

Structures conditionnelles

Forme réduite

Définition

Une **structure conditionnelle simple à forme réduite** admet un seul traitement qui sera exécuté uniquement **si une condition est vraie**.

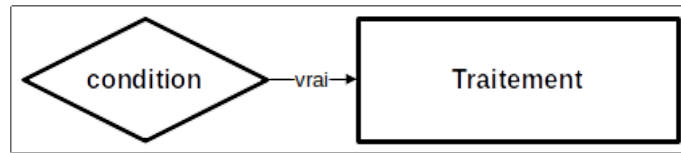


Figure 1, Forme simple réduite

Forme simple réduite

Algorithme

```

Si condition Alors
  // Traitements
Fin Si
  
```

Pascal

```

if condition then begin
  // Traitements
end;
  
```

Python

```

if condition:
  # Traitements
  
```

Exemple

Je suis dans une chambre et je veux en sortir. **Donner l'algorithme correspondant.**



Figure 2, Exemple réel de si à forme simple réduite

Si la porte est fermée
alors l'ouvrir.
 Sortir.

Dans cet exemple, si la porte est initialement ouverte **il n'est pas nécessaire de l'ouvrir.**

Activité 1

Ecrire un programme qui calcule et affiche **la valeur absolue d'un entier donné**, sans utiliser la fonction prédéfinie abs.

Indice : Utiliser une **structure simple réduite** pour inverser x uniquement s'il est négatif.

On rappelle que :

- $|x| = x$, si $x \geq 0$
- $|x| = -x$, si $x < 0$

Activité 2

Ecrire un programme qui saisit deux entiers **a** et **b** et les affiche en ordre décroissant. **a** devra toujours contenir la valeur la plus grande et **b** la plus petite.

Indice : Utiliser une **structure simple réduite** pour permuter **a** et **b** s'il ne sont pas dans l'ordre.

Exemple

Exemple d'exécution n°1

```

Donner a ? 25
Donner b ? 13
25 - 13
  
```

Exemple d'exécution n°2

```

Donner a ? 8
Donner b ? 17
17 - 8
  
```

Forme alternative

Définition

Une structure conditionnelle à forme alternative admet deux traitements différents :

- Le premier est exécuté uniquement **si la condition est vraie**,
- Le second est exécuté uniquement **si la condition est fausse**.

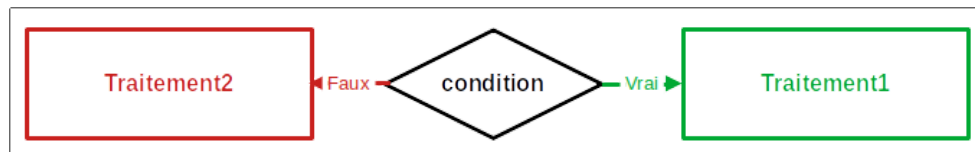


Figure 3, Forme simple alternative

Forme simple alternative

Algorithme

```
Si condition Alors
    // Traitement
    // si condition = vrai
Sinon
    // Traitement
    // si condition = faux
Fin Si
```

Pascal

```
if condition then begin
    // Traitement
    // if condition = true
end else begin
    // Traitement
    // if condition = false
end;
```

Python

```
if condition:
    # Traitement
    # if condition = True
else:
    # Traitement
    # if condition = False
```

Exemple

Je suis dans un rond point, quel chemin dois-je choisir ? **Donner l'algorithme correspondant.**

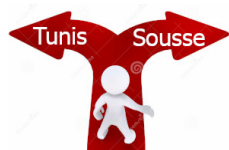


Figure 4, Quel est le bon chemin

Si je veux aller à tunis
alors suivre le chemin qui est à gauche
sinon suivre le chemin qui est à droite.

Dans cet exemple il faut **choisir entre deux chemin** selon le besoin.

Activité 3

Ecrire un programme qui saisit la **largeur** et l'**hauteur** d'un quadrilatère qui possède quatre angles droits, puis affiche s'il s'agit d'un **carré** ou d'un **rectangle**.

