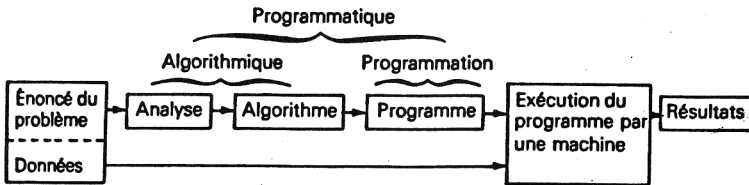


L'ENSEIGNEMENT DE LA PROGRAMMATIQUE AU CENTRE DE FORMATION À L'INFORMATIQUE PÉDAGOGIQUE DES ENSEIGNANTS DE VILLETANEUSE (ACADÉMIE DE CRÉTEIL)

J.-L. JOURDAN, Ch. RICHARD

Chaque année nous essayons d'améliorer notre pédagogie aussi nous avons jugé intéressant de faire partager nos dernières expériences dans le domaine de l'enseignement de la programmation.

Depuis l'an dernier nous n'avons plus seulement un cours d'algorithmique précédant l'enseignement de tous les langages mais un cours de programmation¹, art de la bonne programmation selon le schéma suivant :



La programmation a en programmation une situation assez semblable à celle de la théorie des ensembles en mathématiques. La théorie des ensembles permet une présentation très intéressante d'un cours de mathématiques en l'exposant entièrement en termes d'ensembles.

Or une démarche du même type peut être réalisée en programmation.

¹ Ce mot a été introduit par C. et P. RICHARD dans " Initiation à l'algorithmique" paru chez BELIN

En début d'année nous formons les stagiaires à l'esprit d'analyse descendante structurée utilisant la méthode ALADIN qui les mène à écrire un algorithme qu'ils peuvent exécuter grâce au logiciel ALADIN.

Lorsque nous pensons qu'ils ont assimilé les notions algorithmiques de base c'est à dire qu'ils sont capables d'analyser en français jusqu'à l'écriture algorithmique les problèmes classiques, nous passons à l'étude des langages de programmation.

Nous essayons d'éviter de tomber dans le piège de la présentation syntaxique même au niveau de l'enseignement d'un langage. Pour cela nous veillons systématiquement à ce qu'un algorithme soit écrit avant de passer à son codage et ce n'est qu'à partir des algorithmes déjà étudiés que les éléments de syntaxe spécifiques d'un langage de programmation sont introduits peu à peu.

Jusqu'à l'année dernière nous utilisions cette méthode pour les langages de type PASCAL, LSE mais nous avons voulu aller plus loin en introduisant LOGO lui-même à partir d'ALADIN afin de créer une unité pédagogique dans l'enseignement de la programmation.

Nous allons choisir un exemple ² (*) dont l'algorithme est écrit en ALADIN

En fin de semaine un représentant enregistre ses ventes afin de calculer son chiffre d'affaires. Chaque produit vendu est caractérisé par trois données: un numéro de code, la quantité vendue dans la semaine et le prix unitaire. Les produits doivent être traités dans l'ordre croissant des numéros de produit.

L'analyse

\$: Calcul du chiffre d'affaires hebdomadaire d'un représentant

\$1 : Entrée des données concernant le premier produit : 1\$

2\$ TANT QUE il y a encore des produits à traiter REPETE

\$21: SI le produit est bien classé

ALORS \$211: Calcul du prix de vente de ce produit

Cumul avec le prix de vente déjà calculé

: 211\$

SINON \$212: erreur, le produit n'est pas bien classé

: 212\$

Passage au produit suivant

² Voir p. 97 du livre de C. et P. RICHARD *Programmative* paru chez BELIN.

:\$
 :2\$
 \$3: Le chiffre d'affaires total a été calculé :3\$
 :\$
 * Voir p 97 du livre de C et P RICHARD "Programmétique" paru chez BELIN.

L'algorithmme

```
$
                                CHIFFRE D'AFFAIRES
ECRIS "Calcul du chiffre d'affaires d'un représentant"
$1 : ECRIS Donner pour le premier produit, dans l'ordre son "
"numéro, la quantité vendue et son prix unitaire"
LIS NPROD,QUANT,PRIX
EN PRECED RANGE .0 EN CA RANGE 0
:1$
$2: TANT QUE NPROD <> 0 REPETE
$21: %traitement d'un produit%
SI NPROD > PRECED
ALORS $211: % Prix de vente de et produit%
EN PVENT RANGE QUANT*PRIX
EN PRECED RANGE NPROD
EN CA RANGE CA+PVENT
:211$
SINON $212: ECRIS "Erreur sur le numéro de ",
" produit ",NPROD
:212$
ECRIS "Taper les données d'un autre produit"
LIS NPROD,QUANT,PRIX
:21$
:2$
ECRIS "Le chiffre d'affaires de la semaine est ", CA, "F" :3$
:$
```

La traduction LOGO proposée est la suivante ³ :

3 La traduction est faite sans les commentaires.

Pour les incorporer il suffit d'utiliser la primitive % :

définie par : POUR % :

FIN où :L est une liste

Il s'agit d'une simple transposition des blocs et des instructions sauf pour la structure répétitive

"TANT QUE ... REPETE"

située au début du bloc n° 2 et qui concerne le bloc n° 21.

