

---

# TP 1 : RÉ-ÉCHANTILLONNAGE

---

## Introduction

Pour le premier TP nous allons faire du ré-échantillonnage. Votre programme va recevoir en entrées une suite de valeurs qui représente une courbe. Ensuite, le logiciel va ré-échantillonner les points sur la courbe en utilisant une méthode d'approximation. Finalement, la nouvelle séquence de valeurs est affichée sur la console. Les points en entrées sont équidistants sur l'axe des  $x$ , cette distance sera dénotée  $h$ . Les points en sorties seront aussi équidistants, mais la distance entre deux points sera de  $\frac{h}{4}$ .

## Description

### Entrées

Dans un premier temps, votre application devra lire le nom d'un fichier au clavier. Ensuite, il devra lire les entrées dans ce fichier. Le fichier va contenir une valeur par ligne. La première valeur ( $d$ ) sera un entier entre 1 et 3 inclusivement. Il représente le degré de l'équation utilisé pour trouver les valeurs du ré-échantillonnage. Ces équations sont décrites plus loin dans le texte.

La valeur suivante est de type double. Elle représente la distance ( $h$ ) sur l'axe des  $x$  entre deux valeurs consécutives. Elle est suivie d'une autre valeur de type double :  $x_0$ . Cette valeur représente la première valeur de  $x$  qui a été échantillonné.

Finalement, il y aura une suite de  $n$  valeurs, toutes de type double, qui représente les échantillons  $(y_0, y_1, \dots, y_{n-1})$ .

Toute information incorrecte implique l'affichage d'un message d'erreur et la fin du programme.

Exemple de lecture au clavier (vous **ne pouvez pas** utiliser la classe Clavier).

```
Scanner clavier = new Scanner( System.in );  
String nom = clavier.nextLine();  
clavier.close();
```

Exemple de message d'erreur et d'arrêt d'un programme.

```
System.err.println( mssg );  
System.exit( -1 );
```

Voici un code de départ pour la lecture dans un fichier.

```
String nom = "nom.txt";
int x = 0;
double d = 0.0;

try {
    FileReader fr = new FileReader(nom);
    BufferedReader fichier = new BufferedReader(fr);
    Scanner sc = new Scanner(fichier);

    sc.useLocale( Locale.CANADA );

    if( sc.hasNextInt() ) {
        x = sc.nextInt();
    }

    if( sc.hasNextDouble() ) {
        d = sc.nextDouble();
    }

    sc.close();
} catch ( FileNotFoundException e ) {
    // Traitement d'erreur ici.
} catch ( IOException e ) {
    // Traitement d'erreur ici.
}
```

## Calcule

Votre logiciel doit ajouter 3 valeurs ( $y_{i,j} \in \{y_{i,1}, y_{i,2}, y_{i,3}\}$ ) entre chaque valeur de l'axe des  $y$ . Pour cela vous avez besoin de calculer trois valeurs ( $x_{i,j} \in \{x_{i,1}, x_{i,2}, x_{i,3}\}$ ) entre  $x_i$  et  $x_{i+1}$ .

$$x_i = x_0 + i \cdot h$$

$$x_{i,1} = x_i + 0.25h$$

$$x_{i,2} = x_i + 0.5h$$

$$x_{i,3} = x_i + 0.75h$$

Ensuite, il faut trouver l'équation d'approximation pour l'intervalle  $[x_i, x_{i+1}]$ . Il y a trois équations d'approximation possible, dépendant du degré ( $d$ ) demandé par l'utilisateur.

Degré ( $d$ )	Calcule	Polynôme résultant
$d = 1$	$\dot{\delta}_i = y_{i+1} - y_i$ $a = \frac{\dot{\delta}_i}{h}$ $b = y_i - ax_i$	$y_{i,j} = ax_{i,j} + b$
$d = 2$	$\dot{\delta}_i = y_{i+1} - y_i$ $\dot{\delta}_{i+1} = y_{i+2} - y_{i+1}$ $\ddot{\delta}_i = \dot{\delta}_{i+1} - \dot{\delta}_i$ $a = \frac{\ddot{\delta}_i}{2h^2}$ $b = \frac{\dot{\delta}_i}{h} - a(x_i + x_{i+1})$ $c = y_i - bx_i - ax_i^2$	$y_{i,j} = ax_{i,j}^2 + bx_{i,j} + c$
$d = 3$	$\dot{\delta}_i = y_{i+1} - y_i$ $\dot{\delta}_{i+1} = y_{i+2} - y_{i+1}$ $\dot{\delta}_{i+2} = y_{i+3} - y_{i+2}$ $\ddot{\delta}_i = \dot{\delta}_{i+1} - \dot{\delta}_i$ $\ddot{\delta}_{i+1} = \dot{\delta}_{i+2} - \dot{\delta}_{i+1}$ $\ddot{\delta}_i = \ddot{\delta}_{i+1} - \ddot{\delta}_i$ $a = \frac{\ddot{\delta}_i}{6h^3}$ $b = \frac{\ddot{\delta}_i}{2h^2} - a(x_i + x_{i+1} + x_{i+2})$ $c = \frac{\dot{\delta}_i}{h} - b(x_i + x_{i+1}) - a(x_i^2 + x_i x_{i+1} + x_{i+1}^2)$ $d = y_i - cx_i - bx_i^2 - ax_i^3$	$y_{i,j} = ax_{i,j}^3 + bx_{i,j}^2 + cx_{i,j} + d$

