Structure de données

```
PACKAGE Outils IS

   SUBTYPE T_Mot IS String(1..30);
   TYPE T_Semaine IS(Lundi, Mardi, Mercredi, Jeudi, Vendredi,
Samedi, Dimanche);

   PROCEDURE Ctrl_Entier(Int : IN OUT Integer);
   PROCEDURE Ctrl_String(Chaine : IN OUT T_Mot);
   PROCEDURE Jour_Suiv(Jour : IN OUT T_Semaine);

END Outils;
```

```
WITH Outils;
USE Outils;

PACKAGE Gestion_Baby_Sitter IS

   TYPE T_Creneau IS(Matin,Aprem,Soir);
   TYPE T_Jour_Ouvre
IS(Lundi,Mardi,Mercredi,Jeudi,Vendredi,Samedi);

   TYPE T_Planning IS ARRAY(T_Jour_Ouvre,T_Creneau) OF T_Mot;

   TYPE T_Baby_Sitter IS RECORD
      Nom,Prenom : T_Mot := (OTHERS => ' ');
      Age : Positive; -- >= 16
      Quitte : Boolean := False;
      Nb_Creneaux : Integer := 0;
      Facture_Veille : Integer := 0;
      Facture_Semaine : Integer := 0;
      P1,P2 : T_Planning; -- := (OTHERS => (OTHERS => " "))
   END RECORD;

   TYPE T_Cell;
   TYPE T_Pteur IS ACCESS T_Cell;
   TYPE T_Cell IS RECORD
      Bs : T_Baby_Sitter;
      Suiv : T_Pteur;
   END RECORD;


   PROCEDURE Reset_Planning(Planning : IN OUT T_Planning);
   PROCEDURE Ajout_Bs(Bs : OUT T_Baby_Sitter; Liste : IN OUT
T_Pteur);
   PROCEDURE Demande_Depart_Bs(Liste : IN T_Pteur);
   PROCEDURE Supp_Bs(Bs : IN T_Baby_Sitter; Liste : IN OUT
T_Pteur);
   PROCEDURE Visu_Bs(Liste : IN T_Pteur);
```

```
   PROCEDURE Visu_Facture_Veille(Liste : IN T_Pteur);
   PROCEDURE Visu_Facture_Bs(Liste : IN T_Pteur);
   PROCEDURE Visu_P1(Liste : IN T_Pteur);
   PROCEDURE Visu_P2(Liste : IN T_Pteur);
   PROCEDURE Visu_Planning_Bs(Liste : IN T_Pteur);
   PROCEDURE Visu_Garde_Jour(Liste : IN T_Pteur);
   PROCEDURE Maj_Planning(Liste : IN T_Pteur);
   PROCEDURE Tri(Liste : IN T_Pteur);


END Gestion_Baby_Sitter;
```

```
WITH Outils, Gestion_Baby_Sitter;
USE Outils, Gestion_Baby_Sitter;

PACKAGE Gestion_Famille IS

   TYPE T_Age_Enf IS ARRAY(Integer RANGE 1..8) OF Integer RANGE -
1..12;

   TYPE T_Famille IS RECORD
      Nom : T_Mot := (OTHERS => ' ');
      Nb_Enfants : Integer RANGE 1..8;
      Age_Enfants : T_Age_Enf := (OTHERS => -1);
      Facture : Integer := 0;
      Last_Bs : T_Pteur := NULL;
   END RECORD;

   -- ABR trié par nom
   TYPE T_Noeud;
   TYPE T_Arbre IS ACCESS T_Noeud;
   TYPE T_Noeud IS RECORD
      Famille : T_Famille;
      Fg,Fd : T_Arbre;
   END RECORD;

   PROCEDURE New_Fam(Famille : OUT T_Famille; K : OUT Integer);
   PROCEDURE Insert_Fam(Famille : IN OUT T_Famille; K : Integer; A
: IN OUT T_Arbre);
   PROCEDURE Visu_Fam(A : IN T_Arbre);
   PROCEDURE Visu_Facture_Fam(A : IN T_Arbre);


END Gestion_Famille;
```

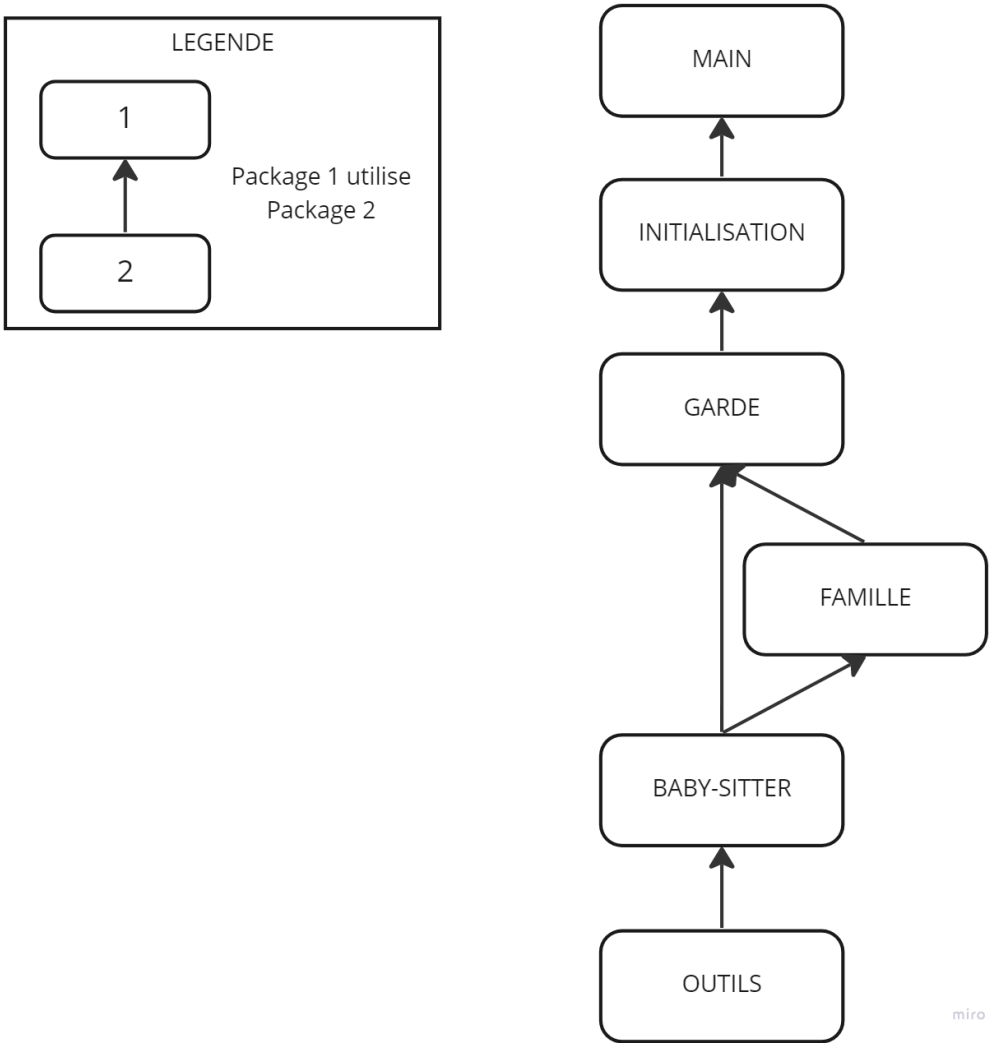
```
WITH Outils, Gestion_Baby_Sitter, Gestion_Famille;
USE Outils, Gestion_Baby_Sitter, Gestion_Famille;
```

```
PACKAGE Gestion_Garde IS

   PROCEDURE Demande_Garde(Famille : IN T_Famille; Liste :
T_Pteur);
   PROCEDURE Annule_Garde(Famille : IN T_Famille; Jour : IN
T_Semaine; Semaine : IN Integer; Liste : IN T_Pteur);
   PROCEDURE Visu_Garde_Fam(Famille : IN T_Famille; Liste : IN
T_Pteur);
   PROCEDURE Supp_Fam(Famille : IN T_Famille; Liste : IN T_Pteur;
Arbre : T_Arbre);

END Gestion_Garde;
```