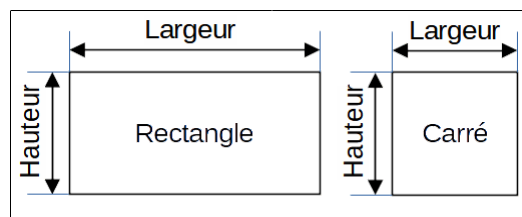


Figure 5, Rectangle ou Carré

Activité 4

Un des moyens, possibles, pour **tester si deux vecteurs du plan sont orthogonaux** est de calculer leurs **produit scalaire**. Si le **produit scalaire des deux vecteurs** est **nul** on peut conclure qu'ils sont **orthogonaux**.

La formule qui permet de calculer produit scalaire de deux vecteurs dans le plan est la suivante :

$$\vec{u} = \begin{pmatrix} x_u \\ y_u \end{pmatrix} \quad \vec{v} = \begin{pmatrix} x_v \\ y_v \end{pmatrix} \quad \vec{u} \cdot \vec{v} = x_u \cdot x_v + y_u \cdot y_v$$

Figure 6, Produit scalaire de deux vecteurs

Ecrire un programme qui demande à l'utilisateur de saisir les composantes de 2 vecteurs du plan u et v, puis détermine et affiche s'ils sont orthogonaux ou non.

Exemple

On veut déterminer dans l'exemple suivant si :

- Les vecteurs AB et HC sont orthogonaux
- Les vecteurs AB et AC sont orthogonaux

Graphiquement, **AB** et **HC** sont orthogonaux, alors que **AB** et **AC** ne sont pas orthogonaux.

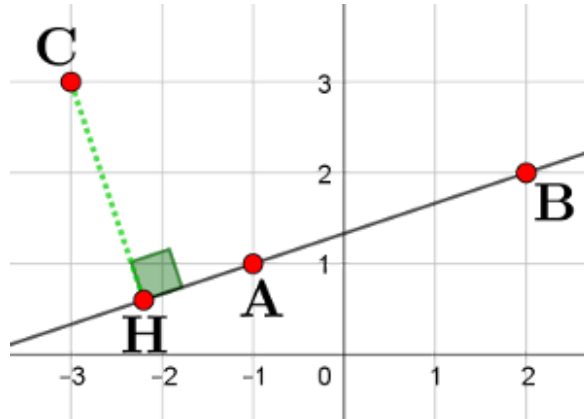


Figure 7, Les vecteurs AB et HC sont orthogonaux, Les vecteurs AB et AC ne sont pas orthogonaux.

On donne les coordonnées des points : A(-1, 1), B(2, 2), C(-3, 3), H(-2.2, 0.6)

$$\overrightarrow{AB} = \begin{pmatrix} 3 \\ 1 \end{pmatrix}, \overrightarrow{HC} = \begin{pmatrix} -0.8 \\ 2.4 \end{pmatrix}, \overrightarrow{AC} = \begin{pmatrix} -2 \\ 2 \end{pmatrix}$$

$$\overrightarrow{AB} \cdot \overrightarrow{HC} = \begin{pmatrix} 3 \\ 1 \end{pmatrix} \cdot \begin{pmatrix} -0.8 \\ 2.4 \end{pmatrix} = 3 \times (-0.8) + 1 \times 2.4 = 0$$

$$\overrightarrow{AB} \cdot \overrightarrow{AC} = \begin{pmatrix} 3 \\ 1 \end{pmatrix} \cdot \begin{pmatrix} -2 \\ 2 \end{pmatrix} = 3 \times (-2) + 1 \times 2 = -4$$

Figure 8, Calcul du produit scalaire

Le calcul arithmétique montre que les vecteurs AB et HC sont orthogonaux, et que AB et AC ne sont pas.

Forme généralisée

Définition

Une **structure conditionnelle** simple à **forme généralisée** admet plus de deux traitements différents qui seront exécutés en fonction de plusieurs conditions.

