

Une instruction est une opération de base du processeur ou d'un langage de programmation, une opération que le programmeur demande à la machine d'exécuter. Une instruction est une opération élémentaire, c'est à dire l'ordre le plus basique que peut comprendre un ordinateur. Fondamentalement, elle comporte deux éléments : l'action que le processeur va effectuer et les éléments sur laquelle l'action va être effectuée. Par exemple une instruction demandant l'addition des nombres *a* et *b* fera intervenir l'opérateur d'addition et les valeurs identifiées par les noms *a* et *b*.

Les instructions typiques

Les sections suivantes présentent les instructions typiques de la plupart des langages de programmation.

Déclaration

La déclaration permet à un compilateur ou interpréteur de connaître à quoi correspond un identifiant. En général on déclare :

- les variables : en général un type précis est associé à l'identifiant. Exemple :

```
A:ENTIER
```

EN BASIC : Dim A as String * 5 [*qui donnera affecte une zone de mémoire à la variable A pour une chaîne de caractère comportant 5 caractères*]

- les fonctions : l'identifiant est associé à une fonction retournant un résultat d'un type spécifié, et prenant comme arguments ceux qui sont spécifiés. Exemple :

```
FONCTION MAX(A:ENTIER, B:ENTIER) RETOURNE ENTIER
```

- les classes : en programmation objet, une classe définit un nouveau type de données auquel peut s'appliquer un certain nombre de fonctions (appelées méthodes). Exemple :

```
CLASSE VEHICULE
```

- d'autres entités qui dépendent du langage de programmation utilisé.

Certains langage n'ont pas besoin de déclarer les variables au préalable. D'autres ne définissent pas le type des variables. Dans ce cas là, le type est celui de la valeur à laquelle on l'affecte.

Affectation

L'instruction d'affectation permet de stocker le résultat d'une expression ou la valeur d'une constante dans une variable.

Exemples :

```
A ← 5  
A ← A + 2
```

Expressions

Arithmétique

Comparaison

Appel de fonction

Bloc d'instruction

Plusieurs instructions peuvent être regroupées en une seule en les encadrant par des accolades, ou par deux mots-clés du langage de programmation utilisé.

Exemple :

```
DEBUT
  TMP ← B
  B ← A
  A ← TMP
FIN
```

Le bloc tout entier est alors considéré comme une seule instruction et peut être utilisé pour définir une fonction, ou l'utiliser avec une instruction de contrôle d'exécution.

Contrôle d'exécution

Les instructions contrôlant l'ordre d'exécution sont les suivantes :

- La condition : l'instruction qui suit n'est exécutée que si une condition est remplie. Exemple :

```
SI A < 0 ALORS A ← 0
```

On peut ajouter une instruction à exécuter dans le cas contraire :

```
SINON A ← 1
```

- Une boucle permet de répéter un certain nombre de fois une même instruction. Exemple :

```
POUR I DE 1 À 10 FAIRE
  A ← A + I
FIN_POUR
```

Ordre d'exécution des instructions

Sauf instruction contraire, l'ordre d'exécution des instructions est celui de la lecture du code source : de la gauche vers la droite, de haut en bas.

Exemple :

```
A ← 5 ;      B ← A + 2 ;  
A ← 6 ;      B ← A + 4 ;
```

L'ordre d'exécution des instructions de cet exemple est le suivant :

1. $A \leftarrow 5$; La variable A contient la valeur 5.
2. $B \leftarrow A + 2$; La variable B contient la valeur 7 (5+2).
3. $A \leftarrow 6$; La variable A contient maintenant la valeur 6.
4. $B \leftarrow A + 4$; La variable B contient maintenant la valeur 10 (6+4).

Les instructions qui utilisent un ordre d'exécution différent sont des instructions de contrôle d'exécution.

Récupérée de « <https://fr.wikibooks.org/w/index.php?title=Programmation/Instructions&oldid=564728> »

La dernière modification de cette page a été faite le 23 juillet 2017 à 13:25.

Les textes sont disponibles sous licence Creative Commons attribution partage à l'identique ; d'autres termes peuvent s'appliquer.

Voyez les termes d'utilisation pour plus de détails.