

voyageurDeCommerce

1

Généré par Doxygen 1.7.1

Fri Dec 28 2012 21 :21 :21

Table des matières

1	Index des classes	1
1.1	Liste des classes	1
2	Index des fichiers	3
2.1	Liste des fichiers	3
3	Documentation des classes	5
3.1	Référence de la structure Distance	5
3.1.1	Description détaillée	5
3.2	Référence de la structure Errors	5
3.2.1	Description détaillée	6
3.3	Référence de la structure Instance	6
3.3.1	Description détaillée	6
3.4	Référence de la structure Tour	6
3.4.1	Description détaillée	6
3.5	Référence de la structure Town	7
3.5.1	Description détaillée	7
4	Documentation des fichiers	9
4.1	Référence du fichier /development/cours/L2/algo3/TP/voyageCommerce/lib/bruteForce.h .	9
4.1.1	Description détaillée	9
4.1.2	Documentation des fonctions	9
4.1.2.1	bruteForce_bestPath	9
4.2	Référence du fichier /development/cours/L2/algo3/TP/voyageCommerce/lib/distance.h . .	10
4.2.1	Description détaillée	10
4.2.2	Documentation des fonctions	10
4.2.2.1	distance_betweenTowns	10
4.2.2.2	distance_calculDistance	11
4.2.2.3	distance_new	11

4.3	Référence du fichier /development/cours/L2/algo3/TP/voyageCommerce/lib/errors.h . . .	11
4.3.1	Description détaillée	11
4.3.2	Documentation des fonctions	12
4.3.2.1	errors_displayErrorsMessage	12
4.3.2.2	errors_initialize	12
4.3.2.3	errors_setFileNotFound	12
4.3.2.4	errors_setLsrNotValidParameter	12
4.3.2.5	errors_setNbArguments	12
4.3.2.6	errors_setNoAlgoSpecified	12
4.3.2.7	errors_setTagFNotFound	12
4.4	Référence du fichier /development/cours/L2/algo3/TP/voyageCommerce/lib/instance.h . .	13
4.4.1	Description détaillée	13
4.4.2	Documentation des fonctions	13
4.4.2.1	instance_display	13
4.4.2.2	instance_displayLinearVector	14
4.4.2.3	instance_displayMatrix	14
4.4.2.4	instance_initialize	14
4.4.2.5	instance_initializeDistances	14
4.4.2.6	instance_push	14
4.5	Référence du fichier /development/cours/L2/algo3/TP/voyageCommerce/lib/parsing.h . . .	14
4.5.1	Description détaillée	15
4.5.2	Documentation du type de l'énumération	15
4.5.2.1	AlgoType	15
4.5.3	Documentation des fonctions	15
4.5.3.1	parsing_algoType	15
4.5.3.2	parsing_parseFileName	16
4.5.3.3	parsing_parseVerboseMode	16
4.6	Référence du fichier /development/cours/L2/algo3/TP/voyageCommerce/lib/tour.h	16
4.6.1	Description détaillée	17
4.6.2	Documentation des fonctions	17
4.6.2.1	tour_2opt	17
4.6.2.2	tour_calculLength	17
4.6.2.3	tour_display	17
4.6.2.4	tour_initialize	17
4.6.2.5	tour_nextPermutation	17
4.6.2.6	tour_randomWalk	18

4.7	Référence du fichier <code>/development/cours/L2/algo3/TP/voyageCommerce/lib/town.h</code>	18
4.7.1	Description détaillée	18
4.7.2	Documentation des fonctions	18
4.7.2.1	<code>town_initialize</code>	18
4.8	Référence du fichier <code>/development/cours/L2/algo3/TP/voyageCommerce/src/bruteForce.c</code> .	19
4.8.1	Description détaillée	19
4.8.2	Documentation des fonctions	21
4.8.2.1	<code>bruteForce_bestPath</code>	21

Chapitre 1

Index des classes

1.1 Liste des classes

Liste des classes, structures, unions et interfaces avec une brève description :

Distance (Objet des distances)	5
Errors (Objet des erreurs)	5
Instance (Objet des instances)	6
Tour (Objet d'une tournée)	6
Town (Objet des ville)	7

