Variables et structures simples

Algorithmique procédurale
ING1

2013–2014

Rémi Vernay
remi.vernay@eisti.eu

I – Structure générale
1 – Algorithme

Algorithme

- Algorithme
  - tâches élémentaires à exécuter dans un ordre donné
- Exemple de la recette de cuisine
  - liste des ingrédients : données en entrée
  - liste des tâches élémentaires à effectuer : algorithme
  - plat terminé : données en sortie

I – Structure générale
2 – Programme

Programme

- Programme
  - contient un/des algorithme(s)

programme
/* *Déclaration des variables* */
/* *Ensemble d'instructions* */
fin programme

I – Structure générale
3 – Variables

Variables

- « Dans une recette de pâte brisée, il faut 250g de farine, 125g de beurre et 1 œuf. »
  - Farine
    - ⋆ nom : quantiteFarine
    - ⋆ valeur : 250
    - ⋆ type : entier

Variables

- ≪ Dans une recette de pâte brisée, il faut 250g de farine, 125g de beurre et 1 œuf. ≫
  - Beurre
    - nom : quantiteBeurre
    - valeur : 125
    - type : entier

Variables

- ≪ Dans une recette de pâte brisée, il faut 250g de farine, 125g de beurre et 1 œuf. ≫
  - Oeuf
    - nom : nbOeufs
    - valeur : 1
    - type : entier

Variables

- ≪ On regarde les indicateurs sur l'obésité et/ou la surcharge pondérale. On nous indique notamment qu'un homme mesure 1m78 et pèse 80kg800. ≫
  - Taille
    - nom : taille
    - valeur : 1,78
    - type : réel

Variables

- ≪ On regarde les indicateurs sur l'obésité et/ou la surcharge pondérale. On nous indique notamment qu'un homme mesure 1m78 et pèse 80kg800. ≫
  - Poids
    - nom : poids
    - valeur : 80,8
    - type : réel

Variables

- « On gère les résultats de QCM à l'aide d'un logiciel. Chaque réponse est considérée comme étant une variable : une réponse peut être vraie ou fausse. »
  - Réponse
    - nom : reponse
    - valeur : FAUX
    - type : booléen

Pseudo-code

- Rigueur nécessaire
- Compréhension sans connaissances informatiques poussées
- Commun à tous les langages impératifs
- Commun à toutes les méthodes de développement
- Commun à toutes les lieux d'enseignement

Pseudo-code

- Déclaration d'une variable
  - nomVariable: type
- Affectation d'une variable
  - nomVariable ← valeur
  - nomVariable ← expression

Pseudo-code

- Exemple

```
quantiteFarine: entier
taille: réel
reponse: booléen
prixHt, prixTtc, tva: réel
quantiteFarine ← 250
taille ← 1,78
reponse ← FAUX
prixHt ← 15,45
tva ← 5,5
prixTtc ← prixHt × tva × 0,01
```

Pseudo-code

- respect absolu des règles de nommage des variables
  - commence par une lettre minuscule
  - ne contient que des lettres ou chiffres
  - une lettre majuscule à chaque changement de mot
  - jamais d'accent sur les lettres

Pseudo-code

- Algorithme qui échange la valeur de 2 variables a et b

```
programme Permuter
    a, b, c: entier
    a ← 1
    b ← 2
    c ← a
    a ← b
    b ← c
fin programme
```

- Au final, a vaut 2 et b vaut 1.

Opérateurs

- Mathématiques
  - $+$: addition
  - $-$: soustraction
  - $\times$: multiplication
  - $\div$: division
  - div : division entière (facultatif)
  - mod : modulo

Opérateurs

- Booléen
  - $=$: égal
  - $\neq$: différent
  - $<$, $\leqslant$: inférieur (ou égal)
  - $>$, $\geqslant$: supérieur (ou égal)
  - ET : conjonction
  - OU : disjonction (non exclusive)
  - NON : négation

Opérateurs

- Chaînes
  - "..." : chaîne de caractères
  - $=$: égal
  - $\neq$: différent
  - & : concaténation

Opérateurs

- Fonctions d'entrées-sorties
  - lire(a) : stocke dans la variable a le contenu tapé au clavier
  - ecrire(a) : affiche à l'écran la valeur de a

Conditions

- si ... alors
- si ... alors ... sinon
- si ... alors ... sinon si ...

si $a < 3$ **alors**
  /* *instructions* */
fin si

Conditions

- Modèle

si a<3 **alors**
  /* *instructions* */
sinon
  /* *instructions* */
fin si

si a<3 **alors**
  /* *instructions* */
sinon
  si a>3 **alors**
    /* *instructions* */
  sinon
    /* *instructions* */
  fin si

Conditions

- Modèle

```
programme Exemple
    note: entier
    lire(note)
    si note>16 alors
        ecrire("C'est bien")
    fin si
fin programme
```


Boucles

- répéter ... jusqu'à ...
- tant que ... faire ...

```
répéter
    /* instructions */
jusqu'à /* condition d'arrêt */

tant que /* condition de continuité */ faire
    /* instructions */
fin tant que
```


Boucles

- Exemple

```
programme Exemple
    n: entier
    n ← 0
    répéter
        ecrire(n)
        n←n+1
    jusqu'à n = 10
fin programme
```


Boucles

- Exemple

```
programme Exemple
    n: entier
    n ← 0
    tant que n<10 faire
        ecrire(n)
        n←n+1
    fin tant que
fin programme
```

Boucles

- pour ... à ... faire
- pour ... à ... par ...

```
pour i←a à b faire
    /* instructions */
fin pour

pour i←a à b par c faire
    /* instructions */
fin pour
```


Boucles

- Exemple

```
programme Exemple
    n: entier
    pour n←0 à 9 faire
        ecrire(n)
    fin pour
fin programme

programme Exemple
    n: entier
    pour n←0 à 30 par 5 faire
        ecrire(n)
    fin pour
fin programme
```