

I. Les unités

```
unit nomDuProgramme;  
interface  
    { Partie publique. Contient appels d'unités, types, constants, signatures des sous-  
      programmes. }  
implementation  
    { Partie privée. Contient appels d'unités, types privés, constantes privées, corps  
      des sous-programmes publics et privés. }  
initialization  
    { code executé au chargement de l'unité }  
end.
```

En cas d'inclusion circulaire d'unités, au moins une inclusion doit se faire dans la partie implementation.

II. Tests unitaires

Le but est de tester chaque sous-programme sur plusieurs exemples significatifs (cas général et cas particuliers) afin de vérifier leur bon fonctionnement.

```
program tests;  
uses uniteTestee;  
procedure testFct1();  
begin  
    write('fct1 : ');  
    if fct1(val) = val then  
        writeln('OK')  
    else  
        writeln('KO')  
    end;  
...  
begin  
    writeln('Tests de l''unité uniteTestee :');  
    testFct1();  
    testFct2();  
    ...  
end.
```

III. Documentation RobotDOC

```
{**** ?*   ? = d (constante) / f (fonction) / h (unité) / s (record) / t (type) / v (var)  
* NAME / AUTHOR / CREATION DATE / DESCRIPTION / FUNCTION / INPUTS / OUTPUT / EXAMPLE  
* Texte de l'élément choisi. Exemples :  
* NAME  
*   nomFonction  
* INPUTS  
*   * x (Type) : description  
*   * y (Type2) : description  
*   ...  
****}  
unit / fonction / procedure / type / var ...
```