

RTVRPTW

1.0

Generated by Doxygen 1.9.8

1 Hierarchical Index	1
1.1 Class Hierarchy	1
2 Class Index	3
2.1 Class List	3
3 File Index	5
3.1 File List	5
4 Class Documentation	7
4.1 es.ull.esit.utilities.BellmanFord Class Reference	7
4.1.1 Constructor & Destructor Documentation	7
4.1.1.1 BellmanFord()	7
4.1.2 Member Function Documentation	8
4.1.2.1 getDistances()	8
4.1.2.2 getValue()	8
4.2 es.ull.esit.utilities.ExpositoUtilities Class Reference	8
4.3 top.mainTOPTW Class Reference	9
4.3.1 Member Function Documentation	9
4.3.1.1 main()	9
4.4 es.ull.esit.utils.Pair< F, S > Class Template Reference	9
4.5 es.ull.esit.utilities.PowerSet< E > Class Template Reference	10
4.6 top.TOPTW Class Reference	11
4.6.1 Constructor & Destructor Documentation	12
4.6.1.1 TOPTW()	12
4.6.2 Member Function Documentation	13
4.6.2.1 addNode()	13
4.6.2.2 addNodeDepot()	13
4.6.2.3 getDistance() [1/3]	13
4.6.2.4 getDistance() [2/3]	13
4.6.2.5 getDistance() [3/3]	15
4.6.2.6 getDueTime()	15
4.6.2.7 getNodes()	15
4.6.2.8 getReadyTime()	16
4.6.2.9 getScore() [1/2]	16
4.6.2.10 getScore() [2/2]	16
4.6.2.11 getServiceTime()	16
4.6.2.12 getTime()	17
4.6.2.13 getVehicles()	17
4.6.2.14 getX()	17
4.6.2.15 getY()	18
4.6.2.16 isDepot()	18
4.6.2.17 setDueTime()	18

4.6.2.18 setNodes()	19
4.6.2.19 setReadyTime()	19
4.6.2.20 setScore()	19
4.6.2.21 setServiceTime()	19
4.6.2.22 setX()	20
4.6.2.23 setY()	20
4.6.2.24 toString()	20
4.7 top.TOPTWEvaluator Class Reference	20
4.8 top.TOPTWGRASP Class Reference	21
4.8.1 Detailed Description	21
4.8.2 Member Function Documentation	21
4.8.2.1 aleatorySelectionRCL()	21
4.8.2.2 computeGreedySolution()	22
4.8.2.3 fuzzySelectionAlphaCutRCL()	22
4.8.2.4 fuzzySelectionBestFDRCL()	22
4.9 top.TOPTWReader Class Reference	23
4.9.1 Detailed Description	23
4.10 top.TOPTWRoute Class Reference	23
4.11 top.TOPTWSolution Class Reference	24
4.11.1 Detailed Description	24
5 File Documentation	25
5.1 src/utilities/ExpositoUtilities.java File Reference	25
5.1.1 Detailed Description	25
5.2 src/utilities/PowerSet.java File Reference	26
5.2.1 Detailed Description	26
5.3 src/utills/Pair.java File Reference	26
5.3.1 Detailed Description	27
Index	29

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

es.ull.esit.utilities.BellmanFord	7
es.ull.esit.utilities.ExpositoUtilities	8
Iterable	
es.ull.esit.utilities.PowerSet< E >	10
Iterator	
es.ull.esit.utilities.PowerSet< E >	10
top.mainTOPTW	9
es.ull.esit.utils.Pair< F, S >	9
top.TOPTW	11
top.TOPTWEvaluator	20
top.TOPTWGRASP	21
top.TOPTWReader	23
top.TOPTWRoute	23
top.TOPTWSolution	24

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

es.ull.esit.utilities.BellmanFord	7
es.ull.esit.utilities.ExpositoUtilities	8
top.mainTOPTW	9
es.ull.esit.utils.Pair< F, S >	9
es.ull.esit.utilities.PowerSet< E >	10
top.TOPTW	11
top.TOPTWEvaluator	20
top.TOPTWGRASP	
Clase que implementa el algoritmo GRASP para el problema TOPTW	21
top.TOPTWReader	23
top.TOPTWRoute	23
top.TOPTWSolution	24

