

ISN S2 : Les algorithmes

Un algorithme est une série d'actes ou d'opérations élémentaires qu'il faut exécuter en séquence pour accomplir une tâche quelconque, en suivant un enchaînement strict.

Lorsqu'il sera demandé d'élaborer un algorithme, la méthode pour atteindre cet objectif sera de rédiger en français la succession des opérations élémentaires (phases courtes et précises) puis de passer à une écriture conventionnelle appelée pseudo-code ou algorigramme.

1. Organisation d'un algorithme :

- ✚ L'en-tête : Nom de l'algorithme. description du traitement effectué, données en entrée (paramètres, arguments), données en sortie (résultats).
- ✚ La partie déclarative : Liste des "objets" utilisés par l'algorithme (constantes, variables).
- ✚ Le corps : Une suite d'instructions délimitée par les mots "début" en "fin".

```

fonction Surface
Calculer la surface d'un disque
à partir de son rayon

Paramètre :
    rayon : réel
Résultat :
    surface : réel

Pi = 3,14
surface : réel

début
    surface <- Pi * rayon * rayon
    retourner surface
fin
    
```

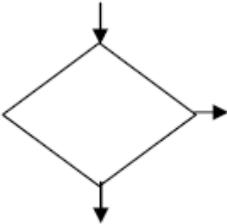
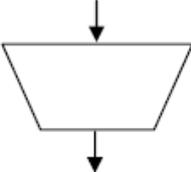
Le pseudo-code consiste à exprimer en langage naturel, mais selon une disposition particulière et des mots choisis, les différentes opérations constituant l'algorithme, conformément au code donné dans le tableau qui suit.

Remarque :

Le pseudo-code se prête mieux que l'algorigramme à la description des structures complexes existant dans les langages de haut niveau.

Mots et symboles du pseudo-code	Opérations réalisées
Début	Début de l'algorithme, permet de le nommer
Fin	Fin de l'algorithme
Faire	Exécution d'une opération
Entrer	Acquisition ou chargement d'une donnée
Sortir	Edition ou sauvegarde d'un résultat
Retourner	Retourner le résultat d'une fonction
←	Affectation d'une valeur à une variable
Symboles d'opérateur (+-*/ ET OU,...)	Opérations arithmétiques ou logiques
Aller à	Branchement inconditionnel (déconseillé)
Si...alors...[sinon]	Branchement inconditionnel
Selon cas...[autrement]	Branchement conditionnel généralisé
Tant que...faire...	} Répétition conditionnelle
Répéter...jusqu'à...	
Pour...=...à...	
	Répétition contrôlée

L'algorigramme est la représentation graphique de l'algorithme, il permet de représenter chaque opération élémentaire au moyen d'un symbole graphique normalisé.

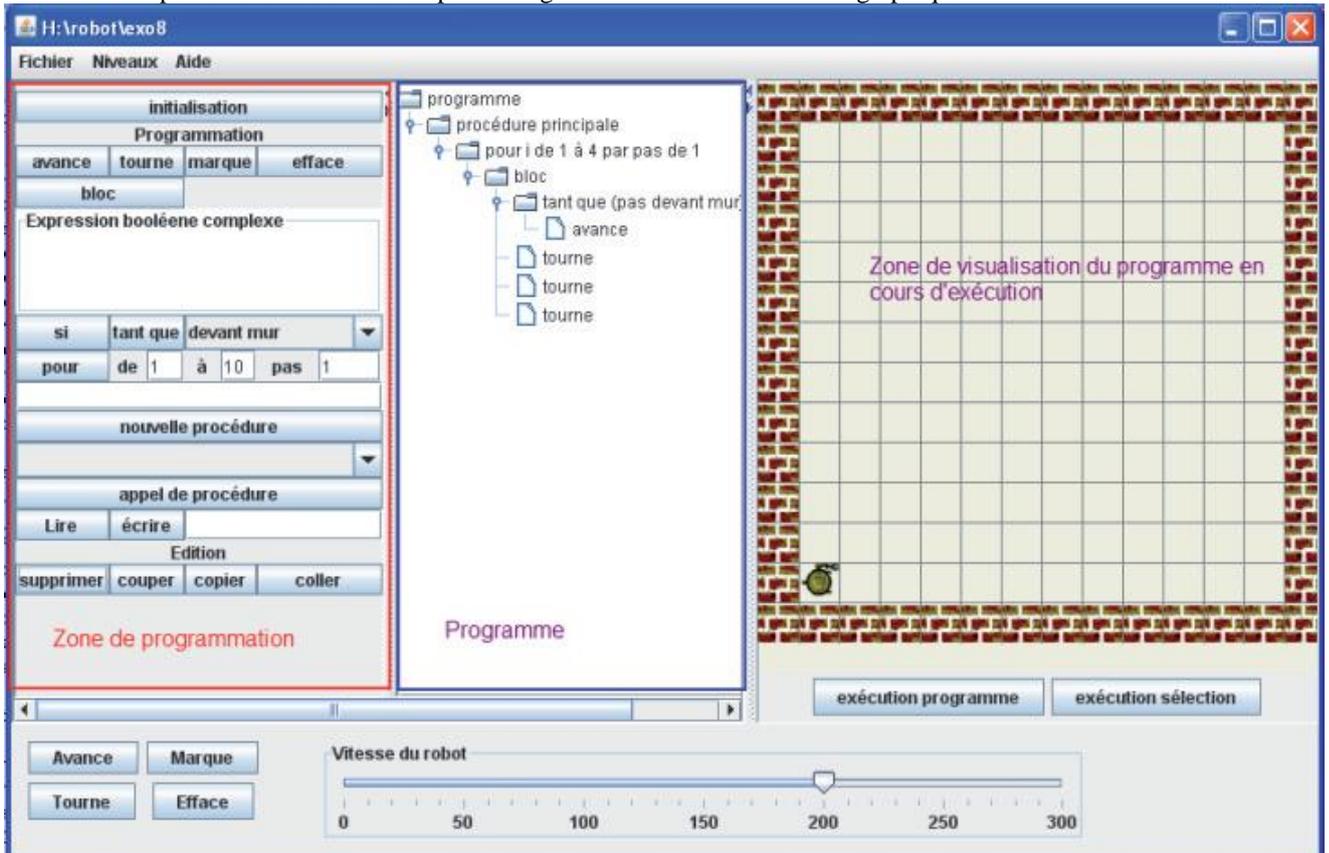
 <p>début / fin</p>	 <p>Traitement</p>	 <p>Sous programme</p>	 <p>Entrée / sortie</p>
 <p>Test / condition</p>	 <p>Opération manuelle</p>	 <p>Symbole de renvoi utilisé 2 fois pour assurer la continuité lorsqu'une partie de la ligne de liaison n'est pas représentée.</p>	 <p>Commentaires : symbole utilisé pour donner des indications marginales</p>

