Initiation à l'algorithmique

ID3 2020-2021

Contenu :

- Initiation à la logique booléenne
- Introduction à la représentation par organigramme
- Les structures algorithmiques :
  - Séquence
  - Alternative
  - Itération
- Exercices

CFP Arts Genève
http://edu.ge.ch/cfpaa/

Johann Sievering

v04

cc BY NC ND

OBJECTIFS du COURS

Introduction aux structures algorithmiques
Introduction aux organigrammes
Résoudre des problèmes au moyen d'algorithmes

Objectifs :

- Être capable de faire seul des exercices avec les structures algorithmiques
  - Séquence
  - Alternative
  - Itération
- Etre capable d'élaborer des organigrammes à partir d'un problème donné
- Etre capable de résoudre des petits problèmes au moyen d'algorithmes

(cc) BY-NC-ND

johann.sievering@edu.ge.ch

INTRODUCTION

(cc) BY-NC-ND

johann.sievering@edu.ge.ch

Introduction

Programme = Algorithme + Données

(Niklaus Wirth 1985)

Algorithme

«Un **algorithme** est une suite **finie** et **non ambiguë** d'opérations ou d'instructions permettant de **résoudre** un problème, ou **d'obtenir** un résultat, en un **temps fini** »

« Un algorithme est une méthode générale pour résoudre un type de problèmes »

Le mot algorithme vient du nom arabe « الخوارزمي » du mathématicien perse du IX e siècle Al-Khwârizmî. Le domaine qui étudie les algorithmes est appelé l'algorithmique.

Données

«Une **donnée** est une **description** élémentaire d'une **réalité**. C'est par exemple une observation ou une mesure»

(cc) BY-NC-ND

Source : Wikipédia

johann.sievering@edu.ge.ch

Exemples d'algorithmes

- **Une recette de cuisine**
  - Entrée : ingrédients
  - Résultat : plat préparé
- **La recherche dans un dictionnaire**
  - Entrée : mot
  - Résultat : définition
- **La division entière**
  - Entrée : deux entiers
  - Résultat : quotient
- **Le tri d'une séquence**
  - Entrée : séquence
  - Résultat : séquence ordonnée

Source : Structures de données et algorithmes, Pierre Geurts

(cc) BY-NC-ND

johann.sievering@edu.ge.ch

Programme

- **Structures de contrôle**
  - Séquence
  - Alternative / conditionnelle
  - Itération / boucle

- **Structures de donnée**
  - Constantes
  - Variables
  - Tableau
  - Structures récursives (liste, arbre, graphe, etc.)

(cc) BY-NC-ND

johann.sievering@edu.ge.ch

Complexité / performance

L'analyse de la complexité d'un algorithme : calcul de la quantité de ressources nécessaires à l'exécution de cet algorithme

- **Temps**
  - Mesure du temps utilisé en fonction de la taille de l'entrée.
- **Ressources / Espace / mémoire**
  - Mesure de l'espace mémoire utilisé en fonction de la taille de l'entrée.

IL EST IMPORTANT DE BIEN PENSER SON CODE

(cc) BY-NC-ND

johann.sievering@edu.ge.ch

Bonnes pratiques (1/2)

- Je **planifie** mon travail avant de me lancer
- Je **modélise** les traitements
- J'écris du code **clair** et le plus **simple** possible
- J'écris du code **robuste**
- Je **documente** mon code
- Je **teste** mon code
- Je **sauvegarde** mon code avec des versions

(cc) BY-NC-ND

johann.sievering@edu.ge.ch

Bonnes pratiques (2/2)

"I will, in fact, claim that the difference between a bad programmer and a good one is whether he considers his code or his data structures more important.

Bad programmers worry about the code.

Good programmers worry about data structures and their relationships."

Linus Torvalds (créateur de Linux)

Traduction française

« J'affirme en fait que la différence entre un mauvais et un bon programmeur consiste à savoir s'il considère son code ou ses structures de données comme le plus important.