Chapter 3

File Index

3.1 File List

Here is a list of all documented files with brief descriptions:

src/utilities/ ExpositoUtilities.java	
Utilities class	25
src/utilities/ PowerSet.java	
PowerSet class	26
src/utls/ Pair.java	
Pair class	26

Chapter 4

Class Documentation

4.1 es.ull.esit.utilities.BellmanFord Class Reference

Public Member Functions

- **BellmanFord** (int[][] distanceMatrix, int nodes, ArrayList< Integer > path)
Constructor of BellmanFord class.
- int[] **getDistances** ()
Returns the path of the graph.
- int **getValue** ()
- void **solve** ()
Solves the Bellman-Ford algorithm.

4.1.1 Constructor & Destructor Documentation

4.1.1.1 BellmanFord()

```
es.ull.esit.utilities.BellmanFord.BellmanFord (
    int distanceMatrix[ ][ ],
    int nodes,
    ArrayList< Integer > path )
```

Constructor of BellmanFord class.

Parameters

<i>distanceMatrix</i>	
<i>nodes</i>	
<i>path</i>	

4.1.2 Member Function Documentation

4.1.2.1 getDistances()

```
int[] es.ull.esit.utilities.BellmanFord.getDistances ( )
```

Returns the path of the graph.

Returns

4.1.2.2 getValue()

```
int es.ull.esit.utilities.BellmanFord.getValue ( )
```

Returns

The documentation for this class was generated from the following file:

- src/utilities/BellmanFord.java

4.2 es.ull.esit.utilities.ExpositoUtilities Class Reference

Static Public Member Functions

- static void **printFile** (String file)
- static String **simplifyString** (String string)
- static double[][] **multiplyMatrices** (double a[], double b[])
- static void **writeTextToFile** (String file, String text) throws IOException
- static String **getFormat** (String string)
- static String **getFormat** (double value)
- static String **getFormat** (double value, int zeros)
- static String **getFormat** (String string, int width)
- static String **getFormat** (String string, int width, int alignment)
- static String **getFormat** (ArrayList< String > strings, int width)
- static String **getFormat** (ArrayList< Integer > strings)
- static String **getFormat** (String[] strings, int width)
- static String **getFormat** (String[][] matrixStrings, int width)
- static String **getFormat** (String[] strings)
- static String **getFormat** (String[] strings, int[] width)
- static String **getFormat** (String[] strings, int[] width, int[] alignment)
- static boolean **isInteger** (String str)
- static boolean **isDouble** (String str)
- static boolean **isAcyclic** (int[][] distanceMatrix)
- static boolean **therelsPath** (int[][] distanceMatrix, int node)

Static Public Attributes

- static final int **DEFAULT_COLUMN_WIDTH** = 10
- static final int **ALIGNMENT_LEFT** = 1
- static final int **ALIGNMENT_RIGHT** = 2

The documentation for this class was generated from the following file:

- src/utilities/ **ExpositoUtilities.java**

4.3 top.mainTOPTW Class Reference

Static Public Member Functions

- static void **main** (String[] args)
Método principal del programa.

4.3.1 Member Function Documentation

4.3.1.1 main()

```
static void top.mainTOPTW.main (
    String[] args ) [static]
```

Método principal del programa.

Parameters

<i>args</i>	Argumentos de la línea de comandos.
-------------	-------------------------------------

Returns

No devuelve nada.

El método principal del programa se encarga de leer los ficheros de instancias, crear los objetos necesarios para resolver el problema y ejecutar el algoritmo GRASP con los parámetros indicados.