Chapitre 2

Index des fichiers

2.1 Liste des fichiers

Liste de tous les fichiers documentés avec une brève description :

/development/cours/L2/algo3/TP/voyageCommerce/lib/ bruteForce.h (Programme de tests) . . .	9
/development/cours/L2/algo3/TP/voyageCommerce/lib/ distance.h (Programme de tests)	10
/development/cours/L2/algo3/TP/voyageCommerce/lib/ errors.h (Programme de tests)	11
/development/cours/L2/algo3/TP/voyageCommerce/lib/ instance.h (Programme de tests)	13
/development/cours/L2/algo3/TP/voyageCommerce/lib/ parsing.h (Programme de tests)	14
/development/cours/L2/algo3/TP/voyageCommerce/lib/ tour.h (Programme de tests)	16
/development/cours/L2/algo3/TP/voyageCommerce/lib/ town.h (Programme de tests)	18
/development/cours/L2/algo3/TP/voyageCommerce/lib/ util.h	??
/development/cours/L2/algo3/TP/voyageCommerce/src/ bruteForce.c (Fonctions utiles)	19

Chapitre 3

Documentation des classes

3.1 Référence de la structure Distance

Objet des distances.

```
#include <distance.h>
```

Attributs publics

- [Town](#) firstTown
- [Town](#) secondTown
- double distance

3.1.1 Description détaillée

Objet des distances. [Town](#) bla bla bla est un petit objet de gestion de chaînes de caractères. La chaîne se termine obligatoirement par un zéro de fin et l'objet connait la taille de chaîne contient !

La documentation de cette structure a été générée à partir du fichier suivant :

- /development/cours/L2/algo3/TP/voyageCommerce/lib/[distance.h](#)

3.2 Référence de la structure Errors

Objet des erreurs.

```
#include <errors.h>
```

Attributs publics

- char * errorNbArguments
- char * errorTagNotFound
- char * errorFileNotFound
- char * errorNoAlgoSpecified
- char * errorLsrNotValidParameter
- int nbErrors

3.2.1 Description détaillée

Objet des erreurs. [Town](#) bla bla bla est un petit objet de gestion de chaînes de caractères. La chaîne se termine obligatoirement par un zéro de fin et l'objet connaît la taille de chaîne contient !

La documentation de cette structure a été générée à partir du fichier suivant :

- /development/cours/L2/algo3/TP/voyageCommerce/lib/[errors.h](#)

3.3 Référence de la structure Instance

Objet des instances.

```
#include <instance.h>
```

Attributs publics

- [Town](#) towns [N]
- [Distance](#) distances [N]
- int nbTowns
- char * name
- char * type

3.3.1 Description détaillée

Objet des instances. [Town](#) bla bla bla est un petit objet de gestion de chaînes de caractères. La chaîne se termine obligatoirement par un zéro de fin et l'objet connaît la taille de chaîne contient !

La documentation de cette structure a été générée à partir du fichier suivant :

- /development/cours/L2/algo3/TP/voyageCommerce/lib/[instance.h](#)

3.4 Référence de la structure Tour

Objet d'une tournée.

```
#include <tour.h>
```

Attributs publics

- [Town](#) towns [N]
- int nbTowns
- double length
- [Distance](#) * distances

3.4.1 Description détaillée

Objet d'une tournée. [Town](#) bla bla bla est un petit objet de gestion de chaînes de caractères. La chaîne se termine obligatoirement par un zéro de fin et l'objet connaît la taille de chaîne contient !

La documentation de cette structure a été générée à partir du fichier suivant :

- /development/cours/L2/algo3/TP/voyageCommerce/lib/[tour.h](#)

3.5 Référence de la structure Town

Objet des ville.

```
#include <town.h>
```

Attributs publics

- int **x**
- int **y**
test x

- int **id**
test y

3.5.1 Description détaillée

Objet des ville. **Town** bla bla bla est un petit objet de gestion de chaînes de caractères. La chaîne se termine obligatoirement par un zéro de fin et l'objet connaît la taille de chaîne contient !