Les mauvais programmeurs s'inquiètent du code.

Les bons programmeurs s'inquiètent des structures de données et de leurs relations.»

Linus Torvalds (créateur de Linux)

(cc) BY-NC-ND

johann.sievering@edu.ge.ch

STRUCTURES DE CONTRÔLE

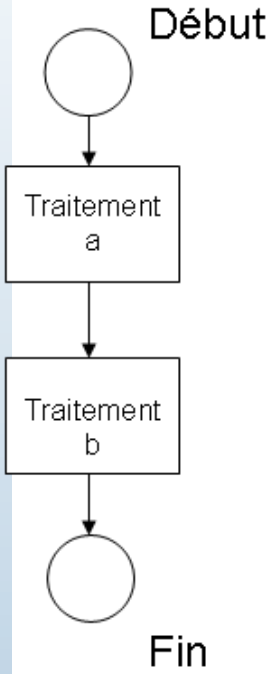
(cc) BY-NC-ND

johann.sievering@edu.ge.ch

STRUCTURES DE BASE

Séquence linéaire

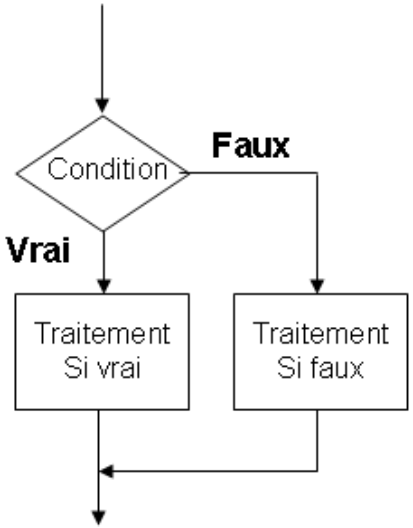
« et puis, et puis, … »



Début
Traitement A
Traitement B
Fin

Alternative

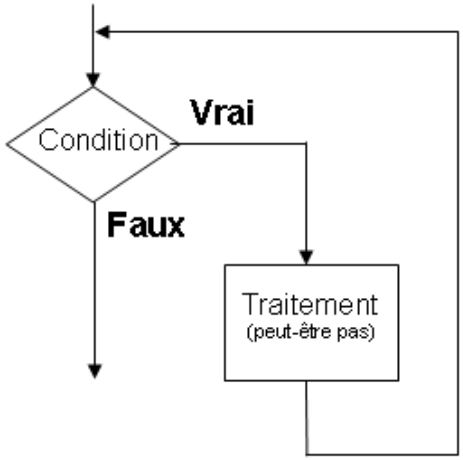
« si…alors…sinon »



Si (condition)
Alors Traitement A
Sinon Traitement B
FinSi

Répétitive

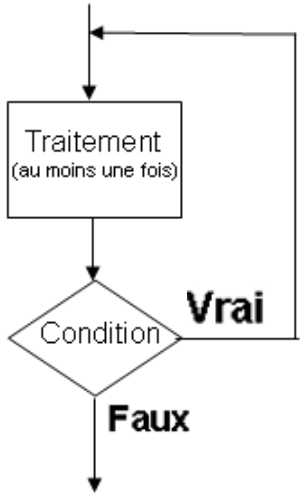
« tant que … faire … »



TantQue (condition)
Faire Traitement A
FinTantQue

Répétitive

« répéter…jusqu'à… »



Répéter (condition)
Faire Traitement A
JusquA (condition)