The documentation for this class was generated from the following file:

- src/top/mainTOPTW.java

4.4 es.ull.esit.utils.Pair< F, S > Class Template Reference

Public Member Functions

- **Pair** (F first, S second)
- boolean **equals** (Object o)
- int **hashCode** ()

Static Public Member Functions

- static< A, B > **Pair**< A, B > **create** (A a, B b)

Public Attributes

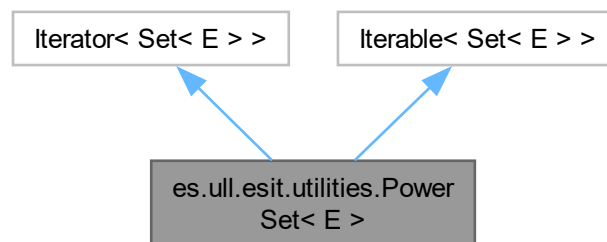
- final F **first**
- final S **second**

The documentation for this class was generated from the following file:

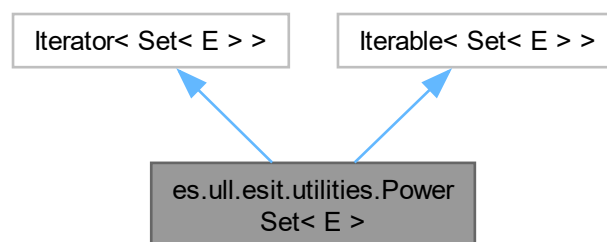
- src/utils/ **Pair.java**

4.5 es.ull.esit.utilities.PowerSet< E > Class Template Reference

Inheritance diagram for es.ull.esit.utilities.PowerSet< E >:



Collaboration diagram for es.ull.esit.utilities.PowerSet< E >:



Public Member Functions

- **PowerSet** (Set< E > set)
- boolean **hasNext** ()
- Set< E > **next** ()
- void **remove** ()
- Iterator< Set< E > > **iterator** ()

The documentation for this class was generated from the following file:

- src/utilities/ **PowerSet.java**

4.6 top.TOPTW Class Reference**Public Member Functions**

- **TOPTW** (int nodes, int routes)
Constructor de la clase TOPTW.
- boolean **isDepot** (int a)
Método que comprueba si un nodo es un depósito.
- double **getDistance** (int[] route)
Método que devuelve la distancia entre dos nodos.
- double **getDistance** (ArrayList< Integer > route)
Método que devuelve la distancia entre dos nodos.
- double **getDistance** (ArrayList< Integer >[] routes)
Método que devuelve la distancia entre dos nodos.
- void **calculateDistanceMatrix** ()
Método que calcula la distancia entre dos nodos.
- double **getMaxTimePerRoute** ()
Método que devuelve el tiempo máximo por ruta.
- void **setMaxTimePerRoute** (double maxTimePerRoute)
Método que establece el tiempo máximo por ruta.
- double **getMaxRoutes** ()
Método que devuelve el número máximo de rutas.
- void **setMaxRoutes** (double maxRoutes)
Método que establece el número máximo de rutas.
- int **getPOIs** ()
Método que devuelve el tiempo máximo por ruta.
- double **getDistance** (int i, int j)
Método que devuelve el tiempo máximo por ruta.
- double **getTime** (int i, int j)
Método que devuelve el tiempo máximo por ruta.
- int **getNodes** ()
Método que devuelve los nodos.
- void **setNodes** (int nodes)
Método que establece los nodos.
- double **getX** (int index)
Método que devuelve los depósitos.

- void **setX** (int index, double x)
Método que establece los depósitos.
- double **getY** (int index)
Método que devuelve los depósitos.
- void **setY** (int index, double y)
Método que establece los depósitos.
- double **getScore** (int index)
Método que devuelve el score.
- double[] **getScore** ()
Método que devuelve el score.
- void **setScore** (int index, double score)
Método que establece el score.
- double **getReadyTime** (int index)
Método que devuelve el ready time.
- void **setReadyTime** (int index, double readyTime)
Método que establece el ready time.
- double **getDueTime** (int index)
Método que devuelve el due time.
- void **setDueTime** (int index, double dueTime)
Método que establece el due time.
- double **getServiceTime** (int index)
Método que devuelve el service time.
- void **setServiceTime** (int index, double serviceTime)
Método que establece el service time.
- int **getVehicles** ()
Método que devuelve los depósitos.
- String **toString** ()
Método que convierte el objeto en una cadena de texto.
- int **addNode** ()
Método que añade un nodo.
- int **addNodeDepot** ()
Método que añade un nodo depósito.

4.6.1 Constructor & Destructor Documentation

4.6.1.1 TOPTW()

```
top.TOPTW.TOPTW (
    int nodes,
    int routes )
```

Constructor de la clase TOPTW.

