

- **value** : retourne le résultat de l'évaluation de la fonction pour la valeur x .
 - **derivative** : retourne le résultat de l'évaluation de la dérivée de la fonction en x .
 - **integral** : retourne le résultat de l'évaluation de l'intégrale de la fonction entre deux bornes a et b .
 - **toString** retourne une description de la fonction sous forme de chaîne de caractère.
 - **clone** duplique la fonction.
- Toutes les méthodes d'évaluation pourront lever l'exception suivante s'il y a un problème lors de l'évaluation (par exemple : évaluation de $1/x$ quand x vaut 0).

```
1 public class ImpossibleEvaluationException extends Exception {  
2     public ImpossibleEvaluationException (double x) {  
3         super ("Evaluation impossible pour x = " + x);  
4     }  
5 }
```

Question 5 (2 points) : En Java, écrivez entièrement l'interface **Function** déclarant l'ensemble des services disponibles sur nos fonctions.

Question 6 (4 points) : Sachant que pour toute fonction $f(x)$ on peut approximer numériquement l'intégrale de $f(x)$ entre a et b en utilisant la formule suivante

$$\int_a^b f(x) dx \approx \sum_{i=0}^{precision-1} f\left(a + \frac{(i+0.5)(b-a)}{precision}\right) * \frac{b-a}{precision}$$

et que pour toute fonction $f(x)$ on peut approximer numériquement la dérivée de $f(x)$ avec la formule suivante :

$$f'(x) \approx \frac{f(x+\epsilon) - f(x-\epsilon)}{2*\epsilon}$$

Ecrivez la classe abstraite **AbstractFunction** qui donnera une implémentation par défaut des services d'intégration et de dérivée de notre interface **Function**. Dans votre implémentation, vous considérerez que $\epsilon = \frac{1}{precision}$ et que *precision* sera passé en paramètre à l'instanciation de la classe abstraite.

Question 7 (2 points) : Donnez l'implémentation de la classe **InvFunction** qui permet de modéliser la fonction $f(x) = 1/x$, la valeur de la précision du calcul des approximations sera fixée à 10000. Ecrivez un programme principal (main) dans la classe **InvFunction** qui crée une instance de **InvFunction**, qui affiche l'intégrale de $1/x$ entre 1 et 10, la dérivée de $1/x$ pour $x = 5$ et la valeur de $1/x$ pour $x = 0$. Expliquez ce qu'il va se passer à l'exécution de ce programme ?

tournez la feuille svp

Question 8 (2 points) : Nous disposons d'une hiérarchie de classe existante permettant de calculer des valeurs sur des polynômes (voir ci-dessous). Un polynôme étant une fonction, nous voudrions créer une classe **FunctionPolynomeArray** qui implémente simultanément les interfaces **Function** et **Polynome**. En utilisant la délégation, donnez l'implémentation de la classe **FunctionPolynomeArray**.

```

1 public interface Polynome {
2     void setCoefficient (int i, double coeff);
3     double getCoefficient (int i);
4     double value (double x);
5     String toString ();
6     Polynome clone ();
7 }
8
9
10 public class PolynomeArray implements Polynome {
11     public PolynomeArray (int maxDegree) {
12         coeff = new double [maxDegree];
13         for (int i = 0; i < coeff.length; ++i)
14             coeff[i] = 0;
15     }
16     public void setCoefficient (int i, double c) {
17         coeff[i] = c;
18     }
19     public double getCoefficient (int i) {
20         return coeff[i];
21     }
22     public double value (double x) {
23         double result = 0;
24         for (int i = 0; i < coeff.length; ++i)
25             result += Math.pow (x, i) * coeff[i];
26         return result;
27     }
28     public String toString () {
29         StringBuffer str = new StringBuffer ();
30         boolean prev = false;
31         for (int i = 0; i < coeff.length; ++i) {
32             if (coeff[i] != 0.0) {
33                 str.append (" + ");
34                 str.append (coeff[i] + " * x^" + i);
35                 prev = true;
36             }
37         }
38         return str.toString ();
39     }
40     public Polynome clone () {
41         Polynome tmp = new PolynomeArray (coeff.length);
42         for (int i = 0; i < coeff.length; ++i)
43             tmp.setCoefficient (i, coeff[i]);
44         return tmp;
45     }
46     private double coeff [];
47 }

```

Question 9 (2 points) : Dans le cas des polynômes il n'est pas nécessaire d'utiliser des approximations pour calculer leurs intégrales ou leurs dérivées. En héritant de votre classe **FunctionPolynomeArray**, implémentez une classe **FunctionPolynomArrayOptimized** qui utilise cette propriété.

Pour rappel soit $P(x)$ un polynôme les formules de dérivation et d'intégration sont les suivantes :

$$P(x) = \alpha_0 x^0 + \alpha_1 x^1 + \alpha_2 x^2 + \dots + \alpha_n x^n$$

$$P'(x) = \alpha_1 x^0 + 2.\alpha_2 x^1 + \dots + n * \alpha_n x^{n-1}$$

$$\int P(x)dx = \alpha_0 x^1 + \frac{\alpha_1}{2} x^2 + \frac{\alpha_2}{3} x^3 + \dots + \frac{\alpha_n}{n+1} x^{n+1}$$

(La constante d'intégration de $\int P(x)$ n'est pas présente dans la formule car elle disparaît pendant le calcul de \int_a^b .)