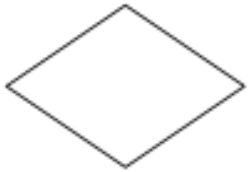
(cc) BY-NC-ND

johann.sievering@edu.ge.ch

Structurer un programme : les symboles

Départ
Fin

Traitement
Opération

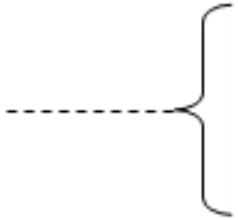


Alternative

Lecture, écriture
Entrée, sortie

Sous-programme

Stockage



Commentaire

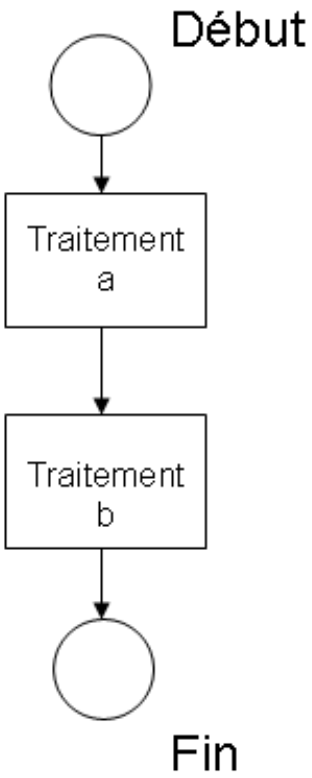
Flux orienté

(cc) BY-NC-ND

johann.sievering@edu.ge.ch

Séquence

« et puis, et puis, … »



Début
  Traitement A
  Traitement B
Fin

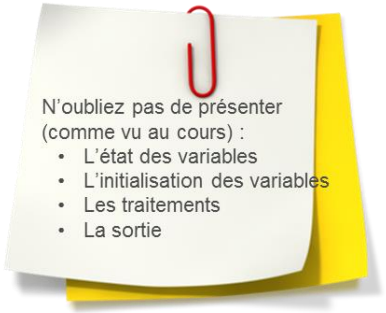
(cc) BY-NC-ND

johann.sievering@edu.ge.ch

Exercice 01

Écrire l'algorithme simple (sous forme d'un organigramme) pour ranger les achats que vous venez d'effectuer dans votre cuisine (armoire et frigo).

Votre sac contient : un pain, une brique de lait, un paquet de pommes et un sachet de fruits secs.

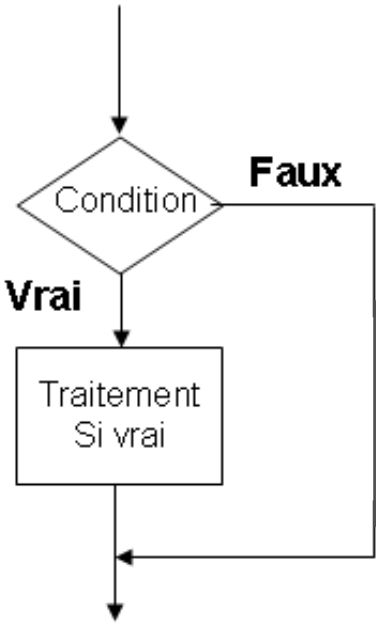


(cc) BY-NC-ND

johann.sievering@edu.ge.ch

Alternative

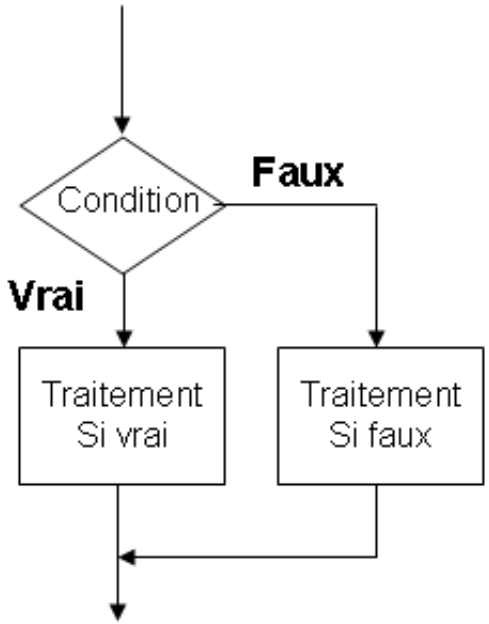
« si…alors… »



Si (condition)
Alors Traitement A

FinSi

« si…alors…sinon »