Parameters

<i>nodes</i>	Número de nodos del problema.
<i>routes</i>	Número de rutas del problema.

Returns

No devuelve nada.

El constructor de la clase TOPTW se encarga de inicializar los atributos de la clase.

4.6.2 Member Function Documentation

4.6.2.1 addNode()

```
int top.TOPTW.addNode ( )
```

Método que añade un nodo.

Returns

Devuelve el número de nodos.

4.6.2.2 addNodeDepot()

```
int top.TOPTW.addNodeDepot ( )
```

Método que añade un nodo depósito.

Returns

Devuelve el número de nodos depósito.

4.6.2.3 getDistance() [1/3]

```
double top.TOPTW.getDistance (
    ArrayList< Integer > route )
```

Método que devuelve la distancia entre dos nodos.

Parameters

<i>route</i>	
--------------	--

Returns

Devuelve la distancia entre dos nodos.

4.6.2.4 getDistance() [2/3]

```
double top.TOPTW.getDistance (
    ArrayList< Integer >[] routes )
```

Método que devuelve la distancia entre dos nodos.

Parameters

<i>route</i>	
--------------	--

Returns

Devuelve la distancia entre dos nodos.

4.6.2.5 getDistance() [3/3]

```
double top.TOPTW.getDistance (
    int[] route )
```

Método que devuelve la distancia entre dos nodos.

Parameters

<i>route</i>	
--------------	--

Returns

Devuelve la distancia entre dos nodos.

4.6.2.6 getDueTime()

```
double top.TOPTW.getDueTime (
    int index )
```

Método que devuelve el due time.

Parameters

<i>index</i>	
--------------	--

Returns

Devuelve el due time.

4.6.2.7 getNodes()

```
int top.TOPTW.getNodes ( )
```

Método que devuelve los nodos.

Returns

4.6.2.8 getReadyTime()

```
double top.TOPTW.getReadyTime (
    int index )
```

Método que devuelve el ready time.

Parameters

<i>index</i>	
--------------	--

Returns

Devuelve el ready time.

4.6.2.9 getScore() [1/2]

```
double[] top.TOPTW.getScore ( )
```

Método que devuelve el score.

Returns

Devuelve el score.

4.6.2.10 getScore() [2/2]

```
double top.TOPTW.getScore (
    int index )
```

Método que devuelve el score.

Parameters

<i>index</i>	
--------------	--

Returns

Devuelve el score.

4.6.2.11 getServiceTime()

```
double top.TOPTW.getServiceTime (
    int index )
```

Método que devuelve el service time.

Parameters

<i>index</i>	
--------------	--

Returns

Devuelve el service time.

4.6.2.12 getTime()

```
double top.TOPTW.getTime (
    int i,
    int j )
```

Método que devuelve el tiempo máximo por ruta.

Parameters

<i>i</i>	
<i>j</i>	

Returns**4.6.2.13 getVehicles()**

```
int top.TOPTW.getVehicles ( )
```

Método que devuelve los depósitos.

Returns

Devuelve los depósitos.

4.6.2.14 getX()

```
double top.TOPTW.getX (
    int index )
```

Método que devuelve los depósitos.

Parameters

<i>index</i>	
--------------	--

Returns

Devuelve los depósitos.

4.6.2.15 getY()

```
double top.TOPTW.getY (
    int index )
```

Método que devuelve los depósitos.

Parameters

<i>index</i>	
--------------	--

Returns

Devuelve los depósitos.

4.6.2.16 isDepot()

```
boolean top.TOPTW.isDepot (
    int a )
```

Método que comprueba si un nodo es un depósito.

Parameters

<i>a</i>	
----------	--

Returns

Devuelve true si el nodo es un depósito, false en caso contrario.

4.6.2.17 setDueTime()

```
void top.TOPTW.setDueTime (
    int index,
    double dueTime )
```

Método que establece el due time.

Parameters

<i>index</i>	
<i>dueTime</i>	

4.6.2.18 setNodes()

```
void top.TOPTW.setNodes (
    int nodes )
```

Método que establece los nodos.

