ALGORITHMIQUE · PARCOURS · REPRÉSENTATIONS

Représenter et expliquer un algorithme

Choisissez une représentation graphique adaptée : organigramme ou activité UML. Apprenez leurs symboles et reliez chaque chemin aux instructions.

Par Ali ZAINOUL · Cours gratuit

Ce que vous allez apprendre

- Distinguer l'organigramme et le diagramme d'activité UML.
- Expliquer chaque symbole avant de suivre un chemin.
- Relier les entrées, actions et sorties d'un schéma au code correspondant.

Prérequis. Avoir découvert les conditions, les boucles et les fonctions.

Choisir une notation graphique et annoncer sa légende

Représenter un algorithme en schéma permet de suivre un ordre d'exécution sans lire immédiatement toute la syntaxe d'un langage. Le dessin doit cependant annoncer ses conventions. Un organigramme et un diagramme d'activité UML peuvent décrire une même décision avec des formes différentes ; il ne faut pas mélanger leurs symboles sans l'expliquer.

Le premier chapitre définit les formes de notre organigramme et les relie au code d'un test de signe. Le second définit un diagramme d'activité UML limité à un choix séquentiel : nœud initial, actions, décision, gardes, fusion et fin d'activité. L'entrée du problème et sa sortie sont précisées séparément des flèches qui guident l'exécution.

Les chapitres de cette partie

- 26. Dessiner un organigramme lisible

- 27. Découvrir UML avec un diagramme d'activité

Lire les données, les symboles puis le parcours

Commencez par nommer l'entrée et le résultat attendu. Repérez ensuite la forme de départ et lisez chaque flèche dans son sens. Pour une décision, calculez la réponse avant de choisir une branche. Pour une boucle, dites quelle valeur change avant de revenir au test. Le même essai doit produire une histoire cohérente dans le texte, le diagramme et le code.

Les animations disposent de boutons de lecture, de pause et de progression manuelle. Elles conservent une transcription de toutes les étapes. Les corrections demandent d'expliquer les symboles, notamment pourquoi le rond de départ ne représente pas une donnée d'entrée et pourquoi le cercle final ne représente pas une valeur de sortie.

À retenir - fiche mémo

Légende

Les formes ne doivent pas être devinées : annoncez la notation choisie.

Données

L'entrée et le résultat du problème se distinguent des flèches de contrôle.

Correspondance

Le texte, le schéma et le code doivent décrire le même chemin pour une même entrée.

Pseudo-code

```
// example.pseudo
// diagram_check
// Define the input.
// Evaluate the test.
// Follow one allowed branch.
// Compare the observed output.
```

Point de vigilance. Une animation d'un cas ne prouve pas tous les comportements possibles.

Consulter la cheatsheet : Organigrammes et UML · **Télécharger la fiche du thème en PDF**

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