ALGORITHMIQUE · CHAPITRE 10 · FONDATIONS

Lire et écrire du pseudo-code

Lisez et écrivez du pseudo-code en anglais : affectation, commentaires, entrées et sorties. Comparez un calcul de commande en Python et MATLAB.

Par Ali ZAINOUL · Cours gratuit

Ce que vous allez apprendre

- Lire une séquence d'instructions dans l'ordre.
- Appliquer des conventions explicites d'affectation et de commentaire.
- Vérifier une traduction par une table de trace.

Prérequis. Connaître les variables, expressions et comparaisons.

Pourquoi écrire une version intermédiaire ?

Deux personnes peuvent connaître des langages différents et vouloir discuter de la même méthode. Un **pseudo-code** décrit les étapes avec une notation choisie pour être comprise. Il n'existe pas une unique syntaxe universelle : il faut annoncer ses conventions et les conserver.

Les conventions du parcours

- `<-` signifie affecter ; `==` compare deux valeurs dans le pseudo-code.
- Les commentaires commencent par `//` et ne sont pas exécutés.
- Les noms décrivent les données : `price`, `quantity`, `total`.
- L'indentation rend les blocs visibles. Les mots-clés de fermeture seront définis avec les conditions et les boucles ; aucune de ces constructions n'est nécessaire pour la séquence présentée ici.

- Les bornes des répétitions seront indiquées comme incluses ou exclues ; les suites seront indexées à partir de 0 dans le pseudo-code.

Les premières lignes d'un exemple peuvent nommer l'algorithme, sans lui inventer un fichier exécutable. Les fichiers Python et MATLAB seront indiqués dans un commentaire valide de leur langage.

Une séquence sans branchement

Nous calculons le total d'une commande simple. `price` est un prix unitaire en centimes, `quantity` une quantité entière et `delivery` des frais fixes en centimes. Employer des centimes entiers évite de mélanger euros et centimes dans les étapes.

Pseudo-code

```
// example.pseudo
// compute_order
price <- 250
quantity <- 3
delivery <- 100
subtotal <- price * quantity
total <- subtotal + delivery
PRINT total
```

Python

```python
# compute_order.py
price = 250
quantity = 3
delivery = 100
subtotal = price * quantity
total = subtotal + delivery
print(total)  # 850 cents
```

MATLAB

```
% compute_order.m
price = 250;
quantity = 3;
delivery = 100;
subtotal = price * quantity;
total = subtotal + delivery;
disp(total); % 850 cents
```

Relier chaque ligne à un état

Trace d'une commande de trois unités

Après cette étape	subtotal	total
Initialisation des entrées	Non défini	Non défini
subtotal ← price × quantity	750	Non défini
total ← subtotal + delivery	750	850

La trace indique le moment où une valeur devient disponible. On ne peut pas calculer `total` à partir de `subtotal` avant d'avoir défini ce dernier. Un texte qui contient toutes les bonnes opérations dans le mauvais ordre peut donc être inexécutable.

Modifier une donnée et prévoir

Remplacez la quantité 3 par 1 : le sous-total devient 250 et le total 350 centimes. Expliquez pourquoi les frais de livraison restent ajoutés une seule fois. Une méthode qui multiplie aussi ces frais par la quantité décrirait une convention commerciale différente.

Compléter sans inventer une règle

▼ **Comment ajouter une réduction fixe de 50 centimes ?**

Il faut préciser à quoi elle s'applique. Si elle diminue le total de la commande, on peut calculer total ← subtotal + delivery − discount avec discount = 50, ce qui donne ici 800 centimes. Il faut aussi décider si un total négatif est autorisé pour d'autres entrées.

▼ **"Afficher le bon total" est-il une instruction suffisante pour le calculer ?**

Non. Elle décrit l'affichage d'une valeur déjà calculée. Il manque les étapes qui construisent ce total à partir des données et des conventions.

Lire, tracer, traduire

Le pseudo-code exprime la méthode ; les versions Python et MATLAB en sont deux traductions. Vérifier que les mêmes données conduisent au même résultat permet de repérer une différence de traduction. Les conditions ajoutent maintenant la possibilité de ne pas suivre toujours le même chemin.

À retenir - fiche mémo

Convention

Le pseudo-code décrit une méthode ; il ne possède pas une syntaxe universelle unique.

Séquence

Une instruction utilise les valeurs produites par les instructions précédentes.

Unités

Dans la commande étudiée, prix, sous-total et frais sont en centimes.

Pseudo-code

```
// example.pseudo
// compute_order
subtotal <- price * quantity
total <- subtotal + delivery
PRINT total
```

Point de vigilance. Le pseudo-code n’est pas un fichier directement exécutable.

Consulter la cheatsheet : Fondations · **Télécharger la fiche du thème en PDF**

Version web : https://www.clearcode.fr/courses/programming/algorithmic/foundations/pseudocode