Architecture des packages



Interface

```
WITH Ada.Text_IO,Ada.Integer_Text_IO,Ada.Float_Text_IO;
USE Ada.Text_IO,Ada.Integer_Text_IO,Ada.Float_Text_IO;
```

```
PROCEDURE Principal IS
   Mon_Choix, Visual, Bab, Fam, Ins, Sauv: Integer;


BEGIN

   Put_Line (" Le projet Ada : DONG Laurene et DEWITTE Lisa");
New_Line;

   LOOP
      Put_Line ("  Nous sommes le -nom du jour-"); New_Line;
      Put_Line ("  -- Affichage les factures de la veille --");
New_Line;
      Put_Line ("  ----- Menu principal -----");
      Put_Line (" 1 : Visualisation des donnees");
      Put_Line (" 2 : Gestion des baby-sitters");
      Put_Line (" 3 : Gestion des familles");
      Put_Line (" 4 : Gestion de garde");
      Put_Line (" 5 : Sauvegarde et restauration");
--       Put_Line ("6 : ");
--       Put_Line ("7 : ");
      Put_Line ("0 : Sortir");
      New_Line;

      Put_Line (" Saisissez votre choix."); New_Line; New_Line;

      LOOP
         BEGIN
            Get(Mon_Choix); Skip_Line; EXIT;
            EXCEPTION
            WHEN Data_Error => Skip_Line; Put_Line (" Erreur de
saisie. Recommencez");
         END;
      END LOOP;
      New_Line;
      New_Line;

   -------------------------------- Case MENU PRINCIPAL
-------------------------------------

   CASE Mon_Choix IS

   -------------------------------- Visualisation des donnees
--------------------
      WHEN 1=> -- Visualisation des donnees

            LOOP
               Put_Line ("  ----- Visualisation des donnees
-----"); New_Line;
               Put_Line (" 1 : Liste de tous les baby-sitters");
               Put_Line (" 2 : Liste de tous les familles");
```

```
                Put_Line (" 3 : Planning des baby-sitters de cette
semaine");
                Put_Line (" 4 : Planning des baby-sitters de la
semaine prochaine");
                Put_Line (" 5 : Planning de garde de cette
semaine");
                Put_Line (" 6 : Planning de garde de la semaine
prochaine");
                Put_Line (" 7 : Visualisation des factures");
                Put_Line (" 0 : Sortir");
                New_Line;

                Put_Line (" Saisissez votre choix."); New_Line;
New_Line;

                LOOP
                   BEGIN
                      Get(Visual); Skip_Line; EXIT;
                      EXCEPTION
                      WHEN Data_Error => Skip_Line; Put_Line ("
Erreur de saisie. Recommencez");
                   END;
                END LOOP;

                CASE Visual IS
                   WHEN 1 => -- Liste de tous les baby-sitters
                      Put_line (" Voici la liste des baby-
sitters:"); New_line;
                      NULL; New_Line; New_Line;
                   WHEN 2 => -- Liste de tous les familles
                      Put_line (" Voici la liste des familles");
New_line;
                      NULL; New_Line; New_Line;
                   WHEN 3 => -- Planning des baby-sitters de cette
semaine
                      Put_line (" Voici le planning de chaque
babysitters cette semaine:"); New_line;
                      NULL; New_Line; New_Line;
                   WHEN 4 => -- Planning des baby-sitters de la