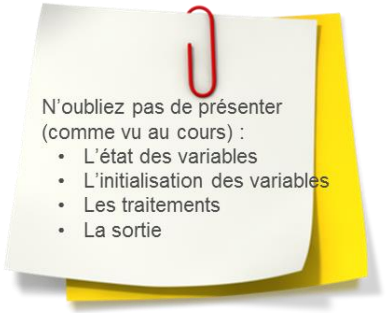
Si (condition)
Alors Traitement A
Sinon Traitement B
FinSi

(cc) BY-NC-ND

johann.sievering@edu.ge.ch

Exercice 02

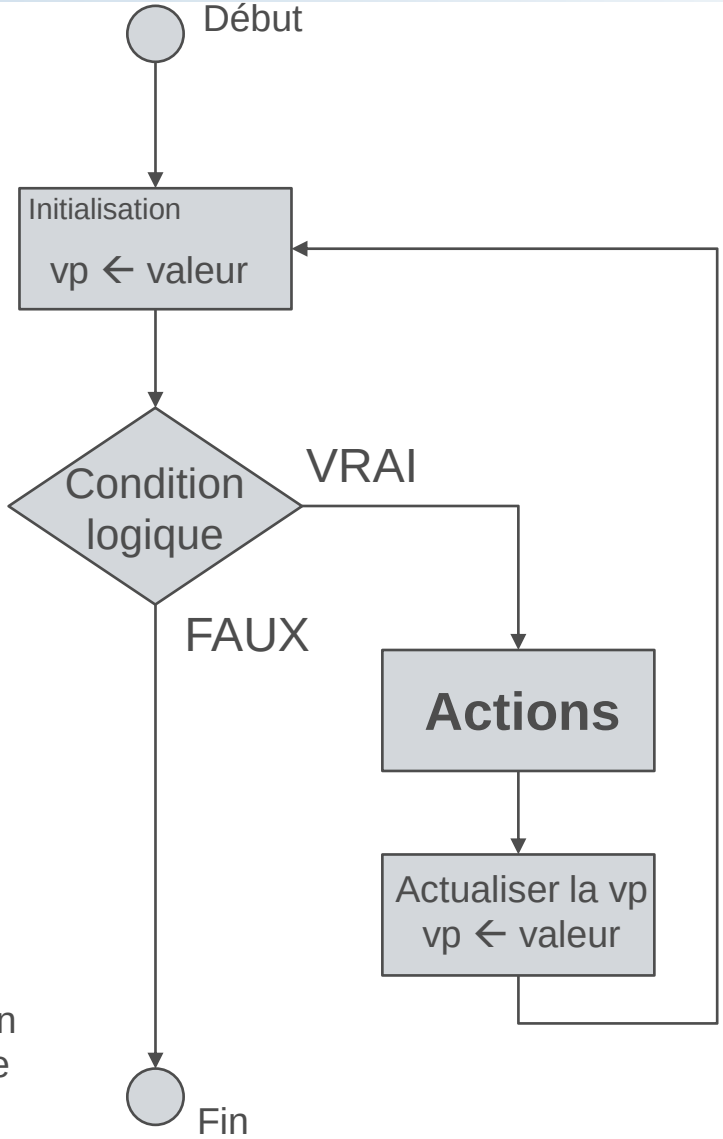
Écrire l'algorithme simple (sous forme d'un organigramme) qui demande un nombre à l'utilisateur au moyen d'un formulaire, qui calcule le carré de ce nombre est qui affiche le résultat sur l'écran.



(cc) BY-NC-ND

johann.sievering@edu.ge.ch

Boucle



Initialisation de la variable de pilotage

Définir la condition d'arrêt en impliquant la variable de pilotage (vp).

