



Les instructions simples en algorithmique

Objectifs

- Utiliser les instructions simples en algorithmique
- Utiliser les opérations primitives en algorithmique

I- Les Instructions Simples

Une instruction est un ordre qui permet de spécifier à la machine l'action à effectuer. Parmi les instructions simples on a :

- L'instruction d'affectation
- L'instruction de lecture
- L'instruction d'écriture

1) Instruction d'affectation

L'instruction d'affectation consiste à attribuer une valeur à une variable.

La syntaxe générale est la suivante : **Nom Variable ← Expression** : qui se lit « Variable prend la valeur de expression ». Après l'affectation, le contenu de « variable » est modifié. Il contient désormais, la valeur de l'expression de droite. Son mode de fonctionnement est le suivant : On évalue d'abord l'expression de droite avant de l'affecter à « variable ».

« Expression » peut être :

- une constante. **Ex** : surface ← 40
- une autre variable. **Ex** : Donnée ← Valeur Méorisée
- le résultat d'une fonction. **Ex** : résultat ← racine (nombre)
- un calcul portant sur ces différents éléments. **Ex** : surface ← (PI * Carre (Diamètre)) / 4

2) Instruction de lecture ou d'affichage

Cette instruction permet d'entrer une donnée à partir du clavier. La machine attend que l'utilisateur tape une valeur au clavier.

- **Mot clé** : saisir
- **Syntaxe** : saisir (nom variable) (nom variable étant une variable d'un type déclaré à l'avance).

3) Instruction d'écriture

Cette instruction permet d'effectuer l'affichage à l'écran.

- **Mot clé** : afficher

[Accueil](#)
[Plan du site](#)
[Mes portails virtuels](#)
[Mon Blog perso](#)
[Mon facebook](#)
[Mes ressources pédagogiques](#)
[Introduction a l'algorithmie](#)
[Notion d'objets en algorithmique](#)
[Structure générale d'un algorithme](#)
[Les instructions simples en algorithmique](#)
[Introduction à EXCEL 2003](#)
[Introduction à WORD 2003](#)
[Histoire de ma soutenance](#)
[Avant ma soutenance](#)

- **Syntaxe** : afficher ('expression'). Expression sera affiché tel quel à l'écran.
- **Exemple** : afficher (Centrer votre nom) l'expression qui apparaîtra à l'écran sera : entrer votre nom

II- Opération primitives

Les instructions de lecture, d'écriture et d'affectation font partir d'un grand ensemble appelé opération primitive.

Une opération primitive est une opération non décomposable. Tout comme les types de base il n'existe pas d'opérations primitives prédéfinies. Tout dépend du type d'exercice à résoudre et du niveau d'abstraction dans lequel on se trouve. Une opération primitive se caractérise par :

- (1) Son nom ;
- (2) Le type d'objet qu'elle prend en entrée ;
- (3) Le type d'objet qu'elle retourne comme résultats ;
- (4) Les valeurs des variables que la fonction modifie ;
- (5) Une description claire de la transformation qu'elle opère sur les entrées.

RETENONS :

En algorithmique, les données sont manipulées selon un certain type. Il existe cinq principaux types : Les entiers, réels, caractères, chaînes de caractères et les booléens.

Avec ces types de bases ou primitifs, on peut construire d'autres types que l'on appelle types construits. Ayant construit un type celui-ci peut être utilisé pour la déclaration des variables au même titre que les types de bases. Chaque champ d'un type construit constitue une variable et l'accès à celle-ci se fait de la façon suivante :

nom_du_type_construit.nom_du_champ, par conséquent on peut y faire des manipulations sur elles.

Une opération primitive est une opération non décomposable. Il existe trois principales opérations primitives : les opérations de lecture, écritures, et d'affectations. En dehors de ces trois il n'existe pas d'opérations primitives prédéfinies.

Commentaires

Vous n'êtes pas autorisé à ajouter des commentaires.