Ensemble

Généré par Doxygen 1.7.6.1

Mercredi Novembre 28 2012 20 :43 :45

Table des matières

Chapitre 1

Projet SD : Ensemble

Cette bibliothèque s'attardera sur les ensembles (non ordonnés).

Vous retrouverez les fonctions essentielles au développement d'un ensemble ainsi que les opérations ensemblistes (tel que l'union ou l'intersection) sur deux ensembles.

Remarque : Notre code reprend le travail de LAMEIRA Yannick et VERDIER Frédéric concernant les types de base. La documentation de leur code source ne sera pas généré ici. Je vous invite donc à vous rediriger vers leur documentation pour de plus amples renseignements concernant ces types de base.

Auteur

CARBONNEL Jessie, LY Julie

Chapitre 2

Index des structures de données

2.1 Structures de données

Liste des structures de données avec une brève description :

Chapitre 3

Index des fichiers

3.1 Liste des fichiers

Liste de tous les fichiers documentés avec une brève description :

Chapitre 4

Documentation des structures de données

4.1 Référence de la structure Element

Struture d'un élement Cette struture permet de créer un élement.

`#include <ensemble.h>`

Champs de données

– Type **elt**
– Element **suivant**

4.1.1 Description détaillée

Struture d'un élement Cette struture permet de créer un élement.

La documentation de cette structure a été générée à partir du fichier suivant :
– ensemble/ensemble.h

4.2 Référence de la structure Ensemble

Struture de Type Ensemble Cette struture permet la création du type 'Ensemble'. Ce type sera appelé par un int égal à 5.

`#include <ensemble.h>`

4.2.1 Description détaillée

Struture de Type Ensemble Cette struture permet la création du type 'Ensemble'. Ce type sera appelé par un int égal à 5.

La documentation de cette structure a été générée à partir du fichier suivant :
– ensemble/ensemble.h

4.3 Référence de la structure ensemble

Struture d'un ensemble Cette struture permet de créer un ensemble.

`#include <ensemble.h>`

Champs de données

– Element **premier**

4.3.1 Description détaillée

Struture d'un ensemble Cette struture permet de créer un ensemble.

La documentation de cette structure a été générée à partir du fichier suivant :
– ensemble/ensemble.h

Chapitre 5

Documentation des fichiers

5.1 Référence du fichier ensemble/ensemble.h

: Déclaration des fonctions sur les Ensembles

```
#include "typesBase/typeBase.h"
```

Structures de données

- struct Element
  Struture d'un élement Cette struture permet de créer un élement.
- struct ensemble
  Struture d'un ensemble Cette struture permet de créer un ensemble.

Définitions de type

- typedef struct Type ∗ **Ensemble**
- typedef struct Element ∗ **Element**
- typedef struct ensemble ∗ **ensemble**

Fonctions

- Ensemble ensemble_creer ()
  Création d'un ensemble vide.
- void ensemble_detruire (Ensemble e)
  Detruire un ensemble.
- Ensemble ensemble_cloner (Ensemble e)
  Cloner un ensemble.
- void ensemble_afficher (Ensemble e)
  Afficher les éléments d'un ensemble.
- int element_present (Ensemble e, Type t)
  Indique si un élément est présent dans un ensemble.
- void element_ajouter (Ensemble e, Type t)
  Ajouter un élément dans un ensemble.
- void element_supprimer (Ensemble e, Type t)
  Supprimer un élément dans un ensemble.

- int nombre_elements (Ensemble e)
  Indique le nombre d'éléments d'un ensemble.
- int ensembles_egaux (Ensemble e1, Ensemble e2)
  Indique si deux ensembles sont égaux.
- int ensemble_meme_cardinalite (Ensemble e1, Ensemble e2)
  Indique si deux ensembles ont la même cardinalité.
- Ensemble ensemble_union (Ensemble e1, Ensemble e2)
  L'union de deux ensembles.
- Ensemble ensemble_intersection (Ensemble e1, Ensemble e2)
  L'intersection de deux ensembles.
- Ensemble ensemble_difference (Ensemble e1, Ensemble e2)
  La différence de deux ensembles.
- Ensemble ensemble_difference_symetrique (Ensemble e1, Ensemble e2)
  La différence symétrique de deux ensembles.

5.1.1 Description détaillée

: Déclaration des fonctions sur les Ensembles

Auteur

CARBONNEL Jessie, LY Julie

Version

0.4

Date

10 octobre 2012

5.1.2 Documentation des fonctions

5.1.2.1 void element_ajouter (Ensemble *e*, Type *t*)

Ajouter un élément dans un ensemble.

Paramètres

e	L'ensemble dans lequel on veut ajouter un élément.
t	Un type quelconque qui représente l'élément à ajouter dans l'ensemble.

Renvoie

Ne renvoie rien.

Permet d'ajouter un élément dans l'ensemble passé en paramètre à la condition qu'il ne s'y trouve pas déjà. Possibilité d'ajouter un ensemble dans un ensemble.

