

Chapitre 5 : Démarche de résolution de problèmes

I. Introduction :

- Le traitement de l'information se fait au moyen des programmes. Pour se faire, Diverses applications ont été développées par des particuliers ou des sociétés pour résoudre différents problèmes (MS Word pour le traitement de texte, Audacity pour le traitement de son...).
- Que fait-on pour réaliser nous-même des petites applications pour résoudre des petits problèmes ?
- Comment rédiger une solution au problème posé ?
- Comment traduire cette solution à l'ordinateur ? Quel est le langage à utiliser ?

II. Les étapes de résolution d'un problème :

Activité 1 :

Quelle est la démarche à suivre pour écrire un programme qui permet de calculer et d'afficher la somme de deux entiers a et b donnés.

1. Compréhension de l'énoncé : Pré-analyse

Cette étape consiste à bien lire et comprendre l'énoncé, c.à.d. présenter le problème d'une manière simplifiée.

2. Spécification et Analyse du problème :

- Il s'agit de déterminer :
 - Le résultat souhaité
 - Le traitement à effectuer
 - Les données à utiliser
- Pour définir les actions, on utilise un vocabulaire d'écriture précis :
 - Pour traduire une action de lecture ou d'entrée des données : **nom_ objet = donnée.**
Exemple : a=donnée
 - On utilise le signe **←** pour traduire l'affectation d'une valeur à une variable.
Exemple : s←a+b
 - Pour afficher le résultat sur l'écran : **Ecrire (nom_ objet).**
Exemple : Ecrire (S)
- **Les objets** : sont des cases mémoires au niveau de la RAM pouvant contenir les valeurs manipulés.
Exemples : a, b, s

- Il existe deux méthodes d'analyse :

| Activité 1 (Analyse Descendante) | Activité 1 (Analyse Ascendante) |
|---|---|
| <p>Résultat : Ecrire (s)</p> <p>Traitement :</p> <p style="text-align: center;">$S \leftarrow a + b$</p> <p>$a = \text{donnée}$ $b = \text{donnée}$</p> <p>Données : a, b</p> | <p>Données : a, b</p> <p>Traitement :</p> <p>$a = \text{donnée}$ $b = \text{donnée}$</p> <p style="text-align: center;">$S \leftarrow a + b$</p> <p>Résultat : Ecrire (s)</p> |

→ On va adopter **l'approche descendante** pour la spécification de nos problèmes. Dans Cette méthode on part du résultat du problème posé et à chaque fois qu'un nouvel objet apparaît il est défini jusqu'à atteindre les données.

TDOU (Tableau de Déclaration des Objets Utilisés)

Ce tableau regroupe tous les objets utilisés en indiquant leurs noms, leurs types ainsi que leurs rôles.

Exemple : TDOU (activité 1)

| Nom | Type/Nature | Rôle |
|------------|--------------------|---|
| a | entier | Contenir la valeur du 1 ^{er} entier |
| b | entier | Contenir la valeur du 2 ^{ème} entier |
| S | entier | Contenir la somme |

3. Ecriture de l'algorithme :

La numérotation dans l'ordre chronologique de toutes les actions évoquées dans l'analyse, donne **une suite finie et ordonnée d'actions (instructions) servant à résoudre un problème, C'est l'algorithme.**

| Structure d'un algorithme : | Exemple (activité 1) : |
|--|--|
| <p>0) Début Nom-Algorithmme</p> <p>1) Instruction1</p> <p>2) Instruction2</p> <p>...</p> <p>n) Fin Nom-Algorithmme</p> | <p>0) Début Somme</p> <p>1) Lire (a)</p> <p>2) Lire (b)</p> <p>3) Somme $\leftarrow a + b$</p> <p>4) Ecrire (S)</p> <p>5) Fin Somme</p> |

4. Ecriture du programme :

- Traduction de l'algorithme en un **programme source** développé **sous forme d'un fichier** moyennant un **langage de programmation.**

- Il existe plusieurs langages de programmation tels que C, Java, Pascal ...,
- on va utiliser le langage PASCAL.

| Structure d'un programme écrit en pascal : | Exemple (Activité) |
|---|--|
| Program nom_du_programme ; Uses wincrt ; {Déclaration des éventuels objets }; Begin Instruction 1; Instruction 2; ... Instruction n; End. | Program Somme; Uses wincrt; Var a,b :Integer ; Begin Readln (a); Readln (b); S := a+b ; Writeln (S) ; End. |

5. La compilation :

Le programme source ainsi obtenu doit être compilé (**Compiler**). Le langage traduit ainsi l'ensemble du programme source en langage machine, pour donner un fichier exécutable.

6. Exécution et tests :

Cette étape consiste à lancer l'exécution (**exécuter**) du programme compilé et faire des tests bien choisis afin de pouvoir corriger les éventuelles erreurs et s'assurer de l'efficacité du programme.

III. Application

Ecrire un programme qui permet de calculer et d'afficher la moyenne en informatique d'un élève.

Spécification et Analyse :

Résultat : Ecrire ("moyenne = ", Moy)

Traitement :

$$\text{Moy} \leftarrow (\text{DC} + 2 * \text{DS}) / 3$$

DS = donnée (" Donner DS :")

DC = donnée (" Donner DC :")

Données : DC, DS

TDOU

| Nom | Type/Nature | Rôle |
|------------|--------------------|--------------------------|
| DC | réel | Contenir la valeur du DC |
| DS | réel | Contenir la valeur du DS |

| | | |
|-----|------|----------------------|
| Moy | réel | Contenir la moyenne. |
|-----|------|----------------------|

Algorithme :

- 0) Début Moyenne
- 1) Ecrire ("Donner DC :") ; Lire (DC)
- 2) Ecrire ("Donner DS :") ; Lire (DS)
- 3) Moy \leftarrow (DC + 2 * DS) / 3
- 4) Ecrire ("Moyenne = ", moy)
- 5) Fin Moyenne

Traduction en pascal