

is2010final

1

Generated by Doxygen 1.7.2

Tue Nov 2 2010 00:22:32



# Contents

<b>1</b>	<b>Main Page</b>	<b>1</b>
<b>2</b>	<b>Class Index</b>	<b>5</b>
2.1	Class List . . . . .	5
<b>3</b>	<b>File Index</b>	<b>7</b>
3.1	File List . . . . .	7
<b>4</b>	<b>Class Documentation</b>	<b>9</b>
4.1	util.Pair Class Reference . . . . .	9
4.1.1	Detailed Description . . . . .	9
4.1.2	Constructor & Destructor Documentation . . . . .	10
4.1.2.1	Pair . . . . .	10
4.1.3	Member Function Documentation . . . . .	10
4.1.3.1	getX . . . . .	10
4.1.3.2	getY . . . . .	10
4.1.3.3	setX . . . . .	10
4.1.3.4	setY . . . . .	10
4.2	test.testClass Class Reference . . . . .	11
4.2.1	Detailed Description . . . . .	11
4.2.2	Member Function Documentation . . . . .	11
4.2.2.1	checkDateTest . . . . .	11
4.2.2.2	checkDniTest . . . . .	12
4.2.2.3	checkHour . . . . .	12
4.2.2.4	checkStringTest . . . . .	13
4.2.2.5	daysDiff . . . . .	13
4.2.2.6	hourDiffTest . . . . .	13
4.2.2.7	medicPayTest . . . . .	14
4.2.2.8	totalPayTest . . . . .	14
4.3	util.utilTest Class Reference . . . . .	15
4.3.1	Detailed Description . . . . .	15
4.3.2	Member Function Documentation . . . . .	15
4.3.2.1	checkDate . . . . .	15
4.3.2.2	checkDni . . . . .	16
4.3.2.3	checkHour . . . . .	17
4.3.2.4	checkString . . . . .	17
4.3.2.5	daysDiff . . . . .	18
4.3.2.6	hourDiff . . . . .	19
4.3.2.7	medicPay . . . . .	19

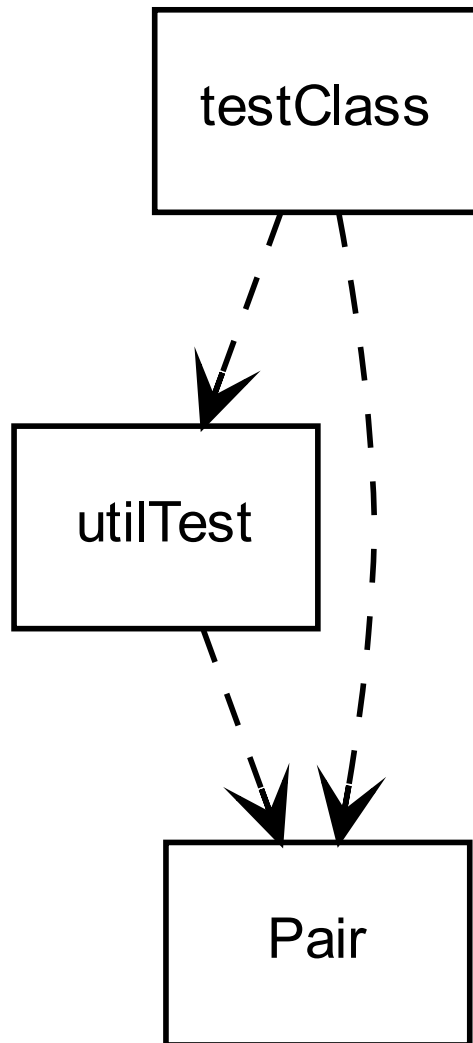
4.3.2.8	<a href="#">totalPay</a>	20
<b>5</b>	<b>File Documentation</b>	<b>23</b>
5.1	<a href="#">src/test/testClass.java File Reference</a>	23
5.1.1	<a href="#">Detailed Description</a>	23
5.2	<a href="#">src/util/Pair.java File Reference</a>	23
5.2.1	<a href="#">Detailed Description</a>	24
5.3	<a href="#">src/util/utilTest.java File Reference</a>	24
5.3.1	<a href="#">Detailed Description</a>	24