POUR \$
 ECRIS [CALCUL DU CHIFFRE D'AFFAIRE D'UN REPRESENTANT]
 \$1
 \$2
 \$3
 FIN

POUR \$1
 ECRIS [DONNER POUR LE PREMIER PRODUIT, DANS L'ORDRE,
 SON NUMERO; LA
 ECRIS [QUANTITE VENDUE ET SON PRIX UNITAIRE]
 RELIE "NPROD PREM LISLISTE
 RELIE "QUANT PREM LISLISTE
 RELIE "PRIX PREM LISLISTE
 RELIE "PRECED 0 RELIE "CA O
 FIN

POUR \$2
 SI (:NPROD # 0)ALORS [REPETES\$21]
 FIN

POUR \$3
 TAPE [LE CHIFFRE D'AFFAIRES DE LA SEMAINE EST :]
 TAPE :CA ECRIS "F
 FIN

POUR REPETES\$21
 SI (:NPROD > :PRECED >ALORS [\$211] SINON [\$212]
 ECRIS [TAPER LES DONNEES D'UN AUTRE PRODUIT]
 RELIE "NPROD PREM LISLISTE
 RELIE "QUANT PREM LISLISTE
 RELIE "PRIX PREM LISLISTE
 SI (:NPROD # 0)ALORS (REPETES\$21]
 FIN

POUR \$211
 RELIE "PVENT (:QUANT * :PRIX)
 RELIE "PRECED :NPROD
 RELIE "CA (:CA + :PVENT)
 FIN

```
POUR $212
TAPE [ERREUR SUR LE NUMERO DE PRODUIT ]
ECRIS :NPROD
FIN
```

Pour montrer qu'il est répété, le bloc n° 21 est appelé REPETE \$21 et le bloc n° 2 s'écrit alors sous la forme :

```
POUR $2
SI (:NPROD t 0 )ALORS [REPETES$21 ]
FIN
```

peu différente de l'écriture ALADIN.

Par contre, le bloc n° 21 appelé REPETE \$21 doit se terminer par :

```
SI (:NPROD 0 )ALORS [REPETES$21 ]
```

La structure répétitive d'ALADIN est ainsi remplacée par une récursivité terminale de LOGO.

Il va de soi que LOGO n'est pas le seul champ d'applications de la méthode précédente. Ainsi, par le même procédé, le même algorithme ALADIN se traduit directement en L.S.E sous la forme

```
1 &D());TERMINER
2 PROCEDURE &D()
4 AFFICHER[U,/]'CALCUL DU CHIFFRE D'"AFFAIRE D'"UN REPRESENTANT';
   &D1() ;&D2() ; &D3()
5 RETOUR
6 PROCEDURE &D1()
7 AFFICHER2/,'DONNER DANS L'ORDRE :
   ',NUMERO:'L,3X,'QUANTITE:'L,3X,'PRIX UNITAIRE:']
8 LIRE[U,/U,/U/]NPROD,QUANT,PRIX
9 PRECE←O;CA←O
10 RETOUR
11 PROCÉDURE &D2()
12 SI NPROD#O ALORS &RD21()
13 RETOUR
14 PROCEDURE &D3()
15 AFFICHER[3U/]'LE CHIFFRE D'"AFFAIRE DE LA SEMAINE EST: ',CA,'F'
16 RETOUR
17 PROCEDURE &RD21()
18 SI NPROD>PRECE ALORS &D211() ,SINON &D212()
19 AFFICHER[U,/]'TAPER LES DONNÉES D'"UN AUTRE PRODUIT: '
20 LIRECU[U,/U,/U/]NPROD,QUANT,PPRIX
21 SI NPROD#O ALORS &RD21()
22 RETOUR
LE BULLETIN DE L'EPI
```

23 PROCEDURE &D211()
24 PVENT*QUANT*PRIX
25 PRECE+NPROD
26 CA←CA+PVENT
27 RETOUR
28 PROCÉDURE &D212(>
29 AFFICHER[2U,/]'ERREUR SUR LE NUMERO DE PRODUIT',NPROD
30 RETOUR

Les langages tels que LOGO, ou LISP, méritent toutefois qu'on s'y attarde car il est certain que leurs utilisateurs acquièrent rapidement une forme d'esprit assez particulière.

La situation parait alors assez semblable à ce qui s'est produit en mathématiques avec l'apparition de la théorie des catégories.

De même que tout exposé ensembliste est implicitement exprimé en termes de catégories, toute programmation en ALADIN contiendrait alors une programmation en termes d'une théorie existant déjà implicitement dans ALADIN.

J.-L. JOURDAN
Ch. RICHARD