La documentation de cette structure a été générée à partir du fichier suivant :

- /development/cours/L2/algo3/TP/voyageCommerce/lib/[town.h](#)

Chapitre 4

Documentation des fichiers

4.1 Référence du fichier /development/cours/L2/algo3/TP/voyageCommerce/lib/br

Programme de tests.

```
#include "instance.h"
#include "tour.h"
#include "util.h"
```

Fonctions

– [Tour bruteForce_bestPath](#) (Instance pInstance)

4.1.1 Description détaillée

Programme de tests.

Auteur

Antoine de Roquemaurel

Version

0.1

Date

27/12/2012 17 :58 :36

Programme de test pour l'objet de gestion des chaînes de caractères Str_t.

4.1.2 Documentation des fonctions

4.1.2.1 Tour bruteForce_bestPath (Instance *pInstance*)

Paramètres

pInstance

Renvoie

4.2 Référence du fichier /development/cours/L2/algo3/TP/voyageCommerce/lib/di

Programme de tests.

```
#include <math.h>
#include "town.h"
```

Classes

- struct [Distance](#)
Objet des distances.

Fonctions

- void [distance_new](#) ([Distance](#) *pDistance, [Town](#) *pFirstTown, [Town](#) *pSecondTown)
- double [distance_calculDistance](#) (const [Town](#) pTown1, const [Town](#) pTown2)
- double [distance_betweenTowns](#) ([Distance](#) *pDistances, int i, int j)

4.2.1 Description détaillée

Programme de tests.

Auteur

Antoine de Roquemaurel

Version

0.1

Date

01/12/2012 20 :33 :44

Programme de test pour l'objet de gestion des chaînes de caractères Str_t.

4.2.2 Documentation des fonctions

4.2.2.1 double distance_betweenTowns ([Distance](#) * *pDistances*, int *i*, int *j*)

Paramètres

pDistances

i

j

Renvoie

4.2.2.2 double distance_calculDistance (const Town *pTown1*, const Town *pTown2*)

Paramètres

pTown1

pTown2

Renvoie

4.2.2.3 void distance_new (Distance * *pDistance*, Town * *pFirstTown*, Town * *pSecondTown*)

Paramètres

pDistance

pFirstTown

pSecondTown

4.3 Référence du fichier /development/cours/L2/algo3/TP/voyageCommerce/lib/er

Programme de tests.

```
#include <stdio.h>
#include <stdlib.h>
```

Classes

- struct [Errors](#)
Objet des erreurs.

Fonctions

- void [errors_initialize](#) ([Errors](#) *pErrors)
- void [errors_displayErrorMessage](#) (const [Errors](#) pErrors)
- void [errors_setNbArguments](#) ([Errors](#) *pErrors)
- void [errors_setTagNotFound](#) ([Errors](#) *pErrors)
- void [errors_setFileNotFound](#) ([Errors](#) *pErrors, char *fileName)
- void [errors_setNoAlgoSpecified](#) ([Errors](#) *pErrors)
- void [errors_setLsrNotValidParameter](#) ([Errors](#) *pErrors)

4.3.1 Description détaillée

Programme de tests.

Auteur

Antoine de Roquemaurel

Version

0.1

Date

21/11/2012 17 :42 :37

Programme de test pour l'objet de gestion des chaînes de caractères Str_t.

4.3.2 Documentation des fonctions**4.3.2.1 void errors_displayErrorsMessage (const Errors *pErrors*)****Paramètres**

pErrors

4.3.2.2 void errors_initialize (Errors * *pErrors*)**Paramètres**

pErrors

4.3.2.3 void errors_setFileNotFound (Errors * *pErrors*, char * *fileName*)**Paramètres**

pErrors

fileName

4.3.2.4 void errors_setLsrNotValidParameter (Errors * *pErrors*)**Paramètres**

pErrors

4.3.2.5 void errors_setNbArguments (Errors * *pErrors*)**Paramètres**

pErrors

4.3.2.6 void errors_setNoAlgoSpecified (Errors * *pErrors*)**Paramètres**

pErrors

4.3.2.7 void errors_setTagFNotFound (Errors * *pErrors*)**Paramètres**

pErrors

4.4 Référence du fichier /development/cours/L2/algo3/TP/voyageCommerce/lib/instance.h

Programme de tests.

```
#include <stdbool.h>
#include <string.h>
#include "util.h"
#include "town.h"
#include "distance.h"
```

Classes

- struct [Instance](#)
Objet des instances.