semaine a venir
                      Put_line (" Voici le planning de chaque
babysitters la semaine prochaine:"); New_line;
                      NULL; New_Line; New_Line;
                   WHEN 5 => -- Planning de garde de cette semaine
                      Put_line (" Voici le planning de garde de
cette semaine:"); New_line;
                      NULL; New_Line; New_Line;
                   WHEN 6 => -- Planning de garde de la semaine a
venir
                      Put_line (" Voici le planning de garde de la
semaine prochaine:"); New_line;
                      NULL; New_Line; New_Line;
                   WHEN 7 => -- Visualisation des factures
```

```
                        Put_line (" Voici le montant des factures par
baby-sitter:"); New_line;
                        NULL; New_Line; New_Line;
                        Put_line (" Voici le montant des factures par
famille:"); New_line;
                        NULL; New_Line; New_Line;
                     WHEN 0 => EXIT;
                  WHEN OTHERS => NULL;
                  END CASE;
               END LOOP;
               New_Line;
               New_Line;

   ---------------------------------- Gestion des baby-sitters
----------------------------------

      WHEN 2 => -- Gestion des baby-sitters

               LOOP
                  Put_Line ("  ----- Gestion des baby-sitters
-----"); New_Line;
                  Put_Line (" 1 : Inscription de baby-sitters");
                  Put_Line (" 2 : Demande de depart");
                  Put_Line (" 0 : Sortir");
                  New_Line;

                  Put_Line (" Saisissez votre choix."); New_Line;
New_Line;

                  LOOP
                     BEGIN
                        Get(Bab); Skip_Line; EXIT;
                        EXCEPTION
                        WHEN Data_Error => Skip_Line; Put_Line ("
Erreur de saisie. Recommencez");
                     END;
                  END LOOP;

                  CASE Bab IS
                     WHEN 1 => -- Inscription de baby-sitters
                        NULL; New_Line; New_Line;
                     WHEN 2 => -- Demande de départ
                        NULL; New_Line; New_Line;
                     WHEN 0 => EXIT;
                  WHEN OTHERS => NULL;
                  END CASE;
               END LOOP;
               New_Line;
               New_Line;

   ---------------------------------- Gestion des familles
----------------------------------
```

```
      WHEN 3 => -- Gestion des familles

            LOOP
               Put_Line ("  ----- Gestion des familles -----");
New_Line;
               Put_Line (" 1 : Inscription des familles");
               Put_Line (" 2 : Desinscription des familles");
               Put_Line (" 0 : Sortir");
               New_Line;

               Put_Line (" Saisissez votre choix."); New_Line;
New_Line;

               LOOP
                  BEGIN
                     Get(Fam); Skip_Line; EXIT;
                     EXCEPTION
                     WHEN Data_Error => Skip_Line; Put_Line ("
Erreur de saisie. Recommencez");
                  END;