Parameters

<i>nodes</i>	
--------------	--

4.6.2.19 setReadyTime()

```
void top.TOPTW.setReadyTime (
    int index,
    double readyTime )
```

Método que establece el ready time.

Parameters

<i>index</i>	
<i>readyTime</i>	

4.6.2.20 setScore()

```
void top.TOPTW.setScore (
    int index,
    double score )
```

Método que establece el score.

Parameters

<i>index</i>	
<i>score</i>	

4.6.2.21 setServiceTime()

```
void top.TOPTW.setServiceTime (
    int index,
    double serviceTime )
```

Método que establece el service time.

Parameters

<i>index</i>	
<i>serviceTime</i>	

4.6.2.22 setX()

```
void top.TOPTW.setX (
    int index,
    double x )
```

Método que establece los depósitos.

Parameters

<i>index</i>	
<i>x</i>	

4.6.2.23 setY()

```
void top.TOPTW.setY (
    int index,
    double y )
```

Método que establece los depósitos.

Parameters

<i>index</i>	
<i>y</i>	

4.6.2.24 toString()

```
String top.TOPTW.toString ( )
```

Método que convierte el objeto en una cadena de texto.

Returns

Devuelve una cadena de texto con los datos del objeto.

The documentation for this class was generated from the following file:

- src/top/TOPTW.java

4.7 top.TOPTWEvaluator Class Reference

Public Member Functions

- void **evaluate** (**TOPTWSolution** solution)

Static Public Attributes

- static double **NO_EVALUATED** = -1.0

The documentation for this class was generated from the following file:

- src/top/TOPTWEvaluator.java

4.8 top.TOPTWGRASP Class Reference

Clase que implementa el algoritmo GRASP para el problema TOPTW.

Public Member Functions

- **TOPTWGRASP** (**TOPTWSolution** sol)
- void **GRASP** (int maxIterations, int maxSizeRCL)
- int **aleatorySelectionRCL** (int maxTRCL)
Selección aleatoria de un elemento de la lista restringida de candidatos.
- int **fuzzySelectionBestFDRCL** (ArrayList< double[] > rcl)
Selección del mejor elemento de la lista restringida de candidatos.
- int **fuzzySelectionAlphaCutRCL** (ArrayList< double[] > rcl, double alpha)
Selección aleatoria de un elemento de la lista restringida de candidatos.
- void **computeGreedySolution** (int maxSizeRCL)
Construcción de la solución inicial mediante una heurística greedy aleatorizada.
- void **updateSolution** (double[] candidateSelected, ArrayList< ArrayList< Double > > departureTimes)
- ArrayList< double[] > **comprehensiveEvaluation** (ArrayList< Integer > customers, ArrayList< ArrayList< Double > > departureTimes)
- **TOPTWSolution** **getSolution** ()
- void **setSolution** (**TOPTWSolution** solution)
- int **getSolutionTime** ()
- void **setSolutionTime** (int solutionTime)
- double **getMaxScore** ()

Static Public Attributes

- static double **NO_EVALUATED** = -1.0

4.8.1 Detailed Description

Clase que implementa el algoritmo GRASP para el problema TOPTW.

Esta clase implementa el algoritmo GRASP para el problema TOPTW. El algoritmo GRASP se basa en la construcción de una solución inicial mediante una heurística greedy aleatorizada y posteriormente se aplica una búsqueda local para mejorar la solución. El algoritmo se ejecuta un número determinado de veces y se devuelve la mejor solución encontrada.

4.8.2 Member Function Documentation**4.8.2.1 aleatorySelectionRCL()**

```
int top.TOPTWGRASP.aleatorySelectionRCL (
    int maxTRCL )
```

Selección aleatoria de un elemento de la lista restringida de candidatos.

Parameters

<i>maxTRCL</i>	
----------------	--

Returns

Devuelve la posición del elemento seleccionado de la lista restringida de candidatos.

4.8.2.2 computeGreedySolution()

```
void top.TOPTWGRASP.computeGreedySolution (
    int maxSizeRCL )
```

Construcción de la solución inicial mediante una heurística greedy aleatorizada.

Parameters

<i>maxSizeRCL</i>	
-------------------	--

Returns

No devuelve nada.

