

TD-TP 19 c'est le dernier !

Thème : L'objectif est de mettre en oeuvre un certain nombre de concepts peu vus jusqu'à ce jour dans les autres TD-TP. Le TD sera très court on exposera rapidement le problème et on passera en TP. Là, il convient de tester si vous êtes capable de faire une petite synthèse tout seuls !

Concepts : Interfaçage avec le langage C, examen et utilisation de la ligne de commande, fichier de flots, fichier texte, nombre delta.

Le problème :

Voici une copie du programme de test qui utilise le paquetage à réaliser.

```
with P_Tp19;           use P_Tp19;
with Ada.Text_IO;      use Ada.Text_IO;
with Ada.Exceptions;   use Ada.Exceptions;
procedure Tp19 is
begin
    Controle_Commande ;
    Controle_Acronyme ;
    Calcul_Moyenne ;
    Envoi_Moyenne ;

--    Skip_Line;
--exception
--    when Marqueur : others =>
--        Put_Line(Exception_Information(Marqueur));
--        Skip_Line;

end Tp19;
```

-- un fichier de flot contient des séries de couples formés :
-- d'un acronyme et d'une suite de 10 notes maximum voire aucune !
-- l'utilisateur lance TP19 pour émettre la moyenne des notes
-- d'un acronyme donné.
-- la commande doit comporter deux arguments :
-- 1° un caractère N ou C (en minuscule ou en majuscule)
-- 2) l'acronyme pour la moyenne cherchée
-- si la commande est incorrecte l'exception Exc_Commande est levée
-- si l'acronyme n'est pas dans la base de données (c'est un fichier)
-- alors l'exception Exc_Acronyme est levée.
-- Après calcul de la moyenne des notes (10 maxi voire aucune !)
-- On édite la moyenne ou on l'envoie par mail suivant
-- que l'argument n°1 est N (normal on édite la moyenne)
-- ou C (confidentiel on envoie la moyenne par mail)
-- la récupération des exceptions s'imposera

Il est clair que le travail à réaliser est dans les 4 sous programmes utilisés ci dessus.
Voici les spécifications du paquetage :

```
package P_Tp19 is
  Exc_Commande,
  Exc_Acronyme : exception;

  procedure Controle_Commande;
  -- deux arguments, le premier un unique caractère N ou C ou n ou c
  -- le deuxième est contrôlé ensuite

  procedure Controle_Acronyme;
  -- le deuxième argument est dans le fichier de flot

  procedure Calcul_Moyenne;
  -- les notes sont dans l'élément qui suit l'acronyme dans le fichier
  -- il y a de zéro à 10 notes au plus

  procedure Envoi_Moyenne;
  -- le type d'envoi dépend du premier argument

private

  type T_Nb_Note is range 0 .. 10;
  subtype T_Nb_Note_Pos is T_Nb_Note range 1..T_Nb_Note'Last;
  type T_Val_Note is delta 0.01 range 0.0 .. 20.0;
  type T_Vect is array (T_Nb_Note_Pos range <>) of T_Val_Note;
  type Enreg
    (Nb : T_Nb_Note := 0) is
    record
      Vect : T_Vect (1 .. Nb);
    end record;
end P_Tp19;

-- structure du fichier de flot (base de données) :
-- succession (éventuellement vide) de :
-- 1° un Unbounded_String (l'acronyme)
-- 2° un article mutant formé d'un discriminant du type
-- nombre de notes (0..10) et d'un vecteur de notes de type
-- delta 0.01 à valeur dans 0..20
-- voir la partie privée
```

Quand vous aurez réalisé le corps de ce paquetage vous ferez un gnatmake tp19
Puis vous testerez !

Quelques tests possibles :

- Pas d'arguments
- Un seul argument
- Trois arguments
- Deux arguments mais le premier n'est pas conforme
- Deux arguments mais le deuxième n'est pas dans la base (il s'agit d'un fichier texte contenant un acronyme par ligne)

- Enfin deux arguments en alternant N et C comme premier argument et des notes en nombre de 0 à 10 (voire plus !)