

Aide-mémoire pour la classe Deug

La classe `Deug` permet de réaliser plus facilement des entrées/sorties, des dessins, quelques manipulations élémentaires de chaînes de caractères. Cet aide-mémoire décrit brièvement certaines méthodes, regroupées par thème ; la documentation complète se trouve en <http://www.liafa.jussieu.fr/~yunes/deug/Deug/>

1 Entrées

Méthode	Effet	Exemple
<code>Deug.readBoolean()</code>	renvoie le booléen lu	<code>boolean b = Deug.readBoolean();</code>
<code>Deug.readInt()</code>	renvoie le nombre entier lu	<code>int n = Deug.readInt();</code>
<code>Deug.readDouble()</code>	renvoie le nombre à virgule lu	<code>double x = Deug.readDouble();</code>
<code>Deug.readChar()</code>	renvoie le caractère lu	<code>char c = Deug.readChar();</code>
<code>Deug.readString()</code>	renvoie le mot lu	<code>String mot = Deug.readString();</code>
<code>Deug.readLine()</code>	renvoie le ligne lu	<code>String ligne = Deug.readLine();</code>

2 Sorties

- `Deug.println(message)` affiche le message puis va à la ligne.
- `Deug.print(message)` affiche le message.

Le *message* peut être un texte, un nombre, un booléen ou un caractère. Il peut être donné directement (ex. : `Deug.print("Bonjour")`, `Deug.print(42)`), être le contenu d'une variable (`Deug.print(reponse)`), ou être le résultat d'un calcul plus complexe (`Deug.print(32*x+25)`, `Deug.println("Le produit fait " + (x * y))`).

3 Opérations sur les textes

Rappel : les caractères d'un texte sont numérotés à partir de 0.

- `Deug.length(texte)` renvoie la longueur (nombre de caractères) de *texte*.
- `Deug.equals(texte1, texte2)` renvoie vrai (`true`) si *texte1* et *texte2* sont égales.
- `Deug.charAt(texte, position)` renvoie le caractère situé à la position *position* dans *texte*.
- `Deug.substring(texte, début, fin)` renvoie la partie du *texte* commençant au caractère numéro *début* et s'arrêtant **avant** le caractère numéro *fin*. Par exemple, `Deug.substring("bonjour", 2, 5)` renvoie "njo".
- `Deug.toStringToBoolean(texte)` renvoie le booléen dont *texte* est la représentation (*texte* doit donc être "true" ou "false").
- `Deug.toStringToInt(texte)` renvoie l'entier dont *texte* est la représentation (*texte* doit donc être composé de chiffres décimaux, éventuellement précédés d'un signe moins).
- `Deug.toStringToDouble(texte)` renvoie le nombre à virgule dont *texte* est la représentation.

4 Opérations mathématiques

Note : ces méthodes font partie de la classe standard `Math` et non de la classe `Deug`.

<code>Math.sqrt(nombre)</code>	calcule la racine carrée de <i>nombre</i>
<code>Math.cos(nombre)</code>	calcule le cosinus de <i>nombre</i>
<code>Math.sin(nombre)</code>	calcule le sinus de <i>nombre</i>

5 Dessins

<code>Deug.clearArea()</code>	efface la fenêtre de dessin
<code>Deug.drawLine(x0, y0, x1, y1)</code>	dessine une ligne du point $(x0, y0)$ au point $(x1, y1)$
<code>Deug.drawPoint(x, y)</code>	dessine le point de coordonnées (x, y)
<code>Deug.startDrawings()</code>	crée une fenêtre de dessin de taille 200×200
<code>Deug.startDrawings(largeur, hauteur)</code>	crée une fenêtre de dessin de taille <i>largeur</i> \times <i>hauteur</i>
<code>Deug.stopDrawings()</code>	ferme la fenêtre de dessin