               END LOOP;

               CASE Fam IS
                  WHEN 1 => -- Inscription des familles
                     NULL; New_Line; New_Line;
                  WHEN 2 => -- Desinscription des familles
                     NULL;
                     New_Line; New_Line;
                  WHEN 0 => EXIT;
               WHEN OTHERS => NULL;
               END CASE;
            END LOOP;
            New_Line;
            New_Line;

   ---------------------------------- Inscription pour la garde
--------------------------------

      WHEN 4 => NULL;-- Inscription pour la garde

            LOOP
               Put_Line ("  ----- Inscription pour la garde
-----"); New_Line;
               Put_Line (" 1 : Ajouter une garde");
               Put_Line (" 2 : Annuler une garde");
               Put_Line (" 0 : Sortir");
               New_Line;

               Put_Line (" Saisissez votre choix."); New_Line;
New_Line;

               LOOP
                  BEGIN
```

```
                    Get(Ins); Skip_Line; EXIT;
                    EXCEPTION
                    WHEN Data_Error => Skip_Line; Put_Line ("
Erreur de saisie. Recommencez");
                END;
            END LOOP;

            CASE Ins IS
                WHEN 1 => -- Ajouter garde
                    NULL;
                    New_Line; New_Line;
                WHEN 2 => -- Annuler une garde
                    NULL;
                    New_Line; New_Line;
                WHEN 0 => EXIT;
            WHEN OTHERS => NULL;
            END CASE;
        END LOOP;
        New_Line;
        New_Line;

   ------------------------------- Sauvegarde et Restauration
-----------------------------------------

      WHEN 5 => -- Sauvegarde et restauration");
            LOOP
                Put_Line ("  ----- Statistiques -----"); New_Line;
                Put_Line (" 1 : Sauvegarder");
                Put_Line (" 2 : Restaurer");
                Put_Line (" 0 : Sortir");
                New_Line;

                Put_Line ("Saisissez votre choix."); New_Line;
New_Line;

            LOOP
                BEGIN
                    Get(Sauv); Skip_Line; EXIT;
                    EXCEPTION
                    WHEN Data_Error => Skip_Line; Put_Line ("
Erreur de saisie. Recommencez");
                END;
            END LOOP;

            CASE Sauv IS
                WHEN 1 => -- Sauvegarder
                    NULL;
                    New_Line; New_Line;
                WHEN 2 => -- Restaurer
                    NULL;
                    New_Line; New_Line;
                WHEN 0 => EXIT;
            WHEN OTHERS => NULL;
```

```
               END CASE;
            END LOOP;
            New_Line;
            New_Line;

            ---------------------------------- Retour menu
principal ----------------------------------------

      WHEN 0 => EXIT; -- Sortir
      WHEN OTHERS => NULL;

   END CASE;

            END LOOP;
               Put_Line (" Merci de votre visite!!!");
END Principal;



--CASE Mon_Choix IS
--WHEN 1 =>
--WHEN 2 =>
--WHEN 3 =>
--WHEN 4 =>
--WHEN 0 => EXIT;
--WHEN OTHERS => NULL;
```