

Atelier – Trace d'exécution

Objectifs pédagogiques

1. Comprendre le rôle de la trace d'exécution comme outil de vérification.
2. Construire un tableau de trace à partir d'un algorithme.
3. Développer de la rigueur dans l'exécution pas-à-pas d'un algorithme.
4. Identifier et corriger des erreurs grâce à la trace.

Déroulement

1. Introduction & rappel

5. Présentation d'un algorithme simple et sa trace.
6. Discussion :
 - Pourquoi une trace d'exécution aide-t-elle à comprendre un algorithme ?
 - Quelles sont les étapes clés pour la construire ?
 - Présentation de la **liste de vérification** (numéroter les lignes, ajouter une colonne console, etc.).

2. Exercice guidé en groupe

Algorithme proposé : calcul de la moyenne de deux notes

```
ALGORITHME MoyenneDeuxNotes
VARIABLES
    note1, note2, moyenne TYPE NUMÉRIQUE
DÉBUT
    1 AFFICHER("Entrez la première note")
    2 LIRE(note1)
    3 AFFICHER("Entrez la deuxième note")
    4 LIRE(note2)
    5 moyenne ← (note1 + note2) / 2
    6 AFFICHER("La moyenne est ", moyenne)
FIN
```

Atelier – Trace d'exécution

Tâches

7. Construire collectivement la table de trace.
8. Supposer que l'utilisateur entre 70 puis 90.
9. Vérifier étape par étape où apparaissent les changements.

Résultat attendu

10. note1 = 70, note2 = 90, moyenne = 80
11. Console affiche : La moyenne est 80

3. Exercice en binôme

Algorithme proposé : trouver le maximum de deux nombres

```
ALGORITHME MaxDeuxNombres
VARIABLES
a, b, max TYPE NUMÉRIQUE

DÉBUT
  1 AFFICHER("Entrez le premier nombre")
  2 LIRE(a)
  3 AFFICHER("Entrez le deuxième nombre")
  4 LIRE(b)
  5 SI a > b ALORS
  6     max ← a
  7 SINON
  8     max ← b
  9 FINSI
  10 AFFICHER("Le maximum est ", max)
FIN
```

Consignes

1. Complétez le tableau de trace.
2. Faites-le avec deux jeux de valeurs :
 - Cas 1 : a = 12, b = 20
 - Cas 2 : a = 25, b = 10
3. Comparez vos résultats avec un autre binôme pour vérifier la cohérence.

Atelier – Trace d'exécution

4. Exercice autonome (si le temps le permet)

Algorithme proposé : calcul de la factorielle d'un entier n (version avec boucle)

Explication: qu'est-ce qu'une boucle en programmation et comment fonctionne la boucle for ?

```
ALGORITHME Factorielle
VARIABLES
n, i, fact TYPE NUMÉRIQUE

DÉBUT
  1 AFFICHER("Entrez un nombre entier positif")
  2 LIRE(n)
  3 fact ← 1
  4 POUR i ← 1 À n FAIRE
  5     fact ← fact * i
  6 FINPOUR
  7 AFFICHER("La factorielle de ", n, " est ", fact)
FIN
```

Tâches individuelles

1. Complétez le tableau de trace pour le cas **n = 4**.
2. Vérifiez que le résultat final affiché correspond bien à 24.
3. Identifiez les lignes où les variables changent à chaque itération.

5. Mise en commun & discussion

- 1-2 étudiants présentent leur trace au tableau.
- Discussion :
 - Quelles erreurs sont revenues fréquemment ?
 - Qu'est-ce qui vous a aidés à rester rigoureux ?

6. Rappel

- La trace est un **outil de vérification universel**, pas seulement pour l'examen ou pour java, mais aussi pour **déboguer n'importe quel programme** écrit avec n'importe quel langage.