

Manuel d'utilisation

Table des matières

Lancement du menu de commandes.....2

Lancement du tests..... 3

Package Genealogic_Tree.....4

 T_Person..... 4

 T_Genealogic_Tree..... 5

 T_Genealogic_Tree: sous-type de Binary_Tree..... 5

Lancement du menu de commandes

Se placer dans le répertoire du menu sous `src/menu/`

```
(hugo@n7) - [~/projects/projet_prog]  
$ cd src/menu
```

S'assurer que le script peut être exécuter

```
(hugo@n7) - [~/projects/projet_prog/src/menu]  
$ chmod +x ./run_menu.sh
```

Lancer le script de menu

```
(hugo@n7) - [~/projects/projet_prog/src/menu]  
$ sh ./run_menu.sh
```

Le script s'occupe de copier les fichiers nécessaires dans le dossier courant, de compiler avec les bonnes options, de lancer le menu et une fois que le programme terminé, de nettoyer le dossier courant.

Lancement du tests

Se placer dans le répertoire des tests sous `src/tests/`

```
(hugo@n7) - [~/projects/projet_prog]  
$ cd src/tests
```

S'assurer que le script peut être exécuter

```
(hugo@n7) - [~/projects/projet_prog/src/tests]  
$ chmod +x ./run_tests.sh
```

Lancer le script de tests

```
(hugo@n7) - [~/projects/projet_prog/src/tests]  
$ sh ./run_tests.sh
```

Le script s'occupe de copier les fichiers nécessaires dans le dossier courant, de compiler avec les bonnes options, de lancer les tests, d'afficher l'ensemble des erreurs de compilation ou de runtime, de confirmer si les tests sont passés et de nettoyer le dossier courant.

Package Genealogic_Tree

T_Person

Un T_Person représente une personne identifiée par un ID unique. Il est défini comme un type access pointant vers une structure contenant cet ID. Plusieurs opérations sont possibles sur un T_Person, notamment l'initialisation avec un ID, la récupération de cet ID, la vérification si la personne est valide et l'affichage de ses informations. Une liste (T_List_Person) permet de stocker plusieurs personnes.

```
-- ### Constructor ###  
  
-- Initializes a person  
procedure Init(Person: in out T_Person; ID: in Natural) with  
| Post => GetID(Person) = ID;  
  
-- ### Getters / Setters ###  
  
-- Returns the id of the person  
function GetID (Person: in T_Person) return Natural;  
  
-- Checks if the person is null  
function Is_Null(Person : T_Person) return Boolean;  
  
-- Prints the person  
procedure PrintPerson(Person: in T_Person);
```

T_Genealogic_Tree

Un T_Genealogic_Tree est une structure d'arbre binaire représentant un arbre généalogique. Chaque nœud contient une personne (T_Person) et peut avoir deux noeuds référencés droit et gauche représentant les parents (père à gauche, mère à droite). Différentes opérations permettent d'ajouter un parent, de supprimer une personne et ses ancêtres, de rechercher une personne dans l'arbre et d'obtenir diverses informations, comme le nombre d'ancêtres ou la liste des personnes sans parents. Il est également possible d'afficher l'arbre et d'extraire des listes de personnes selon certains critères.

```

-- ### Fonctions ###

-- Add parent with a struct
procedure AddParentByPerson(Tree: in out T_Genealogic_Tree; Person: in T_Person; IsFemale: in Boolean) with
  Post => ((IsFemale = False) and (IsBranchEmpty(Tree, True) = False)) or
  (IsFemale and (IsBranchEmpty(Tree, False) = False)); -- Si on ajoute une mere, on vérifie l'ID de la personne de droite et si on ajoute un pere, l'
ID de gauche

-- Add parent with an id
procedure AddParentById(Tree: in out T_Genealogic_Tree; ID_Person: in Natural; IsFemale: in Boolean) with
  Post => ((IsFemale = False) and (IsBranchEmpty(Tree, True) = False)) or
  (IsFemale and (IsBranchEmpty(Tree, False) = False));

-- Delete person and its ancestors from the tree
procedure DeletePerson(Tree: in out T_Genealogic_Tree; Person: in out T_Person) with
  Pre => (FindInTree(Tree, GetID(Person)) = True);

-- Returns the number of ancestors of the person, him included
function NumberAncestors(Tree: in T_Genealogic_Tree; Person: in T_Person) return Natural with
  Pre => (FindInTree(Tree, GetID(Person)) = True), -- La personne renseignée doit être dans l'arbre fourni
  Post => (NumberAncestors'Result >= 1); -- Le nombre d'ancêtres doit être de 1 minimum car la personne renseignée est compris dans les ancêtres

-- Returns all the ID of the ancestors of the person
function AncestorsGen(Tree: in T_Genealogic_Tree; Generation: in Natural) return T_List_Person;

-- Print the tree
procedure PrintTree(Tree: in T_Genealogic_Tree);

-- Returns the persons who don't have parents
function PersonsWithXParents(Tree: in T_Genealogic_Tree; NumberParent: in Natural) return T_List_Person with
  Pre => ((NumberParent >= 0) and (NumberParent <= 2));

-- Determines if the person is in tree (returns boolean)
function FindInTree(Tree: in T_Genealogic_Tree; ID: Natural) return Boolean;

-- Determines if the person is in tree (returns node)
function FindPersonInTree(Tree: in T_Genealogic_Tree; ID: Natural) return T_Genealogic_Tree;

```

T_Genealogic_Tree: sous-type de Binary_Tree

Binary_Tree est une structure générique implémentant un arbre binaire, où chaque nœud contient une donnée de type T_Data et possède potentiellement un sous-arbre gauche et un sous-arbre droit.

```

procedure Init(Tree: in out T_Genealogic_Tree; Data: in T_Person) renames Genealogic_Tree_Of_Persons.Init;
function GetLeft (Tree: T_Genealogic_Tree) return T_Genealogic_Tree renames Genealogic_Tree_Of_Persons.GetLeft;
procedure SetLeft (Tree: in out T_Genealogic_Tree ; Data: in T_Person) renames Genealogic_Tree_Of_Persons.SetLeft;
procedure ClearLeft (Tree: in out T_Genealogic_Tree) renames Genealogic_Tree_Of_Persons.ClearLeft;
function GetRight (Tree: T_Genealogic_Tree) return T_Genealogic_Tree renames Genealogic_Tree_Of_Persons.GetRight;
procedure SetRight (Tree: in out T_Genealogic_Tree; Data: in T_Person) renames Genealogic_Tree_Of_Persons.SetRight;
procedure ClearRight (Tree: in out T_Genealogic_Tree) renames Genealogic_Tree_Of_Persons.ClearRight;
function GetData (Tree: in T_Genealogic_Tree) return T_Person renames Genealogic_Tree_Of_Persons.GetData;
function IsBranchEmpty (Tree: in T_Genealogic_Tree ; left: in Boolean) return Boolean renames Genealogic_Tree_Of_Persons.IsBranchEmpty;
function IsLeaf (Tree: in T_Genealogic_Tree) return Boolean renames Genealogic_Tree_Of_Persons.IsLeaf;
function IsEmpty (Tree: in T_Genealogic_Tree) return Boolean renames Genealogic_Tree_Of_Persons.IsEmpty;
function GetSize (Tree: in T_Genealogic_Tree) return Natural renames Genealogic_Tree_Of_Persons.GetSize;
procedure Delete (Tree: in out T_Genealogic_Tree) renames Genealogic_Tree_Of_Persons.Delete;

```