## **Chapter 1**

# **Main Page**

Proyecto de Ingenieria de Software 2010

Relaciones entre clases por medio de un diagrama de lineas de dot:



*Nota: para mas informacion click en las clases del grafico*

**AUTORES**

BERNAL, Matias

BRESSAN, Gonzalo

JAULE, Marcos

ODORIZZI, Eduardo

[Repositorio SVN en Google Code](#)





## Chapter 2

# Class Index

### 2.1 Class List

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

[util.Pair](#) (La clase pair implementa el tipo Par ) . . . . . [9](#)  
para verificar el correcto funcionamiento de las funciones de validacion  
de la clase utilTest.  
)[11](#)  
Cadenas, Fechas, Tiempo transcurrido )[15](#)



## Chapter 3

# File Index

### 3.1 File List

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

src/test/ <a href="#">testClass.java</a> (Este archivos contiene la implementacion de las prue- bas ) . . . . .	23
src/util/ <a href="#">Pair.java</a> (Este archivo contiene la impementación del tipo Par ) . . . .	23
src/util/ <a href="#">utilTest.java</a> (This file contains the DoxygenExample class with the main() function ) . . . . .	24



## Chapter 4

# Class Documentation

### 4.1 util.Pair Class Reference

La clase pair implementa el tipo Par.

#### Public Member Functions

- [Pair](#) ()  
*constructor sin parametros  
crea el par (0,0).*
- [Pair](#) (int x, int y)  
*constructor con parametros  
crea el par (x,y).*
- int [getX](#) ()  
*devuelve la primer componente del par.*
- void [setX](#) (int x)
- int [getY](#) ()  
*devuelve la segunda componente del par.*
- void [setY](#) (int y)

#### 4.1.1 Detailed Description

La clase pair implementa el tipo Par. El tipo par representa dos nmeros escritos en un cierto orden.

Usualmente estscritos entre paresis, as4,5)

## 4.1.2 Constructor & Destructor Documentation

### 4.1.2.1 util.Pair.Pair ( int x, int y )

constructor con parametros

crea el par (x,y).

#### Parameters

<i>x</i>	primer valor del par.
<i>y</i>	segundo valor del par.

## 4.1.3 Member Function Documentation

### 4.1.3.1 int util.Pair.getX ( )

devuelve la primer componente del par.

#### Returns

*x*

### 4.1.3.2 int util.Pair.getY ( )

devuelve la segunda componente del par.

#### Returns

*y*

### 4.1.3.3 void util.Pair.setX ( int x )

setea la primer componente del par.

#### Parameters

<i>x</i>	
----------	--

### 4.1.3.4 void util.Pair.setY ( int y )

setea la segunda componente del par.

#### Parameters

<i>y</i>	
----------	--

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

- src/util/[Pair.java](#)

## 4.2 test.testClass Class Reference

La clase [testClass](#) declara una serie de metodos en junit para verificar el correcto funcionamiento de las funciones de validacion de la clase utilTest.

### Public Member Functions

- void [checkStringTest](#) ()
- void [checkDniTest](#) ()
- void [checkDateTest](#) ()
- void [hourDiffTest](#) ()
- void [daysDiff](#) ()
- void [checkHour](#) ()
- void [medicPayTest](#) ()
- void [totalPayTest](#) ()

### 4.2.1 Detailed Description

La clase [testClass](#) declara una serie de metodos en junit para verificar el correcto funcionamiento de las funciones de validacion de la clase utilTest.

### 4.2.2 Member Function Documentation

#### 4.2.2.1 void test.testClass.checkDateTest ( )

Valida el funcionamiento de la funcion checkDate dentro de la clase utilTest.

Se usa tablas de desici

Here is the call graph for this function:

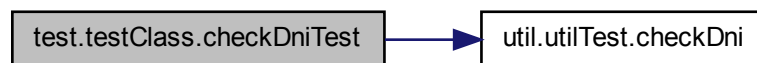


#### 4.2.2.2 void test.testClass.checkDniTest ( )

Valida el funcionamiento de la funcion checkDni dentro de la clase utilTest.

