



3 - Paramétrisation des algorithmes : les Fonctions

Jean-Michel Adam

Université Grenoble Alpes

UFR SHS – Département IMSS

Fonction

- Une fonction est un nom associé à un calcul
- En mathématiques, une fonction est définie par
 - Un **nom**
 - Un **domaine de définition**
 - Un **domaine image**
 - Des **règles de calcul** d'une valeur du domaine image à partir d'une valeur du domaine de définition

Fonction

$$f: \mathcal{D} \rightarrow I$$

$\forall x \in \mathcal{D}, f(x) = \text{expression dépendante de } x$

Exemple:

$$f: \mathbb{R} \rightarrow \mathbb{R}$$

$$\forall x \in \mathbb{R}, f(x) = x^2 - x + 1$$

- *En mathématiques, x est appelé une variable*
- *En informatique, x est appelé **paramètre formel** de la fonction*

Fonction

- Une fonction est un sous-algorithme qui, à partir d'une valeur du domaine de définition, calcule et renvoie une valeur du domaine image
- Certaines fonctions sont prédéfinies dans la notation algorithmique, c'est-à-dire mises à disposition pour être directement utilisées.
- Exemples :
 - Partie entière **pent**, partie décimale **pdec**
 - Valeur absolue **abs**
 - **sousChaine**, **longueur** d'une chaîne
- Nous allons maintenant construire nos propres fonctions

Forme générale de réalisation d'une fonction

Traduction informatique du domaine de définition

Traduction informatique du domaine image

En-tête

fonction nomf (liste paramètres formels) → type valeur renvoyée
// spécification : ce que renvoie la fonction
// définition des paramètres

Corps

lexique de nomf
définition des variables locales nécessaires
au calcul de la valeur à renvoyer

algorithme de nomf

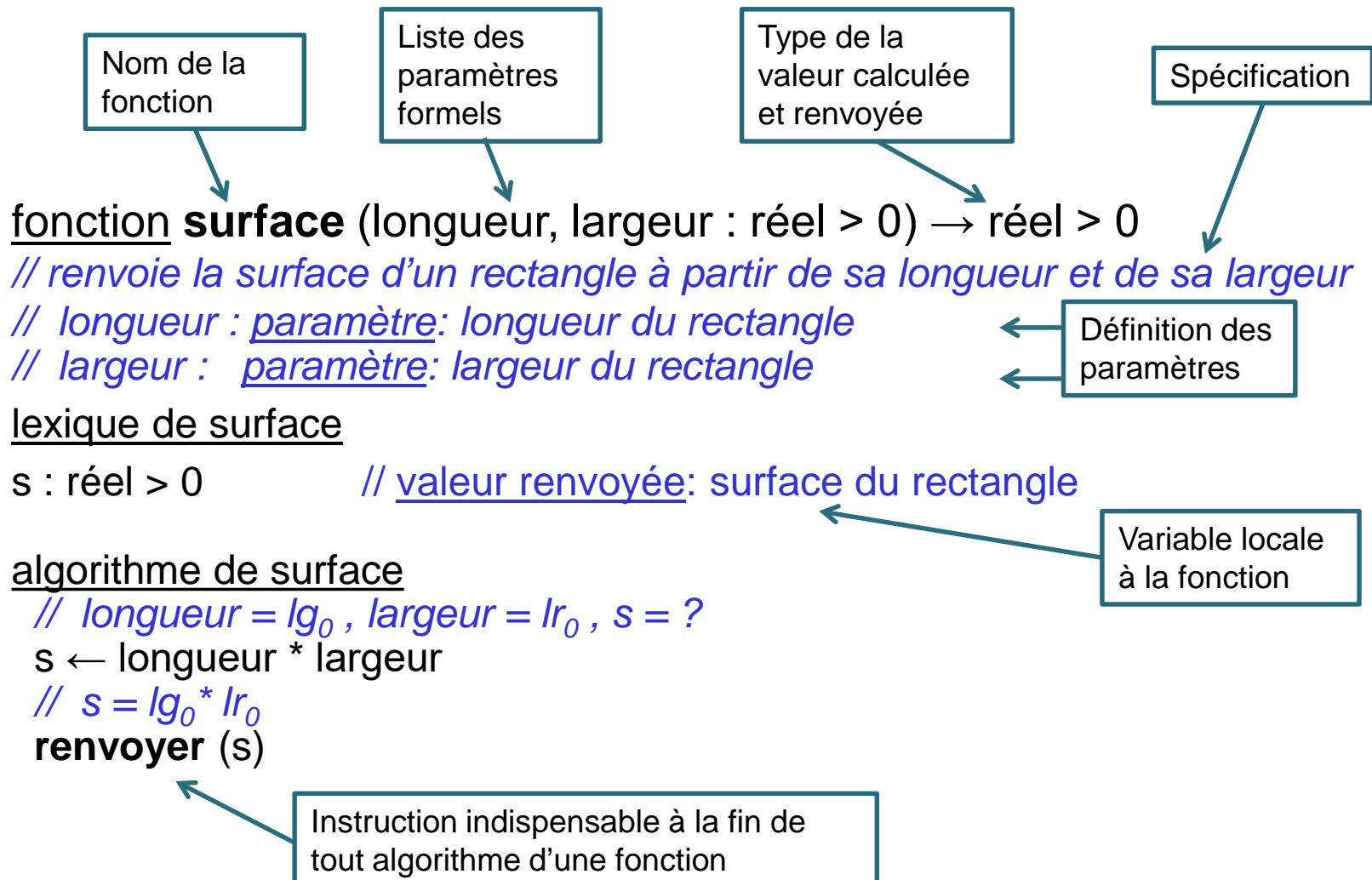
algorithme de calcul de la valeur à renvoyer