5.1.2.2 int element_present (Ensemble *e*, Type *t*)

Indique si un élément est présent dans un ensemble.

Paramètres

e	L'ensemble dans lequel on veut déterminer si un élément est présent ou non.
t	Un type quelconque qui représente l'élément dont on déclare la présence ou non présence dans l'ensemble.

Renvoie

Renvoie 1 si l'élément est présent, 0 sinon.

5.1.2.3 void element_supprimer (Ensemble *e*, Type *t*)

Supprimer un élément dans un ensemble.

Paramètres

e	Un ensemble dans lequel on veut supprimer un élément.
t	Un type quelconque qui représente l'élément à supprimer.

Renvoie

Ne renvoie rien.

5.1.2.4 void ensemble_afficher (Ensemble *e*)

Afficher les éléments d'un ensemble.

Paramètres

e	L'ensemble dont on veut afficher tous les éléments.

Renvoie

Ne renvoie rien.

5.1.2.5 Ensemble ensemble_cloner (Ensemble *e*)

Cloner un ensemble.

Paramètres

e	L'ensemble à cloner.

Renvoie

Renvoie un ensemble.

5.1.2.6 Ensemble ensemble_creer ( )

Création d'un ensemble vide.

Renvoie

Renvoie un ensemble vide.

5.1.2.7 void ensemble_detruire (Ensemble *e*)

Detruire un ensemble.

Paramètres

e	L'ensemble à détruire.

Renvoie

Ne renvoie rien.

Permet de détruire un ensemble correctement (sans fuite mémoire).

5.1.2.8 Ensemble ensemble_difference (Ensemble *e1*, Ensemble *e2*)

La différence de deux ensembles.

Paramètres

e1	L'ensemble qui subira la différence avec e2.
e2	Un ensemble.

Renvoie

Renvoie un ensemble étant la différence des deux ensembles passés en paramètre.

Permet de créer un ensemble e3 qui contiendra tous les éléments de e1 privés des éléments de e2.

5.1.2.9 Ensemble ensemble_difference_symetrique (Ensemble *e1*, Ensemble *e2*)

La différence symétrique de deux ensembles.

Paramètres

e1	Un ensemble 1.
e2	Un ensemble 2.

Renvoie

Renvoie un ensemble étant la différence symetrique des deux ensembles passés en paramètre.

Permet de créer un ensemble e3 qui contiendra l'union de e1 et e2 privée de l'intersection de e1 et e2.

5.1.2.10 Ensemble ensemble_intersection (Ensemble *e1,* Ensemble *e2*)

L'intersection de deux ensembles.

Paramètres

e1	Un ensemble 1.
e2	Un ensemble 2.

Renvoie

Renvoie un ensemble étant l'intersection des deux ensembles passés en paramètre.

Permet de créer un ensemble e3 qui contiendra tous éléments tel que si x appartient à e1 ET e2, x n'appartient pas à e3. Les possibles occurences ne seront pas traitées.

5.1.2.11 int ensemble_meme_cardinalite (Ensemble *e1,* Ensemble *e2*)

Indique si deux ensembles ont la même cardinalité.

Paramètres

e1	L'ensemble à comparer avec e2.
e2	L'ensemble à comparer avec e1.

Renvoie

Renvoie 1 si les deux ensembles ont la même cardinalité, 0 sinon.

Permet de déterminer si deux ensembles possèdent le même nombre d'éléments.

5.1.2.12 Ensemble ensemble_union (Ensemble *e1,* Ensemble *e2*)

L'union de deux ensembles.

Paramètres

e1	Un ensemble 1.
e2	Un ensemble 2.

Renvoie

Renvoie un ensemble étant l'union des deux ensembles passés en paramètre.

Permet de créer un ensemble qui contiendra tous les éléments de l'ensemble e1 et tous les éléments de l'ensemble e2. Les possibles occurences ne seront pas traitées.

5.1.2.13 int ensembles_egaux (Ensemble *e1*, Ensemble *e2*)

Indique si deux ensembles sont égaux.

Paramètres

e1	L'ensemble à comparer avec e2.
e2	L'ensemble à comparer avec e1.

Renvoie

Renvoie 1 si les deux ensembles sont égaux, 0 sinon.

Permet de déterminer si deux ensembles sont égaux, c'est-à-dire s'ils possèdent les mêmes éléments. Remarque : Dans le cas d'un élément 'ensemble' dans e1, on test également si tous ses éléments sont égaux aux éléments de l'élément 'ensemble' de e2.

5.1.2.14 int nombre_elements (Ensemble *e*)

Indique le nombre d'éléments d'un ensemble.

Paramètres

e	L'ensemble dont on veut connaitre le nombre d'éléments.

Renvoie

Renvoie le nombre d'éléments de l'ensemble.

Remarque : Un élement 'ensemble' compte pour un seul élément.