4.8.2.3 fuzzySelectionAlphaCutRCL()

```
int top.TOPTWGRASP.fuzzySelectionAlphaCutRCL (
    ArrayList< double[] > rcl,
    double alpha )
```

Selección aleatoria de un elemento de la lista restringida de candidatos.

Parameters

<i>rcl</i>	
<i>alpha</i>	

Returns

Devuelve la posición del elemento seleccionado de la lista restringida de candidatos.

4.8.2.4 fuzzySelectionBestFDRCL()

```
int top.TOPTWGRASP.fuzzySelectionBestFDRCL (
    ArrayList< double[] > rcl )
```

Selección del mejor elemento de la lista restringida de candidatos.

Parameters

<i>rcl</i>	
------------	--

Returns

Devuelve la posición del elemento seleccionado de la lista restringida de candidatos.

The documentation for this class was generated from the following file:

- src/top/TOPTWGRASP.java

4.9 top.TOPTWReader Class Reference

Static Public Member Functions

- static **TOPTW readProblem** (String filePath)

4.9.1 Detailed Description

TOPTWReader.java Clase que lee un fichero de texto con el formato de los problemas TOPTW.

Version

1.0.0

Since

1.0.0

The documentation for this class was generated from the following file:

- src/top/TOPTWReader.java

4.10 top.TOPTWRoute Class Reference

Public Member Functions

- int **getPredecesor** ()
- int **getSucesor** ()
- int **getId** ()
- void **setPredecesor** (int pre)
- void **setSucesor** (int suc)
- void **setId** (int id)

The documentation for this class was generated from the following file:

- src/top/TOPTWRoute.java

4.11 top.TOPTWSolution Class Reference

Public Member Functions

- **TOPTWSolution** (**TOPTW** problem)
- void **initSolution** ()
- boolean **isDepot** (int c)
- boolean **equals** (**TOPTWSolution** otherSolution)
- int **getAvailableVehicles** ()
- int **getCreatedRoutes** ()
- double **getDistance** (int x, int y)
- void **setAvailableVehicles** (int availableVehicles)
- int **getPredecessor** (int customer)
- int[] **getPredecessors** ()
- **TOPTW** **getProblem** ()
- double **getObjectiveFunctionValue** ()
- int **getPositionInRoute** (int customer)
- int **getSuccessor** (int customer)
- int[] **getSuccessors** ()
- int **getIndexRoute** (int index)
- double **getWaitingTime** (int customer)
- void **setObjectiveFunctionValue** (double objectiveFunctionValue)
- void **setPositionInRoute** (int customer, int position)
- void **setPredecessor** (int customer, int predecessor)
- void **setSuccessor** (int customer, int sucesor)
- void **setWaitingTime** (int customer, int waitingTime)
- String **getInfoSolution** ()
- double **evaluateFitness** ()
- int **addRoute** ()
- double **printSolution** ()

Static Public Attributes

- static final int **NO_INITIALIZED** = -1

4.11.1 Detailed Description

TOPTWSolution.java Clase que representa una solución al problema TOPTW.

Version

1.0.0

Since

1.0.0

The documentation for this class was generated from the following file:

- src/top/TOPTWSolution.java

Chapter 5

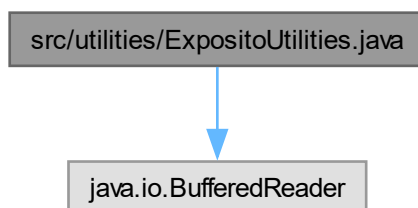
File Documentation

5.1 src/utilities/ExpositoUtilities.java File Reference

Utilities class.

```
import java.io.BufferedReader;
```

Include dependency graph for ExpositoUtilities.java:



Classes

- class **es.ull.esit.utilities.ExpositoUtilities**

5.1.1 Detailed Description

Utilities class.

Date

28/04/2020

Version

1.0.0

Note

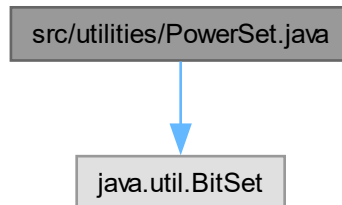
Esta clase contiene métodos de utilidad para el proyecto.

