



PACKAGE Outils IS

```

    SUBTYPE T_Mot IS String(1..40);
    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



```

SUBTYPE T_Range_Age IS Integer RANGE 0..12;
    TYPE T_Creneau IS (Matin, Apres_m, Soir);

    TYPE T_Planning IS ARRAY(1..6, T_Creneau) OF T_Mot; -- 2
plannings

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

    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;
        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 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_Planning(Liste : IN T_Pteur);
    PROCEDURE Supp_Planning(Planning : IN OUT T_Planning);
    PROCEDURE Maj_Planning(Liste : IN T_Pteur);

```





```
PROCEDURE Tri_Bulle(Liste : IN OUT T_Pteur);
```

```
END Gestion_Baby_Sitter;
```

```
WITH Ada.Integer_Text_IO, Ada.Text_IO, Outils;
```

```
USE Ada.Integer_Text_IO, Ada.Text_IO, Outils;
```

```
PACKAGE Gestion_Famille IS
```

```
  TYPE T_Age_Enf IS ARRAY(Integer RANGE 1..8) OF Integer RANGE 0..12;
```

```
  -- TYPE T_Facture IS ARRAY(Integer RANGE 1..Taille_Arbre) OF Integer;
```

```
  TYPE T_Famille IS RECORD
```

```
    Nom : T_Mot;
```

```
    Nb_Enfants : Integer RANGE 1..8;
```

```
    Age : T_Age_Enf;
```

```
  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 Ajout_Fam(Famille : OUT T_Famille; Arbre : IN OUT T_Arbre);
```

```
  FUNCTION Taille_Arbre(Arbre : T_Arbre) RETURN Integer;
```

```
  PROCEDURE Visu_Fam(Arbre : IN T_Arbre);
```

```
  PROCEDURE Visu_Facture(Facture : IN T_Facture);
```

```
END Gestion_Famille;
```

```
WITH Ada.Integer_Text_IO, Ada.Text_IO, Outils,
```

```
Gestion_Baby_Sitter, Gestion_Famille;
```

```
USE Ada.Integer_Text_IO, Ada.Text_IO, 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(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;
```
