

La boucle est donc l'une des multiples notations qui permettent d'exprimer un objet infini avec une expression finie. Et le fait qu'une boucle puisse ne pas terminer est une conséquence du fait que c'est un objet infini.

En Caml, cette instruction s'écrit `while b do p`. En C, elle s'écrit comme en Java.

1.2 Les instructions d'entrée et de sortie

Les constructions d'entrée d'un langage permettent de lire des données au clavier, et plus généralement d'un périphérique : souris, disque, carte réseau, ... Les constructions de sortie permettent d'afficher des données à l'écran ou plus généralement sur un périphérique : imprimante, disque, carte réseau, ...

1.2.1 Les constructions d'entrée

Les constructions d'entrée en Java sont assez complexes, nous utiliserons donc une extension de Java spécialement conçue pour ce livre : la classe `Pp1`¹.

L'évaluation de l'expression `Pp1.readInt()` attend que l'utilisateur tape un nombre entier au clavier et produit ce nombre comme valeur. Une utilisation typique est `n = Pp1.readInt();`. La classe `Pp1` contient également une construction `Pp1.readDouble` qui permet de lire un nombre à virgule au clavier et une construction `Pp1.readChar` qui permet de lire un caractère.

1.2.2 Les constructions de sortie

L'exécution de l'instruction `System.out.print(t)` ; affiche à l'écran la valeur de l'expression `t`. L'exécution de l'instruction `System.out.println()` ; passe à la ligne. L'exécution de l'instruction `System.out.println(t)` ; affiche à l'écran la valeur de l'expression `t`, puis passe à la ligne.

Exercice 1.2

Écrire en Java un programme qui lit un entier `n` au clavier, calcule l'entier 2^n et l'affiche à l'écran.

¹ Le fichier `Pp1.java` est disponible sur la page web de l'auteur. Il suffit de le placer dans le répertoire courant pour pouvoir utiliser les instructions décrites ici.