Définir les actions à réaliser à chaque itération

Actualiser la variable de pilotage

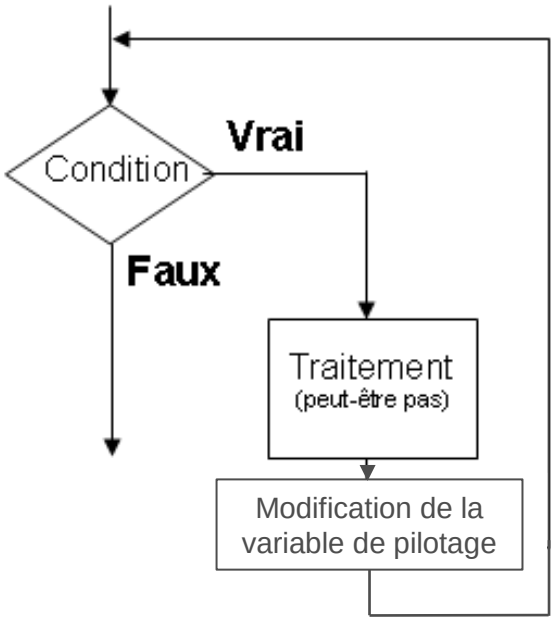
IMPORTANT
Il faut s'assurer que l'évolution de la variable de pilotage converge de telle manière, qu'en un temps fini, la condition devienne fausse (que la boucle s'arrête).

(cc) BY-NC-ND

johann.sievering@edu.ge.ch

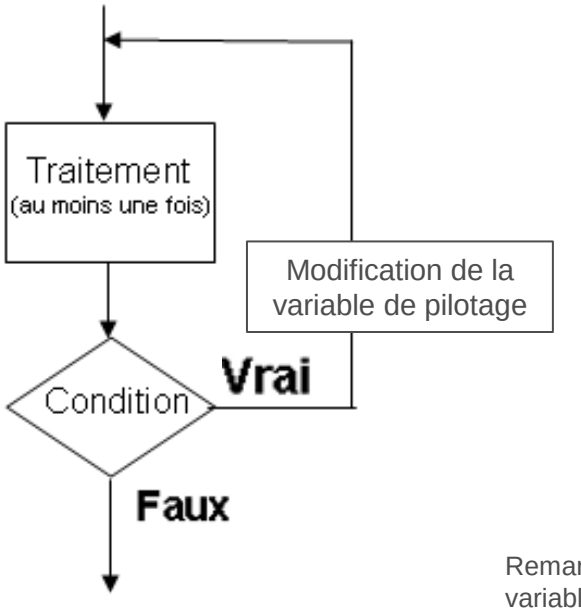
Boucle

« tant que … faire … »



TantQue (condition)
  Faire Traitement A
FinTantQue

« répéter…jusqu'à… »



Répéter (condition)
  Faire Traitement A
JusquA (condition)

Remarque :
variable de pilotage
ou variable de contrôle

(cc) BY-NC-ND

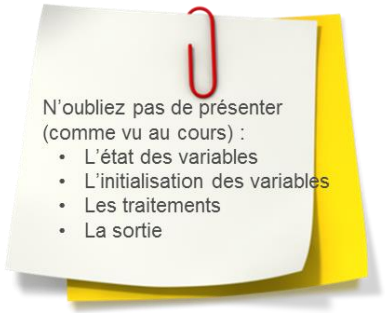
johann.sievering@edu.ge.ch

Exercice 05

Écrire l'algorithme simple (sous forme d'un organigramme) qui affiche les 10 nombres consécutifs de 5 à 15 dans une liste.

