

```

*****
*****fichier p_fusion.ads*****
-----
-- paquetage offrant la fonction de fusion de 2 fichier tries
-- en un fichier unique
--
-- parametres du paquetage:
-- le type d'objet que contiennent les fichiers
-- la fonctions d'ordre permettant la comparaison de 2 objets
-- la fonction d'egalite de 2 objets
-- les procedures de lecture et d'ecriture dans les fichiers d;objets
-----

with Text_io; use text_io ;

generic

    type T_Objete is limited private ; -- le type a` manipuler

    with function ">" (O1, O2 : T_Objete) return Boolean is <>;
        -- la fonction de comparaison d'ordre

    with function Egal (O1, O2 : T_Objete) return Boolean is "=";
        -- la fonction de test d'e'galite'

    with procedure Lecture (F: File_Type; Item : out T_Objete) ;
    with procedure Ecriture (F: File_Type; Item : T_Objete) ;
        -- les fonctions d'entree sortie sur les fichiers

package P_FUSION is

procedure fusion(fusion1,fusion2,resultat:string);
--
-- =====
-- precondition : les trois parametres concernent des fichiers
-- existants
-- : les fichiers ne contiennent pas de valeurs
-- de cles egales
-- re'sultat :
-- cette procedure realise l'interclassement de deux
-- fichiers file1 et file2, fichiers d'objets tries par
-- ordre croissant, le resultat se trouve dans resultat
-- ces trois fichiers doivent exister,
-- en cas de valeurs egales, seule une seule est sortie.
--
-- si les fichiers n'existent pas, l'exception NAME_ERROR
-- est levee

end P_FUSION;
*****
*****fichier p_fusion.adb*****
with Text_Io; use Text_Io;

package body P_FUSION is

procedure fusion(fusion1,fusion2,resultat:string) is
--
-- =====
-- cette procedure realise l'interclassement de deux
-- fichiers file1 et file2, fichiers d'objets tries par
-- ordre croissant, le resultat se trouve dans resultat
-- ces trois fichiers doivent exister,
-- en cas d'egalite des valeurs, seule une seule est ecrite

```

```

-- si les fichiers n'existent pas, l'exception NAME_ERROR
-- est levee
-- il est suppose que les fichiers ne contiennent pas de
-- valeurs multiples

nb_fichier : constant integer := 2 ; -- nbre de fichiers en entree

subtype num_fichier is integer range 1..nb_fichier;
-- les numeros de fichiers en cours de manipulation

sortie : file_type ;
f: array(num_fichier)of file_type;
-- f : les deux fichiers en entree
-- prevoir une extension a k fichiers plus tard

n: array(num_fichier) of T_Objet ; -- les objets lus
-- orginaires des fichiers i

attente: array(num_fichier)of boolean; -- indiquent si des objets lus du
-- fichier i sont encore en attente
-- d'ecriture

-----

procedure recopie(f: in out file_type) is
-- =====
-- recopie le reste du fichier f dans le fichier sortie;
-- ferme f et sortie

x : T_Objet ;

begin
loop
lecture(f,x); ecriture(sortie,x);
end loop;
exception
when end_error => close(f);close(sortie);

end recopie;
-----

procedure entrer (i:in num_fichier) is
-- =====
-- realise la lecture d'un element de fi dans ni
-- et positionne attente(i).
begin
lecture (f(i),n(i)); attente(i):= true;
end entrer;
-----

procedure sortir (i: in num_fichier) is
-- =====
-- ecrit ni dans sortie, indique qu'il n'est
-- plus en attente
begin
ecriture (sortie, n(i)); attente(i):= false;
end sortir;
-----

begin
----- corps de la procedure fusion -----

```

```

open(f(1),in_file,fusion1);
open(f(2),in_file,fusion2);
open(sortie,out_file,resultat); -- on suppose ici que resultat existe!

    attente := (others => false); -- rien en attente au debut

    entrer(1);
    entrer(2);

loop          -- a ce stade n1 et n2 sont lu, il reste a determiner qui sort

    if n(1) > n(2) then sortir(2);
                    entrer(2);
    elsif not egal( n(1), n(2)) then sortir(1);
                    entrer(1);
    else          -- les deux valeurs sont egales
        sortir(1);
        attente(2):=false; -- n(2) doit aussi etre considere comme
                            -- sorti ceci au cas ou la lecture de f1
                            -- est deroutee sur une exception
        entrer(1);entrer(2);
    end if;
end loop;

exception
    when END_ERROR =>
        if      attente(1) then ecriture(sortie, n(1));
        elsif  attente(2) then ecriture(sortie, n(2));
        end if; -- au plus un seul est en attente de sortie
        -- comme au moins un des fichiers est termine, il suffit
        -- de recopier celui qui reste
        if end_of_file(f(1)) then close(f(1)); recopie(f(2));
        else                      close(f(2)); recopie(f(1));
        end if;

end fusion;

end P_FUSION;
*****
*****fichier test_fusion.adb*****
with Text_Io, Integer_Locio;
use Text_Io, Integer_Locio;
with P_Fusion;

procedure Test_Fusion is

procedure Mon_Get(F: File_Type; N: out Integer) is
begin
    Get(F,N);
end Mon_Get;

procedure Mon_Put(F: File_Type; N: Integer) is
begin
    Put(F,N);
end Mon_Put;

package Fusion_Entier is new P_Fusion (Integer,
                                        ">",
                                        "=",
                                        Mon_Get,
                                        Mon_Put);

use Fusion_Entier;

```

```

-----
procedure lect_impr(nom_fichier: string) is
--      =====
--          -- lecture et impression d'un fichier d'entier
--          -- a raison de max_ligne par ligne

n :integer;
compte:integer:= 0;      -- compteur d'impression
max_ligne : constant integer := 10; -- nbre max d'entier par ligne
f: file_type;

begin

    open(f,in_file,nom_fichier);
    loop
        get(f,n); put(n,width=>8);
        compte:=compte+1;
            if compte=max_ligne then new_line;
                compte := 0;
            end if;
        end loop;
exception
    when END_ERROR =>
        new_line;new_line;
        close(f);
end lect_impr;

-----
function Chaîne_Ent(I: Integer) return String is

    Ch : String := Integer'Image(I);
begin
    return Ch(2..Ch'Last);
end Chaîne_Ent;

-----
F : File_Type;
N : Integer;

begin
for I in 1..3 loop

    New_Line;
    Put ("***** test "); Put(I); Put ( " *****");
    New_Line;

    Put("fichier no 1 : "); Put("fusion_data1."& Chaîne_Ent(I));New_Line;
    Lect_Impr ("fusion_data1."& Chaîne_Ent(I));

    New_Line;
    Put_Line("fichier no 2");
    Lect_Impr ("fusion_data2."& chaîne_ent(I));

    Fusion ("fusion_data1."& chaîne_ent(I),
            "fusion_data2."& chaîne_ent(I),
            "fusion_res"& chaîne_ent(I));

    New_Line;
    Put_Line( " et voila le travail");
    Lect_Impr("fusion_res"& chaîne_ent(I));
end loop;
end Test_Fusion;

```