Se usa clases de equivalencia con testing fuerte.

Here is the call graph for this function:



#### 4.2.2.3 void test.testClass.checkHour ( )

Valida el funcionamiento de la funcion daysDiff dentro de la clase utilTest.

Se usa testing de robustez (valor limite).

Here is the call graph for this function:



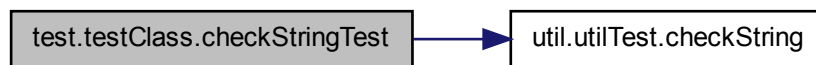


#### 4.2.2.4 void test.testClass.checkStringTest ( )

Valida el funcionamiento de la funcion checkString dentro de la clase utilTest.

Se usa clases de equivalencia con testing fuerte.

Here is the call graph for this function:

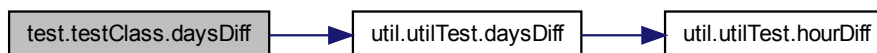


#### 4.2.2.5 void test.testClass.daysDiff ( )

Valida el funcionamiento de la funcion daysDiff dentro de la clase utilTest.

Se usa clases de equivalencia con testing fuerte.

Here is the call graph for this function:



#### 4.2.2.6 void test.testClass.hourDiffTest ( )

Valida el funcionamiento de la funcion hourDiff dentro de la clase utilTest.

Se usa tablas de desici

Here is the call graph for this function:

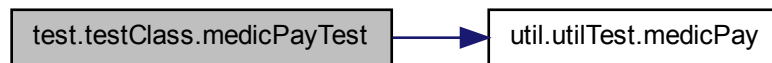


#### 4.2.2.7 void test.testClass.medicPayTest ( )

Valida el funcionamiento de la funcion medicPay dentro de la clase utilTest.

Se usa clases de equivalencia con testing fuerte.

Here is the call graph for this function:



#### 4.2.2.8 void test.testClass.totalPayTest ( )

Valida el funcionamiento de la funcion totalPay dentro de la clase utilTest.

Se usa clases de equivalencia con testing fuerte.

Here is the call graph for this function:



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

- src/test/testClass.java

## 4.3 util.utilTest Class Reference

La clase `utilTest` declara una serie de metodos para realizar validaciones de, entre otros, Cadenas, Fechas, Tiempo transcurrido.

### Static Public Member Functions

- static boolean `checkString` (String s)  
*checkString Metodo que verifica si un String pasado como parametro es valido o no. Un string es valido si solo contiene caracteres de a-z o de A-Z y ademas si su longitud esta entre 1 y 60.*
- static boolean `checkDni` (String dni)
- static boolean `checkDate` (int diaA, int mes, int anio)
- static int `hourDiff` (int diaE, int mesE, int anioE, int hsE, int minE, int diaS, int mesS, int anioS, int hsS, int minS)
- static float `medicPay` (Vector< `Pair` > consultas, float valor, int horasAlquiler, float alquilerxhs)
- static float `totalPay` (Float costoDia, `Pair`[] comidas, Vector< `Pair` > medicamentos, int dias, Float descuento)
- static int `daysDiff` (int diaE, int mesE, int anioE, int hsE, int minE, int diaS, int mesS, int anioS, int hsS, int minS)
- static boolean `checkHour` (int hs, int min)

#### 4.3.1 Detailed Description

La clase `utilTest` declara una serie de metodos para realizar validaciones de, entre otros, Cadenas, Fechas, Tiempo transcurrido.

#### 4.3.2 Member Function Documentation

##### 4.3.2.1 static boolean util.utilTest.checkDate ( int diaA, int mes, int anio ) [static]

`checkDate` valida que la fecha ingresada sea correcta.

una fecha es correcta si el dia es menor que 28 para el mes 2 o 29 en caso de que el aa bisiesto.

el dia es menor o igual que 30 para los meses: 1, 3, 5, 7, 8, 10 y 12

o el dia es menor o igual que 31 para los meses: 4, 6, 9 y 11.