renvoyer (valeur calculée)

Expression du type de la valeur renvoyée

Instruction de renvoi de la valeur calculée, indispensable à la fin de tout algorithme d'une fonction

Réalisation informatique d'une fonction qui calcule la surface d'un rectangle



Utilisation d'une fonction

- Une fonction s'utilise dans une expressions arithmétique ou logique
- On parle **d'appel** de la fonction
- Forme de l'appel:
 $f(\text{liste d'arguments})$
- Un appel de la fonction f représente la valeur renvoyée par la fonction f pour les arguments placés entre parenthèses.
- Les arguments sont aussi appelés **paramètres effectifs**

Exemple d'utilisation d'une fonction

lexique principal

cl : clavier // périphérique d'entrée

e : écran // périphérique de sortie

long1, larg1 : réels > 0 // données : dimensions du premier terrain

long2, larg2 : réels > 0 // données : dimensions du second terrain

surfaceTotale: réel > 0 // résultat : surface totale des terrains

Fonction utilisée : surface

Algorithme principal

e.afficher("entrez les dimensions du premier terrain:")

cl.saisir (long1, larg1)

e.afficher("entrez les dimensions du second terrain:")

cl.saisir (long2, larg2)

surfaceTotale ← surface(long1, larg1) + surface(long2, larg2)

e.afficher("Surface totale : ", surfaceTotale)

Arguments du 1^{er} appel

Deux appels de la fonction surface

Arguments du 2nd appel

Réalisation d'une fonction qui calcule le minimum de deux valeurs

$\text{min2v} : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$

$\forall x \in \mathbb{R}, \forall y \in \mathbb{R}, \text{min2v}(x,y) = x \text{ si } x \leq y$

$\text{min2v}(x,y) = y \text{ si } x > y$

Définition mathématique
de la fonction

fonction **min2v**(x, y : réels) → réel

// min2v(x,y) renvoie la valeur minimale de x et de y

// x, y : paramètres: valeurs dont on calcule le minimum

lexique de min2v

min : réel // valeur renvoyée: minimum de x et de y

algorithme de min2v

// x = x₀, y = y₀, min = ?

selon x,y

x > y : min ← y

x ≤ y : min ← x

fselon

renvoyer (min) // min = minimum(x₀, y₀)

Réalisation informatique de la fonction

De la même manière on peut définir la fonction max2v qui calcule la valeur maximale de 2 réels

Exercice : Calcul de la note finale d'un étudiant

- Utiliser les fonctions `min2V` et `max2v` pour résoudre le problème suivant :
 - La note finale d'un étudiant dans une matière est calculée à partir des trois notes qu'il a obtenues dans cette matière, en prenant la meilleure parmi les deux notes intermédiaires suivantes :
 - 1 - la moyenne de ces trois notes initiales,
 - 2 - la moyenne de sa meilleure et de sa plus mauvaise note.
 - Ecrire un algorithme qui saisit le nom et les trois notes d'un étudiant, calcule les deux notes intermédiaires et affiche la note finale.

Exercice : utilisation des fonctions min2V et max2v pour résoudre le problème posé

lexique principal

cl : clavier // périphérique d'entrée

e : écran // périphérique de sortie

n1, n2, n3: réels ≥ 0 // données : notes obtenues

nom : chaîne // donnée : nom de l'étudiant

nf : réel ≥ 0 // résultat : note finale de l'étudiant

moy : réel ≥ 0 // intermédiaire : moyenne des 3 notes

moyminmax : réel ≥ 0 // intermédiaire : moyenne meilleure et moins bonne

Fonctions utilisées : min2v, max2v

Algorithme principal

e.afficher("entrez le nom de l'étudiant et les notes obtenues :")

cl.saisir (nom, n1, n2, n3)

moy $\leftarrow (n1 + n2 + n3) / 3$

moyminmax $\leftarrow (\min2v(n1, \min2v(n2, n3)) + \max2v(n1, \max2v(n2, n3))) / 2$

nf $\leftarrow \max2v(\text{moy}, \text{moyminmax})$

e.afficher(nom, " a obtenu la note finale de : ", nf)

Composition fonctionnelle : construction de fonctions à partir de fonctions existantes

