

I. Les exceptions

Avant, les ordinateurs plantés suite à des erreurs de programmation. Aujourd'hui, ils plantent... Mais bien moins souvent que si les exceptions et leurs traitements n'avaient été créés.

Soit par exemple le code suivant :

```
var X, Y : double ;
begin
    Y := ln(1) ; // Renvoie 0
    X := 3.14 / Y ;
    ...
```

Lors de la division par Y qui vaut zéro, l'ordinateur se bloque et ne répond plus aux actions de l'utilisateur. Si rien n'a été prévu, il faut redémarrer l'ordinateur.

Il suffit de faire un test pour prévenir cette erreur :

```
var X, Y : double ;
begin
    Y := ln(1) ; // Renvoie 0
    if (abs(Y)>1e-15) then
        X := 3.14 / Y ;
    ...
```

Mais parfois, on oublie, où l'on pense que ce cas est impossible... L'erreur pouvant se produire est dans ce cas très simple à détecter. Parfois il est « presque impossible » de tout vérifier auquel cas il faut gérer les erreurs et prémunir l'utilisateur d'un éventuel blocage du système.

Les instructions Try Finally End et Try Except End permettent de mieux contrôler les erreurs.

A. Try Finally End

L'instruction try finally end permet d'exécuter un bloc de codes entre try et finally et de toujours exécuter le code entre finally et end « quoi qu'il arrive »¹.

Par exemple, on pourra utiliser le bloc de la manière suivante ce qui garantit que l'objet est toujours libéré à la fin de l'exécution du code même si une exception survient :

```
var V : TVecteur ; // où TVecteur est une class
begin
    V := TVecteur.Create ;
    try
        V. ... ;
        // code
    finally
        V.Free ;
    end ;
end ;
```

¹ Certaines erreurs passent au travers de la gestion des exceptions sans aucune raison apparente... si ce n'est d'être en général de très très très grosses erreurs ☹.

B. Try Except End

L'instruction try except end permet de contrôler le type d'erreur survenu et d'agir en conséquence.

```
var V : TVecteur ; // où TVecteur est une class
begin
    V := TVecteur.Create ;
    try
        V. ... ;
        // code
    except
        // traitement de l'exception ici
        // ex simple
        on EDivByZero do Result := 0;
        on E : Exception do ShowMessage('Erreur :' + E.message) ;
    end ;
end ;
```

C. Levée une exception

Il est possible de lever une exception à l'aide du code type suivant :

```
raise Exception.Create('Une erreur est survenue ici ...') ;
```

Enfn, rappelons qu'il est possible de créer ses propres classes d'exception.

Le traitement des exceptions est complexe mais indispensable. Reportez-vous à l'aide en ligne de Delphi pour de plus amples informations.