Macros

- #define N 512

Fonctions

- void [instance_display](#) (const [Instance](#) pInstance)
- void [instance_initialize](#) ([Instance](#) *pInstance, FILE *pFile)
- void [instance_push](#) ([Instance](#) *pInstance, const [Town](#) pTown)
- void [instance_initializeDistances](#) ([Instance](#) *pInstance)
- void [instance_displayLinearVector](#) ([Instance](#) pInstance)
- void [instance_displayMatrix](#) ([Instance](#) pInstance)

4.4.1 Description détaillée

Programme de tests.

Auteur

Antoine de Roquemaurel

Version

0.1

Date

21/11/2012 22 :03 :34

Programme de test pour l'objet de gestion des chaînes de caractères Str_t.

4.4.2 Documentation des fonctions

4.4.2.1 void instance_display (const Instance pInstance)

Paramètres

pInstance

4.4.2.2 void instance_displayLinearVector (Instance *pInstance*)

Paramètres

pInstance

4.4.2.3 void instance_displayMatrix (Instance *pInstance*)

Paramètres

pInstance

4.4.2.4 void instance_initialize (Instance * *pInstance*, FILE * *pFile*)

Paramètres

pInstance

pFile

4.4.2.5 void instance_initializeDistances (Instance * *pInstance*)

Paramètres

pInstance

4.4.2.6 void instance_push (Instance * *pInstance*, const Town *pTown*)

Paramètres

pInstance

pTown

4.5 Référence du fichier /development/cours/L2/algo3/TP/voyageCommerce/lib/pa

Programme de tests.

```
#include <stdbool.h>
```

```
#include "util.h"
```

```
#include "errors.h"
```

Énumérations

```
- enum AlgoType { BRUTEFORCE, LOCALSEARCH_RANDOM, LOCALSEARCH_ -  
  SYSTEMATIC, GENETIC }  
  parsing.h
```

Fonctions

- bool [parsing_parseVerboseMode](#) (char **pTab, const int pSize)
- char * [parsing_parseFileName](#) (char **pTab, const int pSize, [Errors](#) *pErrors)
- [AlgoType parsing_algoType](#) (char **pTab, const int pSize, [Errors](#) *pErrors, int *param1AlgoType, int *param2AlgoType)

4.5.1 Description détaillée

Programme de tests.

Auteur

Antoine de Roquemaurel

Version

0.1

Date

21/11/2012 17 :17 :24

Programme de test pour l'objet de gestion des chaînes de caractères Str_t.

4.5.2 Documentation du type de l'énumération

4.5.2.1 enum AlgoType

[parsing.h](#)

Énumération des types d'algorithmes [Town](#) bla bla bla est un petit objet de gestion de chaînes de caractères. La chaîne se termine obligatoirement par un zéro de fin et l'objet connaît la taille de chaîne contient !

4.5.3 Documentation des fonctions

4.5.3.1 AlgoType parsing_algoType (char ** pTab, const int pSize, Errors * pErrors, int * param1AlgoType, int * param2AlgoType)

Paramètres

pTab

pSize

pErrors

param1AlgoType

param2AlgoType

Renvoie

TODO Erreur paramètre après -lsr non entier ou = 0

TODO Erreur paramètre après -lsnr non entier ou = 0

TODO Erreur paramètre après -lsnr non entier ou = 0

TODO Erreur paramètre après -lsnr non entier ou = 0

4.5.3.2 char* parsing_parseFileName (char ** *pTab*, const int *pSize*, Errors * *pErrors*)

Paramètres

pTab

pSize

pErrors

Renvoie

4.5.3.3 bool parsing_parseVerboseMode (char ** *pTab*, const int *pSize*)

Paramètres

pTab

pSize

Renvoie

4.6 Référence du fichier /development/cours/L2/algo3/TP/voyageCommerce/lib/to

Programme de tests.

```
#include <stdlib.h>
#include <stdbool.h>
#include <time.h>
#include "town.h"
#include "instance.h"
#include "distance.h"
```

Classes

- struct [Tour](#)
Objet d'une tournée.