Réalisation de la fonction min3v qui calcule le minimum de 3 réels :

fonction $\text{min3v}(x, y, z : \text{réels}) \rightarrow \text{réel}$

// $\text{min3v}(x,y,z)$ renvoie la valeur minimale de x , de y et de z

// x, y, z : paramètres: valeurs dont on calcule le minimum

lexique de min3v

$\text{min} : \text{réel}$ *// valeur renvoyée: minimum de x , de y et de z*

fonction utilisée : min2v

algorithme de min3v

$\text{min} \leftarrow \text{min2v}(\text{min2v}(x,y), z)$

renvoyer (min)

Composition fonctionnelle : construction de fonctions à partir de fonctions existantes

Réalisation de la fonction min3v qui calcule le minimum de 3 réels : **autre écriture possible**

fonction $\text{min3v}(x, y, z : \text{réels}) \rightarrow \text{réel}$

// $\text{min3v}(x,y,z)$ renvoie la valeur minimale de x , de y et de z

// x, y, z : paramètres: valeurs dont on calcule le minimum

lexique de min3v

algorithme de min3v

renvoyer ($\text{min2v}(\text{min2v}(x,y), z)$)

Réalisation de fonctions utiles pour manipuler les chaînes

Exercice : réaliser les fonctions suivantes:

pre : $C^+ \rightarrow C$

$\forall x \in C^+, \text{pre}(x) = 1^{\text{er}} \text{ caractère de } x$

der : $C^+ \rightarrow C$

$\forall x \in C^+, \text{der}(x) = \text{dernier caractère de } x$

deb : $C^+ \rightarrow C^*$

$\forall x \in C^+, \text{deb}(x) = x \text{ privée de son dernier caractère}$

fin: $C^+ \rightarrow C^*$

$\forall x \in C^+, \text{fin}(x) = x \text{ privée de son } 1^{\text{er}} \text{ caractère}$

On a : $x = \text{pre}(x) \circ \text{fin}(x) = \text{deb}(x) \bullet \text{der}(x)$

Réalisation de fonctions utiles pour manipuler les chaînes : pre

fonction pre(x : chaîne) → caractère

// pre(x) renvoie le premier caractère de x

// x : paramètre: chaîne dont on extrait le premier caractère

lexique de pre

fonction utilisée : nième

algorithme de pre

renvoyer (nième(x,0))

Réalisation de fonctions utiles pour manipuler les chaînes : der

fonction der(x : chaîne) → caractère

// der(x) renvoie le dernier caractère de x

// x : paramètre: chaîne dont on extrait le dernier caractère

lexique de der

fonction utilisée : nième, longueur

algorithme de der

renvoyer (nième(x, longueur(x)-1))

Réalisation de fonctions utiles pour manipuler les chaînes : deb

fonction deb(x : chaîne) → chaîne

// deb(x) renvoie x privée de son dernier caractère

// x : paramètre: chaîne dont on extrait le début

lexique de deb

fonction utilisée : sousChaîne, longueur

algorithme de deb

renvoyer (sousChaîne(x,0,longueur(x)-1))

Réalisation de fonctions utiles pour manipuler les chaînes : fin

fonction fin(x : chaîne) → chaîne

// fin(x) renvoie x privée de son premier caractère

// x : paramètre: chaîne dont on extrait la fin

lexique de fin

fonction utilisée : sousChaîne, longueur

algorithme de fin

renvoyer (sousChaîne(x,1,longueur(x)-1))

Exemple d'usage

// Conjugaison d'un verbe du 1^{er} groupe au présent de l'indicatif :
lexique principal

cl : clavier // périphérique d'entrée

e : écran // périphérique de sortie

verbe : chaine // donnée : verbe à conjuguer

radical : chaine // intermédiaire : radical du verbe à conjuguer

Fonction utilisée: deb

Algorithme principal

e.afficher("entrez un verbe du premier groupe:")

cl.saisir (verbe) // verbe = v_0

radical ← deb(deb(verbe)) // radical = radical de v_0

e.afficher ("je ", deb(verbe))

e.afficher ("tu ", radical & "es")

e.afficher ("il ou elle", deb(verbe))

e.afficher ("nous ", radical, "ons")

e.afficher ("vous ", deb(verbe)•'z')

e.afficher ("ils ou elles " & radical & "ent")

Paramètres et Arguments

- Pour chaque appel d'une fonction, le nombre **d'arguments** doit correspondre au nombre de **paramètres formels** de la fonction.
- Les arguments peuvent être des expressions
- Le **type** du nième argument doit correspondre au type du nième paramètre.
- Si le nième argument est un nom, il aura, de préférence, un nom différent de celui du nième paramètre.

Fonctionnement de l'appel d'une fonction

- Au moment de l'appel de la fonction, les valeurs des arguments sont affectés aux paramètres
 - 1^{er} paramètre ← valeur du 1^{er} argument
 - 2^{ème} paramètre ← valeur du 2^{ème} argument
 - etc.
- La valeur renvoyée par la fonction doit être du type indiqué dans la spécification de celle-ci.
- La valeur renvoyée est exploitée dans l'expression à l'endroit où apparaît l'appel de la fonction.