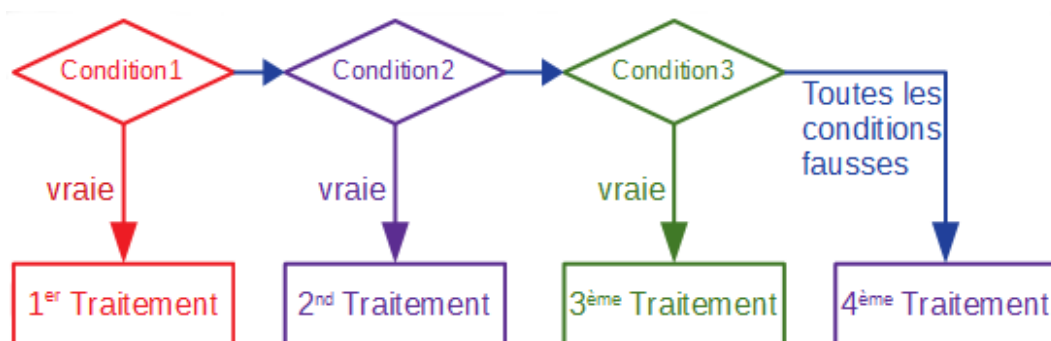


Figure 9, Forme simple alternative

Forme simple généralisée

Algorithme

```

Si condition1 Alors
    // Traitement 1
    // si condition1 = vrai
Sinon Si condition2 Alors
    // Traitement 2
    // si condition2 = vrai
Sinon Si condition3 Alors
    // Traitement 3
    // si condition3 = vrai
Sinon
    // Traitement 4
    // si les conditions sont fausses
Fin Si

```

Pascal

```

if condition1 then begin
    // Traitement 1
    // if condition1 = true
end else if condition2 then begin
    // Traitement 2
    // if condition2 = true
end else if condition3 then begin
    // Traitement 3
    // if condition = true
end else begin
    // Traitement 4
    // if conditions are False
end;

```

Python

```

if condition1:
    # Traitement 1
    # if condition1 = True
elif condition2:
    # Traitement 2
    # if condition2 = True
elif condition3:
    # Traitement 3
    # if condition3 = True
else:
    # Traitement 4
    # if conditions are False

```

Exemple

M Salim est dans un restaurant, il veut commander une Pizza. Il n'a que 8\$ et il n'aime pas le poulet. **Qu'est ce qu'il peut commander ?**



Figure 10, Sélection d'une Pizza à partir d'un menu

Si j'ai 7\$ ou plus et j'aime le poulet
alors commander Hawaiian Chicken
sinon Si j'ai 6\$ ou plus et j'aime le poulet
alors commander Chicken Delight
sinon Si j'ai 8\$ ou plus et j'aime le thon
alors commander Island Tuna
sinon Si j'ai 9\$ ou plus et j'aime le boeuf
alors commander Beef Pepperoni
sinon changer de restaurant

Activité 5

Un élève est admis s'il obtient une moyenne supérieure ou égale à 10, il est refusé s'il obtient une moyenne inférieure à 9, sinon il est contrôle.

Ecrire un programme qui affiche l'appréciation d'un élève selon sa moyenne.

Activité 6

Ecrire un programme qui détermine **la nature d'un triangle** à partir de la longueur de ses cotés.

- Equilatéral, si **$AB=BC=AC$**
- Isocèle, si **$AB=AC$ ou $AB=BC$ ou $AC=BC$**
- Rectangle, si **$AB^2+AC^2=BC^2$ ou $AC^2+BC^2=AB^2$ ou $BC^2+AB^2=AC^2$**
- Quelconque, sinon

Structure à choix

Définition

Une **structure conditionnelle à choix** est une version simplifiée de la **structure conditionnelle généralisée**.

L'exécution d'un traitement dépend d'un **sélecteur** de type ordinal.

Forme simple généralisée

Algorithme

```
Selon var Faire
    valeur1, valeur2:
        // Traitement 1
        // si condition1 = vrai
    valeur3..valeur4:
        // Traitement 2
        // si condition2 = vrai
Sinon
    // Traitement 3
    // si les conditions sont fausses
Fin Si
```

Pascal

```
Case var of
    valeur1, valeur2:
        // Traitement 1
        // si condition1 = vrai
    valeur3..valeur4:
        // Traitement 2
        // si condition2 = vrai
else
    // Traitement 3
    // si les conditions sont fausses
end;
```

Python

```
# Pas d'équivalence
if var in [valeur1, valeur2]:
    # Traitement 1
    # if condition1 = True
elif var in range(valeur3..valeur4):
    # Traitement 2
    # if condition2 = True
else:
    # Traitement 3
    # if condition3 = True
```

Activité 7

Ecrire un programme qui saisit un caractère puis détermine sa nature : Chiffre, Lettre miniscule, Lettre majuscule ou Symbole.

Utiliser une structure à choix.