Fonctions

- void [tour_initialize](#) ([Tour](#) *pTour, [Instance](#) pInstance)
- bool [tour_nextPermutation](#) ([Tour](#) *pPermutation)
- void [tour_calculLength](#) ([Tour](#) *pTour)
- void [tour_display](#) ([Tour](#) pTour, bool pDisplay)
- [Tour](#) [tour_randomWalk](#) ([Instance](#) pInstance)
- void [tour_2opt](#) ([Tour](#) *pTour, int pFirst, int pSecond)

4.6.1 Description détaillée

Programme de tests.

Auteur

Antoine de Roquemaurel

Version

0.1

Date

21/11/2012 22 :04 :13

Programme de test pour l'objet de gestion des chaînes de caractères Str_t.

4.6.2 Documentation des fonctions

4.6.2.1 void tour_2opt (Tour * *pTour*, int *pFirst*, int *pSecond*)

Paramètres

pTour

pFirst

pSecond

4.6.2.2 void tour_calculLength (Tour * *pTour*)

Paramètres

pTour

4.6.2.3 void tour_display (Tour *pTour*, bool *pDisplay*)

Paramètres

pTour

pDisplay

4.6.2.4 void tour_initialize (Tour * *pTour*, Instance *pInstance*)

Paramètres

pTour

pInstance

4.6.2.5 bool tour_nextPermutation (Tour * *pPermutation*)

Paramètres

pPermutation

Renvoie

4.6.2.6 Tour `tour_randomWalk` (Instance *pInstance*)

Paramètres

pInstance

Renvoie

4.7 Référence du fichier `/development/cours/L2/algo3/TP/voyageCommerce/lib/to`

Programme de tests.

```
#include <stdio.h>
#include <stdlib.h>
```

Classes

- struct `Town`
Objet des ville.

Fonctions

- void `town_initialize` (`Town *pTown`, const int `pId`, const int `pX`, const int `pY`)

4.7.1 Description détaillée

Programme de tests.

Auteur

Antoine de Roquemaurel

Version

0.1

Date

21/11/2012 22 :35 :19

Programme de test pour l'objet de gestion des chaînes de caractères `Str_t`.

4.7.2 Documentation des fonctions

4.7.2.1 void `town_initialize` (`Town * pTown`, const int *pId*, const int *pX*, const int *pY*)

Paramètres

pTown

pId

pX

pY

4.8 Référence du fichier /development/cours/L2/algo3/TP/voyageCommerce/src/br

Fonctions utiles.

```
#include "bruteForce.h"
```

```
#include "util.h"
```

Fonctions

– [Tour bruteForce_bestPath](#) (Instance pInstance)

4.8.1 Description détaillée

Fonctions utiles. Fonction main.

Programme de tests.

Auteur

Antoine de Roquemaurel

Version

0.1

Date

19/11/2012 16 :27 :39

Entêtes des fonctions pouvant être utiles dans tout le projet. Ce sont des fonctions simples, qui doivent être indépendantes du projet.

Auteur

Antoine de Roquemaurel

Version

0.1

Date

27/12/2012 17 :58 :29

Programme de test pour l'objet de gestion des chaînes de caractères Str_t.

Auteur

Antoine de Roquemaurel

Version

0.1

Date

01/12/2012 20 :33 :39

Entêtes des fonctions pouvant être utiles dans tout le projet. Ce sont des fonctions simples, qui doivent être indépendantes du projet.

Auteur

Antoine de Roquemaurel

Version

0.1

Date

21/11/2012 17 :42 :31

Entêtes des fonctions pouvant être utiles dans tout le projet. Ce sont des fonctions simples, qui doivent être indépendantes du projet.

Auteur

Antoine de Roquemaurel

Version

0.1

Date

21/11/2012 22 :03 :26

Entêtes des fonctions pouvant être utiles dans tout le projet. Ce sont des fonctions simples, qui doivent être indépendantes du projet.

Auteur

Antoine de Roquemaurel

Version

0.1

Date

19/11/2012 10 :42 :29

Point d'entrée du programme. Aucune fonction ne doit être déclarée Ce sont des fonctions simples, qui doivent être indépendantes du projet.

Auteur

Antoine de Roquemaurel

Version

0.1

Date

21/11/2012 17 :17 :18

Entêtes des fonctions pouvant être utiles dans tout le projet. Ce sont des fonctions simples, qui doivent être indépendantes du projet.