Les nombres sont présentés en ligne séparé par une virgule :

[5, 6, 7, 7, 8, 9, 10, 11, 12, 13, 14, 15,]



(cc) BY-NC-ND

johann.sievering@edu.ge.ch

EXERCICES STRUCTURES ALGORITHMES

(cc) BY-NC-ND

johann.sievering@edu.ge.ch

Séquence

N'oubliez pas de présenter (comme vu au cours) :
- L'état des variables
- L'initialisation des variables
- Les traitements
- La sortie
- Les pré- et post-conditions

Écrire l'algorithme (sous forme d'un organigramme) qui décrit l'opération de placer des ingrédients dans un bol dans l'ordre suivant :

- Farine
- Sucre
- Lait

Remarque : les quantités ne sont pas données

(cc) BY-NC-ND

johann.sievering@edu.ge.ch

Alternative : placer des ingrédients

N'oubliez pas de présenter (comme vu au cours) :
- L'état des variables
- L'initialisation des variables
- Les traitements
- La sortie
- Les pré- et post-conditions

Écrire l'algorithme (sous forme d'un organigramme) qui décrit l'opération de remplissage d'un bol de cuisine avec un paquet de farine.
L'utilisateur indique si le paquet est plein ou s'il est vide.

Si le paquet est plein :
    mettre la farine dans le bol.

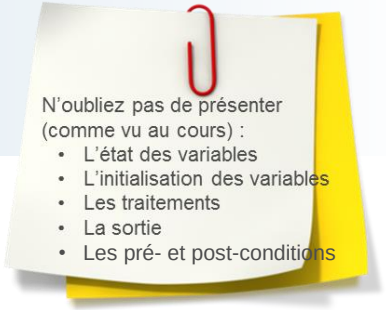
Si le paquet est vide :
    remplir le paquet dans l'armoire
    mettre la farine dans le bol

Le résultat est le bol plein.

(cc) BY-NC-ND

johann.sievering@edu.ge.ch

Itération



Écrire l'algorithme simple (sous forme d'un organigramme) qui décrit la gestion d'un login.

L'application demande à l'utilisateur son login et son mot de passe. Elle compare les saisies avec sa base de données.

Si les saisies sont correctes, l'utilisateur peut accéder à son espace privé et le message « Bienvenue » est affiché.
Sinon, l'application demande à l'utilisateur de tenter une nouvelle saisie et lui affiche le message « veuillez recommencer».

(cc) BY-NC-ND

johann.sievering@edu.ge.ch

LECTURE

(cc) BY-NC-ND

johann.sievering@edu.ge.ch

Lecture

Quelles seront les valeur des variables A et B après l'exécution des instructions suivantes ?

Programme 1

```
var A, B : numérique
DEBUT
    A ← 2;
    B ← A * 2 ;
    A ← 5 ;
FIN
```

Programme 2

```
var A, B, C : numérique
DEBUT
    A ← 7 ;
    B ← 3 ;
    C ← A + B ;
    B ← C * 5 ;
FIN
```


(cc) BY-NC-ND

johann.sievering@edu.ge.ch

Lecture

Décrivez en français ce que fait ce pseudo-code.

Pouvez-vous trouver la fonction de ce programme ?

```
DEBUT
        LIRE quantité
        SI quantité est plus petite que 6
                ECRIRE « prendre carton 1 »
        SINON SI quantité est plus petite que 144
                        ECRIRE « prendre carton 2 »
                SINON SI quantité est plus petite que 400
                                ECRIRE « prendre carton 3 »
                        SINON
                                ECRIRE « erreur : quantité trop grande »
                        FIN SI
                FIN SI
        FIN SI
FIN
```


(cc) BY-NC-ND

johann.sievering@edu.ge.ch

Lecture

Décrivez en français ce que fait ce pseudo-code.

Pouvez-vous trouver la fonction de ce programme ?

```
DEBUT
        nombre <- tirer au hasard un nombre entier entre 0 et 10
        LIRE propositionJoueur
        TANT QUE propositionJoueur pas égale à nombre
                SI nombre plus petit que propositionJoueur
                        ECRIRE « le nombre à trouver est plus petit »
                SINON
                        ECRIRE « le nombre à trouver est plus grand »
                FIN SI
                LIRE propositionJoueur
        FIN TANT QUE
        ECRIRE « bravo »
FIN
```


(cc) BY-NC-ND

johann.sievering@edu.ge.ch

QUESTIONS / RÉPONSES

(cc) BY-NC-ND

johann.sievering@edu.ge.ch