Règles de construction :

- ✚ Centrer l'algorithme sur une feuille
- ✚ Construire l'algorithme afin que sa lecture s'effectue verticalement
- ✚ Les lignes de liaison entre symboles ne doivent pas en principe se couper (utiliser un symbole de renvoi)
- ✚ Une ligne de liaison doit toujours arriver sur le haut et au centre d'un symbole.
- ✚ Les commentaires sont à placer de préférence à droite, et les renvois de branchement à gauche.

Le monde de Nono est un logiciel développé par l'**UHA de Mulhouse** pour s'initier à la programmation algorithmique. Il s'agit de faire exécuter des missions à un petit robot «Nono» qui évolue dans un monde à la surface limitée.

Le déplacement est commandé par des algorithmes décrits de manière graphique.



Selon le niveau d'exécution (Menu **Niveaux**) du programme, les possibilités de programmation augmentent.

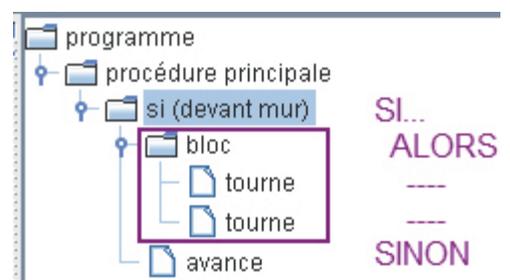
Les instructions de déplacement sont :

- **avance** : fait avancer Nono d'une case
- **tourne** : provoque la rotation horaire de 90°
- **marque** : place un plot sur la case courante
- **efface** : efface le plot de la case courante

Les instructions conditionnelles (*Si Alors Sinon*), et itératives (*tant que, pour*) sont gérées.

Lorsque plusieurs instructions sont exécutées les unes après les autres, un **bloc** (d'instruction) est nécessaire

L'exécution du programme est obtenue grâce au bouton **exécution programme**.



2. Algorithme linéaire:

a) Lancer le logiciel ROBOT.JAR dans un navigateur Internet. Dans le menu, placer le niveau à 6.

b) Mission 1 :

Initialiser le robot de sorte qu'il soit placé dans le coin Sud-Ouest et orienté vers le Nord. Faire avancer Nono de deux cases.

c) Mission 2 :

Initialiser le robot de sorte qu'il soit placé dans le coin Sud-Ouest et orienté vers le Nord. Faire avancer Nono sur la case située au Nord-Est de la case actuelle. En position finale il doit être orienté vers le Nord

3. Algorithme conditionnel:

d) Mission 3 :

Initialiser Nono à une position et une orientation quelconque. Faire avancer Nono de 5 cases sans le faire foncer dans un mur.

4. Algorithme conditionnel et itératif:

e) Mission 4 :

Initialiser le robot de sorte qu'il soit placé dans le coin Sud-Ouest et orienté vers le Nord. Le faire aller au Nord jusqu'au mur.

f) Mission 5 :

Initialiser le robot de sorte qu'il soit placé dans le coin Sud-Ouest et orienté vers le Nord. La surface du terrain est carrée. Tracer une ligne de marque en diagonale.

g) Mission 6 :

Initialiser Nono dans le coin Sud-Ouest dans une orientation quelconque. L'orienter vers le Nord en utilisant des instructions itératives.

h) Mission 7 :

Le robot est n'importe où sur le terrain orienté vers le Nord. Le faire se déplacer dans le coin sud-ouest.

i) Mission 8 :

Nono est placé dans le coin sud-ouest orienté vers le Nord. Trouver, de la manière la plus économique, le minerai.

Remarque : On utilisera le test *pas devant marque* et non *devant marque*. Il est possible également d'utiliser une expression booléenne complexe (Niveau 4).