Auteur

Antoine de Roquemaurel

Version

0.1

Date

21/11/2012 22 :04 :08

Entêtes des fonctions pouvant être utiles dans tout le projet. Ce sont des fonctions simples, qui doivent être indépendantes du projet.

Auteur

Antoine de Roquemaurel

Version

0.1

Date

21/11/2012 22 :35 :14

Entêtes des fonctions pouvant être utiles dans tout le projet. Ce sont des fonctions simples, qui doivent être indépendantes du projet.

4.8.2 Documentation des fonctions

4.8.2.1 Tour bruteForce_bestPath (Instance *pInstance*)

Paramètres

pInstance

Renvoie

Index

/development/cours/L2/algo3/TP/voyageCommerce/lib/bruteForce.c, 9
/development/cours/L2/algo3/TP/voyageCommerce/lib/distance.c, 10
/development/cours/L2/algo3/TP/voyageCommerce/lib/errors.c, 11
/development/cours/L2/algo3/TP/voyageCommerce/lib/instance.c, 13
/development/cours/L2/algo3/TP/voyageCommerce/lib/parsing.c, 14
/development/cours/L2/algo3/TP/voyageCommerce/lib/town.c, 16
/development/cours/L2/algo3/TP/voyageCommerce/lib/town.h, 18
/development/cours/L2/algo3/TP/voyageCommerce/src/bruteForce.c, 19
bruteForce.h, 12
bruteForce_setNbArguments, 12
errors_setNoAlgoSpecified, 12
errors_setTagFNotFound, 12
errors_displayErrorMessage
errors.h, 12
errors_initialize
errors_setFileNotFound
errors_setLsrNotValidParameter
errors_setNbArguments
errors_setNoAlgoSpecified
errors_setTagFNotFound
errors.h, 12
Instance, 6
instance.h
instance_display, 13
instance_displayLinearVector, 13
instance_displayMatrix, 14
instance_initialize, 14
instance_initializeDistances, 14
instance_push, 14
instance_display
instance.h, 13
instance_displayLinearVector
instance.h, 13
instance_displayMatrix
instance.h, 14
instance_initialize
instance.h, 14
instance_initializeDistances
instance.h, 14
instance_push
instance.h, 14
parsing.h
AlgoType, 15
parsing_algoType, 15
parsing_parseFileName, 15
parsing_parseVerboseMode, 16
parsing_algoType
AlgoType, 15
parsing.h, 15
bruteForce.c, 21
bruteForce.h
bruteForce_bestPath, 9
bruteForce_bestPath
bruteForce.c, 21
bruteForce.h, 9
Distance, 5
distance.h
distance_betweenTowns, 10
distance_calculDistance, 10
distance_new, 11
distance_betweenTowns
distance.h, 10
distance_calculDistance
distance.h, 10
distance_new
distance.h, 11
Errors, 5
errors.h
errors_displayErrorMessage, 12
errors_initialize, 12
errors_setFileNotFound, 12
errors_setLsrNotValidParameter, 12

- [parsing.h](#), [15](#)
- [parsing_parseFileName](#)
 - [parsing.h](#), [15](#)
- [parsing_parseVerboseMode](#)
 - [parsing.h](#), [16](#)
- [Tour](#), [6](#)
- [tour.h](#)
 - [tour_2opt](#), [17](#)
 - [tour_calculLength](#), [17](#)
 - [tour_display](#), [17](#)
 - [tour_initialize](#), [17](#)
 - [tour_nextPermutation](#), [17](#)
 - [tour_randomWalk](#), [18](#)
- [tour_2opt](#)
 - [tour.h](#), [17](#)
- [tour_calculLength](#)
 - [tour.h](#), [17](#)
- [tour_display](#)
 - [tour.h](#), [17](#)
- [tour_initialize](#)
 - [tour.h](#), [17](#)
- [tour_nextPermutation](#)
 - [tour.h](#), [17](#)
- [tour_randomWalk](#)
 - [tour.h](#), [18](#)
- [Town](#), [7](#)
- [town.h](#)
 - [town_initialize](#), [18](#)
- [town_initialize](#)
 - [town.h](#), [18](#)