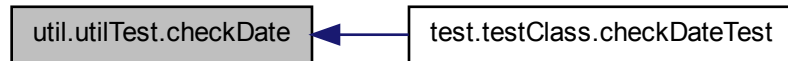
**Parameters**

<i>diaA</i>	
<i>mes</i>	
<i>anio</i>	

**Returns**

true si es valido, false si no lo es

Here is the caller graph for this function:

**4.3.2.2 static boolean util.utilTest.checkDni ( String dni ) [static]**

checkDni metodo que verifica si un String pasado como parametro representando un dni es valido o no. Un dni es valido si cumple con la mascara XX.XXX.XXX donde X debe ser un numero de 0-9

**Parameters**

<i>String</i>	s es el string a verificar.
---------------	-----------------------------

**Returns**

boolean true si es valido y false en caso contrario.

Here is the caller graph for this function:



**4.3.2.3 static boolean util.utilTest.checkHour ( int *hs*, int *min* ) [static]**

checkHour comprueba que la hora pasada como parametro es correcta

**Parameters**

<i>hs</i>	hora
<i>min</i>	minutos

Here is the caller graph for this function:

**4.3.2.4 static boolean util.utilTest.checkString ( String *s* ) [static]**

checkString Metodo que verifica si un String pasado como parametro es valido o no. Un string es valido si solo contiene caracteres de a-z o de A-Z y ademas si su longitud esta entre 1 y 60.

**Parameters**

<i>String</i>	<i>s</i> es el string a verificar.
---------------	------------------------------------

**Returns**

boolean true si es valido y false en caso contrario.

Here is the caller graph for this function:



#### 4.3.2.5 static int util.utilTest.daysDiff ( int *diaE*, int *mesE*, int *anioE*, int *hsE*, int *minE*, int *diaS*, int *mesS*, int *anioS*, int *hsS*, int *minS* ) [static]

daysDiff permite calcular la cantidad de d que un paciente ha estado internado. A partir de las 10:00 de la ma ya se considera un nuevo d es decir, si el paciente es dado de alta a las 10:01 ya se le cobra un nuevo d

##### Parameters

int	diaE representa al dia de inicio.
int	mesE representa al mes de inicio.
int	anioE representa al anio de inicio.
int	hsE representa a la hs de inicio.
int	minS representa a los minutos de fin.
int	diaS representa al dia de fin.
int	mesS representa al mes de fin.
int	anioS representa al anio de fin.
int	hsS representa a la hs de fin.
int	minS representa a los minutos de fin.

##### Returns

float representando la cantidad de dias de internacion que se le deben cobrar al paciente.

Here is the call graph for this function:



Here is the caller graph for this function:



**4.3.2.6** static int util.utilTest.hourDiff ( int *diaE*, int *mesE*, int *anioE*, int *hsE*, int *minE*, int *diaS*, int *mesS*, int *anioS*, int *hsS*, int *minS* ) [static]

hourDiff indica la cantidad de horas la cantidad de horas transcurridas entre dos fechas (fecha y hora).

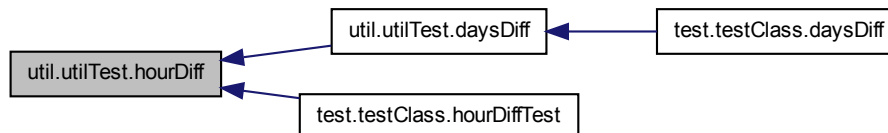
#### Parameters

int	diaE representa al dia de inicio.
int	mesE representa al mes de inicio.
int	anioE representa al anio de inicio.
int	hsE representa a la hs de inicio.
int	minS representa a los minutos de fin.
int	diaS representa al dia de fin.
int	mesS representa al mes de fin.
int	anioS representa al anio de fin.
int	hsS representa a la hs de fin.
int	minS representa a los minutos de fin.
int	anio representa el anio a chequear.

#### Returns

int que representa la cantidad de horas que hay entre 2 fechas.

Here is the caller graph for this function:



**4.3.2.7** static float util.utilTest.medicPay ( Vector< Pair > *consultas*, float *valor*, int *horasAlquiler*, float *alquilerxhs* ) [static]

Una funcie permita calcular el sueldo de un m dico en funci la cantidad de consultas atendidas al mes, las obras sociales y el valor de la consulta. A este resultado se le deberscontar la cantidad de horas que el profesional ha utilizado la clca(en concepto de alquiler).

