

Bases en algorithmique

www.mathmaurer.com – Cours – 1^{ère} ES-L

I – Déclaration de variables

Déclarer nouvelle variable (instruction AlgoBox)

Variables

n1, taux, x22 : nombre ; 3 variables de type nombre

prenom, nom : chaîne ; 2 variables de type chaîne de caractères

tab : liste ; 1 variable de type liste (tableau à une seule ligne)

II – Le corps de l'algorithme : instructions basiques

1 – Lire une variable

Exemple: Ajouter LIRE variable (instruction AlgoBox)

...

Lire nom ; affiche une boîte de dialogue invitant l'utilisateur à taper la valeur de la variable nom.

Lire age ; affiche une boîte de dialogue invitant l'utilisateur à taper la valeur de la variable age.

...

2 – Affectation d'une valeur à une variable

Exemple: AFFECTER valeur à variable (instruction AlgoBox)

...

somme \leftarrow a + b ; L'algorithme affecte la valeur a + b à la variable somme

...

3 – Afficher à l'écran la valeur d'une variable

Exemple: Ajouter AFFICHER variable (instruction AlgoBox)

...

Afficher somme ; L'algorithme affiche à l'écran le contenu de la variable somme

...

4 – Afficher une chaîne de caractères

Exemple: Ajouter AFFICHER Message (instruction AlgoBox)

...

Afficher "La somme est : " ; L'algorithme affiche à l'écran la chaîne de caractères La somme est :

...

5 – Afficher un commentaire

Exemple: Commentaire (instruction AlgoBox)

...

// Voici un programme qui ne sert à rien !

...

III – Le corps de l'algorithme : les tests

1 – Si ... Alors ...

Si *condition* **Alors**
Debut_Si
instructions
Fin_Si

La condition est une comparaison entre 2 valeurs du même type.

Les **opérateurs de comparaison** sont = , ≠ , < , > , ≤ , ≥ .

En langage Algobox, les opérateurs de comparaison s'écrivent == , != , < , > , <= , >= .

La condition peut comporter plusieurs tests.

Les **opérateurs logiques** reliant ces tests sont **ET , OU , NON**.

- (*condition1*) ET (*condition2*) vraie, signifie que *condition1* est vraie et que *condition2* est vraie.
- (*condition1*) OU (*condition2*) vraie, signifie que *condition1* est vraie ou que *condition2* est vraie ou que les 2 conditions sont vraies.
- NON (*condition1*) vraie, signifie que *condition1* est fausse.

2 – Si ... Alors ... Sinon

Si *condition* **Alors**
Debut_Si
instructions
Fin_Si
Sinon
Debut_Sinon
instructions
Fin_Sinon

IV – Le corps de l'algorithme : les boucles

1 – Pour ... allant de ... à ... faire

Avec ce type de boucle, le nombre d'**itérations** (répétitions) est connu à l'avance. On utilise donc un **compteur** qui, une fois atteint une certaine valeur, arrête l'exécution de la boucle. A chaque passage dans la boucle, le compteur sera **incrémenté** (+1) ou **décrémenté** (-1) selon le cas.

Dans ce cas, l'expression utilisée est :

Pour var **allant de** *valeur1* **à** *valeur2* **faire**
Debut_Pour
instructions
Fin_Pour

2 – Tant que ... faire

Avec ce type de boucle, on répète l'exécution d'un bloc d'instruction tant qu'une condition préalablement définie est satisfaite. Le test s'effectue en début de boucle, à chaque passage.

Dans ce cas, l'expression utilisée est :

Tant que *condition* **faire**
Debut_Tant_que
instructions
Fin_Tant_que