5.2 src/utilities/PowerSet.java File Reference

PowerSet class.

```
import java.util.BitSet;
```

Include dependency graph for PowerSet.java:



Classes

- class **es.ull.esit.utilities.PowerSet**< E >

5.2.1 Detailed Description

PowerSet class.

Date

28/04/2020

Version

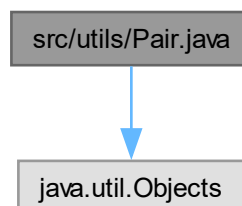
1.0.0

5.3 src/utills/Pair.java File Reference

Pair class.

```
import java.util.Objects;
```

Include dependency graph for Pair.java:



Classes

- class **es.ull.esit.utls.Pair**< F, S >

5.3.1 Detailed Description

Pair class.

Date

28/04/2020

Version

1.0.0

Note

Esta clase contiene la implementación de un par de elementos.

Parameters

<F>	
<S>	

Index

- addNode
 - top.TOPTW, 13
- addNodeDepot
 - top.TOPTW, 13
- aleatorySelectionRCL
 - top.TOPTWGRASP, 21
- BellmanFord
 - es.ull.esit.utilities.BellmanFord, 7
- computeGreedySolution
 - top.TOPTWGRASP, 22
- es.ull.esit.utilities.BellmanFord, 7
 - BellmanFord, 7
 - getDistances, 8
 - getValue, 8
- es.ull.esit.utilities.ExpositoUtilities, 8
- es.ull.esit.utilities.PowerSet< E >, 10
- es.ull.esit.utils.Pair< F, S >, 9
- fuzzySelectionAlphaCutRCL
 - top.TOPTWGRASP, 22
- fuzzySelectionBestFDRCL
 - top.TOPTWGRASP, 22
- getDistance
 - top.TOPTW, 13, 15
- getDistances
 - es.ull.esit.utilities.BellmanFord, 8
- getDueTime
 - top.TOPTW, 15
- getNodes
 - top.TOPTW, 15
- getReadyTime
 - top.TOPTW, 15
- getScore
 - top.TOPTW, 16
- getServiceTime
 - top.TOPTW, 16
- getTime
 - top.TOPTW, 17
- getValue
 - es.ull.esit.utilities.BellmanFord, 8
- getVehicles
 - top.TOPTW, 17
- getX
 - top.TOPTW, 17
- getY
 - top.TOPTW, 18
- isDepot
 - top.TOPTW, 18
- main
 - top.mainTOPTW, 9
- setDueTime
 - top.TOPTW, 18
- setNodes
 - top.TOPTW, 18
- setReadyTime
 - top.TOPTW, 19
- setScore
 - top.TOPTW, 19
- setServiceTime
 - top.TOPTW, 19
- setX
 - top.TOPTW, 20
- setY
 - top.TOPTW, 20
- src/utilities/ExpositoUtilities.java, 25
- src/utilities/PowerSet.java, 26
- src/utils/Pair.java, 26
- top.mainTOPTW, 9
 - main, 9
- top.TOPTW, 11
 - addNode, 13
 - addNodeDepot, 13
 - getDistance, 13, 15
 - getDueTime, 15
 - getNodes, 15
 - getReadyTime, 15
 - getScore, 16
 - getServiceTime, 16
 - getTime, 17
 - getVehicles, 17
 - getX, 17
 - getY, 18
 - isDepot, 18
 - setDueTime, 18
 - setNodes, 18
 - setReadyTime, 19
 - setScore, 19
 - setServiceTime, 19
 - setX, 20
 - setY, 20
 - TOPTW, 12
 - toString, 20
- top.TOPTWEvaluator, 20

- top.TOPTWGRASP, 21
 - aleatorySelectionRCL, 21
 - computeGreedySolution, 22
 - fuzzySelectionAlphaCutRCL, 22
 - fuzzySelectionBestFDRCL, 22
- top.TOPTWReader, 23
- top.TOPTWRoute, 23
- top.TOPTWSolution, 24
- TOPTW
 - top.TOPTW, 12
- toString
 - top.TOPTW, 20