

Master Mathématiques et Informatique 1ère année
Programmation Scientifique Orientée Objet
Examen (3h)
Pas de documents autorisés

Exercice

L'objectif de cet exercice est de créer un ensemble de classes permettant de faire des calculs d'intégrales sur des fonctions. En vous appuyant sur le principe de réutilisabilité de la programmation objet vu en cours, vous devrez créer l'ensemble des classes et méthodes nécessaires à cet objectif et notamment :

1) Une classe abstraite "Fonction" qui possède les attributs suivants:

- deux réels "borneInf" et "borneSup" qui définissent les bornes de définition de la fonction.
- deux booléens "estBorneInfStrict" et "estborneSupStrict" qui indiquent, respectivement, si la borne inférieure et la borne supérieure sont des bornes strictes.

Et les méthodes suivantes :

- une méthode abstraite "f(double x)" : elle a vocation à retourner la valeur de la fonction d'après une valeur x. Elle lève une BorneException si la valeur de x est en dehors de la borne de définition de la fonction.
- une méthode publique "calculIntegral(double borneInf, double borneSup, MethodeIntegrale methode)" : retourne la valeur de l'intégrale entre la borneInf et la borneSup en utilisant la méthode de calcul passée en paramètre. Elle lève une BorneException si borneInf et/ou borneSup sont en dehors de la borne de définition de la fonction ou si borneInf > borneSup.
- une méthode privée "calculMethodeRectangle(double borneInf, double borneSup)" : retourne la valeur de l'intégral entre borneInf et borneSup en effectuant le calcul grâce à la méthode des rectangles.
- une méthode privée "calculMethodeTrapeze(double borneInf, double borneSup)" : retourne la valeur de l'intégral entre borneInf et borneSup en effectuant le calcul grâce à la méthode des trapèzes.

2) Une énumération "MethodeIntegrale" qui propose deux valeurs : une pour la méthode des rectangles, et une autre pour la méthode des trapèzes.

3) Une classe "BorneException" qui permet de lever des exceptions lorsqu'un calcul risque d'être effectué en dehors des bornes de définition d'une fonction.

4) Une classe "FonctionRacineCarre" qui hérite de la classe mère "Fonction". Elle définit "f(double x)" tel que $f(x) = \sqrt{x}$.

5) Une classe "Main" qui test l'ensemble des fonctionnalités.

Annexe

Méthode des rectangles La méthode des rectangles revient à une approximation de f par une fonction en escalier. La valeur approchée R de l'intégrale sur l'intervalle $[a, b]$, et en utilisant n rectangles, vaut alors :

$$R = \frac{b-a}{n} \sum_{i=0}^{n-1} f(x_i)$$

avec $x_0 = a$ et $x_n = b$

Méthode des trapèzes Il s'agit du même principe que la méthode des rectangles mais on remplace ceux-ci par des trapèzes. On utilise une fonction continue affine par morceaux approchant la fonction à intégrer et égale à celle-ci sur les points de la subdivision en n sous-intervalles égaux de l'intervalle d'intégration $[a, b]$ pour obtenir une approximation de la valeur de son intégrale sur $[a, b]$. La valeur approchée R de l'intégrale sur l'intervalle $[a, b]$, et en utilisant n trapèzes, vaut alors :

$$R = \frac{b-a}{n} \left(\frac{f(a) + f(b)}{2} + \sum_{i=1}^{n-1} f(x_i) \right)$$

avec $x_0 = a$ et $x_n = b$

Racine carré La classe `java.util.Math` vous permet d'utiliser la méthode `sqrt(double a)` pour calculer la racine carré d'un nombre.