

INIT est un tout petit langage impératif donné à titre d'illustration, dont les instructions sont des initialisations. On utilisera deux versions, l'une avec des types simples et l'autre étendue avec un type liste.

1 Version simple

Un programme INIT déclare des variables entières, puis les initialise soit par lecture au clavier, soit par affectation d'une constante entière. Une variable ne peut être initialisée que si elle a été déclarée, et n'est initialisée qu'une fois. Les identificateurs d'INIT sont ceux de Java (une lettre ou `_` suivie éventuellement d'une suite de lettres, chiffres ou `_`). Les entiers sont des suites de chiffres.

Structure d'un programme

- Un *programme* est constitué d'un en-tête obligatoire, suivi d'une partie déclaration puis d'une partie instruction optionnelles.
- Un *en-tête* commence par le mot-clé **program**, suivi d'un identificateur, suivi d'un point-virgule (`;`).
- La partie déclaration consiste en une *liste de déclarations*. Une *déclaration* déclare une ou plusieurs variables de type entier, et commence par le mot-clé **int**. Suit ensuite une liste d'identificateurs de variable séparés par des virgules (`,`). Une déclaration termine par un point-virgule.
- La partie instructions consiste en une *liste d'instructions*. Une *instruction* est soit une affectation, soit une lecture, et termine par un point-virgule.
- Une *affectation* affecte à une variable de type entier une constante entière. Elle consiste en un identificateur de variable suivi des caractères deux-points et égal (`:=`) suivis d'un entier.
- Une *lecture* commence par le mot-clé **read** suivi d'un identificateur de variable de type entier.

Exemples de programmes corrects De par la syntaxe du langage on peut écrire un programme INIT qui ne semble pas avoir grand sens (par exemple qui ne déclare pas de variables) : INIT est un langage à but uniquement pédagogique!

Un programme minimal :

```
program monprog ;
```

Un programme qui déclare une variable :

```
program      monprog2;
```

```
int x;
```

Un programme qui déclare des variables et les initialise :

```
program mon_prog3;
int x,y;
int z;
read z;
y := 3;
x := 345;
```

Exemples de programmes incorrects Un programme qui contient des unités lexicales non autorisées (comme **write** par exemple) ou ne respecte pas la grammaire du langage (par exemple entrelace déclarations et initialisations) est bien sûr incorrect. Par ailleurs un programme lexicalement et syntaxiquement correct peut être sémantiquement incorrect, par exemple :

```
program bourde1;  
int x;  
read y;
```

ou

```
program bourde2;  
int x;  
x := 3; read x;
```

2 Version étendue

On étend maintenant INIT avec un nouveau type de données : les *listes d'entiers*. On peut donc déclarer des variables de type liste (d'entiers implicitement). Ces variables ne peuvent être initialisées par une opération de lecture : elles sont uniquement affectées. On représente une liste d'entiers par une liste éventuellement vide d'éléments encadrée par [et]. Un élément peut être soit un entier, soit une liste.

Structure d'un programme Elle reste globalement la même que dans la version simple.

- Une *déclaration* déclare soit des variables de type entier (auquel cas elle commence par le mot-clé **int**), soit des variables de type liste (auquel cas elle commence par le mot-clé **list**). Le mot-clé est suivi comme auparavant par une liste de variables.
- Une *affectation* affecte à une variable de type entier une constante entière ou à une variable de type liste une liste d'entiers.
- Une *lecture* commence par le mot-clé **read** suivi d'un identificateur de variable de type entier.

Exemple de programme correct

Un programme qui utilise des listes :

```
program progliste;  
list x,y,z;  
y := [ [3] [2 1] 1];  
x := [ [ [1] 3 ] 5 ];  
z := [];
```