Pour trouvez les valeurs de  $y_{i,j}$  vous n'avez qu'à appliquer  $x_{i,j}$  au polynôme trouvé.

## Résultat et affichage

Votre programme affiche simplement les valeurs de tous les  $y$  à l'écran, à raison d'une valeur par ligne. Donc, si vous avez  $n$  valeur en entrées, et que l'utilisateur demande un polynôme de degré  $d$ , alors votre programme affiche les valeurs pour :  $y_0$   $y_{0,1}$   $y_{0,2}$   $y_{0,3}$   $y_1$   $y_{1,1}$   $y_{1,2}$   $y_{1,3}$   $y_{n-d-1}$   $y_{n-d-1,1}$   $y_{n-d-1,2}$   $y_{n-d-1,3}$   $y_{n-d}$ .

Remarque :

- Les valeurs en bleu sont une répétition des valeurs de départ, celle dans le fichier d'entrées.
- Les valeurs en rouge sont les valeurs ajoutées par vos calculs.
- Votre logiciel va possiblement arrêter avant de faire le calcul pour les derniers intervalles. Dans le cas où  $d = 1$ , le dernier intervalle n'est pas utilisé et lorsque  $d = 2$ , les deux derniers intervalles ne sont pas calculés.

## Directives

1. Le TP est à faire seul ou en équipe de deux.
2. Code :
  - a. Pas de `goto`, `continue`.
  - b. Les `break` ne peuvent apparaître que dans les `switch`.
  - c. Un seul `return` par méthode.
3. Indentez votre code. Assurez-vous que l'indentation est faite avec des espaces.
4. Commentaires
  - Commentez l'entête de chaque classe et méthode.
  - Une ligne contient soit un commentaire, soit du code, pas les deux.
  - Utilisez des noms d'identificateur significatif.
  - Une ligne de commentaire ne devrait pas dépasser 80 caractères. Continuez sur la ligne suivante au besoin.
  - Nous utilisons Javadoc :
    - La première ligne d'un commentaire doit contenir une description courte (1 phrase) de la méthode ou la classe.
      - Courte.
      - Complète.
      - Commencez la description avec un verbe.
      - Assurez-vous de ne pas simplement répéter le nom de la méthode, donnez plus d'information.
    - Ensuite, au besoin, une description détaillée de la méthode ou classe va suivre.
      - Indépendant du code. Les commentaires d'entêtes décrivent ce que la méthode fait, ils ne décrivent pas comment c'est fait.
      - Si vous avez besoin de mentionner l'objet courant, utilisez le mot 'this'.
    - Ensuite, avant de placer les **tags**, placez une ligne vide.
    - Placez les **tag** `@param`, `@return` et `@throws` au besoin.
      - `@param` : écris les valeurs acceptées pour la méthode.

- Dans les commentaires, placer les noms de variable et autre ligne de code entre les tags `<code> ... </code>`.
- Écrivez les commentaires à la troisième personne.

## Remise

Remettre le TP par l'entremise de Moodle. Placez vos fichiers `*.java` dans un dossier compressé (**zip**) de **Windows**, vous devez remettre l'archive. Le TP est à remettre avant le 13 octobre 23:55.

## Évaluation

- Fonctionnalité (7 pts) : des tests partiels vous seront remis. Un test plus complet sera appliqué à votre TP.
- Structure (2 pts) : veillez à utiliser correctement le mécanisme d'héritage et de méthode. Vous devez avoir une classe de base pour vos fonctions d'échantillonnage. Chaque polynôme ( $d = 1, 2$  ou  $3$ ) doit avoir une classe qui hérite de la classe de base. Utilisez correctement les mécanismes de généralisation et spécialisation pour éliminer la duplication de code.
- Lisibilité (2 pts) : commentaire, indentation et noms d'identificateur significatif.