#### Parameters

<i>consultas</i>	
<i>valor</i>	
<i>ho- rasAlquiler</i>	
<i>alquilerxhs</i>	

**Returns**

pago

Here is the caller graph for this function:

**4.3.2.8 static float util.utilTest.totalPay ( Float *costoDia*, Pair[] *comidas*, Vector< Pair > *medicamentos*, int *dias*, Float *descuento* ) [static]**

Una funcie permita calcular el monto total que debe abonar un paciente internado en funcil costo del de internaciantidad y costos de comidas recibidas, cantidad y costo de medicamentos recibidos. A todo esto debe calcularse un descuento correspondiente al monto que cubre la mutual.

**Parameters**

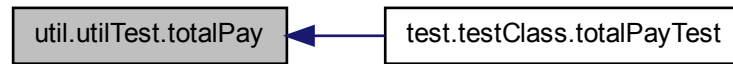
<i>costoDia</i>	
<i>comidas</i>	
<i>medicamen- tos</i>	
<i>dias</i>	
<i>descuento</i>	

**Returns**

monto



Here is the caller graph for this function:



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

- [src/util/utilTest.java](#)



## Chapter 5

# File Documentation

### 5.1 `src/test/testClass.java` File Reference

Este archivos contiene la implementacion de las pruebas.

#### Classes

- class [test.testClass](#)

*La clase [testClass](#) declara una serie de metodos en junit para verificar el correcto funcionamiento de las funciones de validacion de la clase `utilTest`.*

#### 5.1.1 Detailed Description

Este archivos contiene la implementacion de las pruebas.

#### Author

BERNAL, Matias  
BRESSAN, Gonzalo  
JAULE, Marcos  
ODORIZZI, Eduardo

#### Date

November, 1st 2010

### 5.2 `src/util/Pair.java` File Reference

Este archivo contiene la impementación del tipo `Par`.

## Classes

- class [util.Pair](#)

*La clase pair implementa el tipo Par.*

### 5.2.1 Detailed Description

Este archivo contiene la implementación del tipo Par.

#### Author

BERNAL, Matias  
BRESSAN, Gonzalo  
JAULE, Marcos  
ODORIZZI, Eduardo

#### Date

November, 1st 2010

## 5.3 src/util/utilTest.java File Reference

This file contains the DoxygenExample class with the main() function.

## Classes

- class [util.utilTest](#)

*La clase [utilTest](#) declara una serie de metodos para realizar validaciones de, entre otros, Cadenas, Fechas, Tiempo transcurrido.*

### 5.3.1 Detailed Description

This file contains the DoxygenExample class with the main() function.

#### Author

BERNAL, Matias  
BRESSAN, Gonzalo  
JAULE, Marcos  
ODORIZZI, Eduardo

#### Date

November, 1st 2010

# Index

checkDate  
    util::utilTest, [15](#)  
checkDateTest  
    test::testClass, [11](#)  
checkDni  
    util::utilTest, [16](#)  
checkDniTest  
    test::testClass, [12](#)  
checkHour  
    test::testClass, [12](#)  
    util::utilTest, [16](#)  
checkString  
    util::utilTest, [17](#)  
checkStringTest  
    test::testClass, [13](#)  
  
daysDiff  
    test::testClass, [13](#)  
    util::utilTest, [17](#)  
  
getX  
    util::Pair, [10](#)  
getY  
    util::Pair, [10](#)  
  
hourDiff  
    util::utilTest, [18](#)  
hourDiffTest  
    test::testClass, [13](#)  
  
medicPay  
    util::utilTest, [19](#)  
medicPayTest  
    test::testClass, [14](#)  
  
Pair  
    util::Pair, [10](#)  
  
setX  
    util::Pair, [10](#)  
setY  
    util::Pair, [10](#)  
  
src/test/testClass.java, [23](#)  
src/util/Pair.java, [23](#)  
src/util/utilTest.java, [24](#)  
  
test::testClass, [11](#)  
    checkDateTest, [11](#)  
    checkDniTest, [12](#)  
    checkHour, [12](#)  
    checkStringTest, [13](#)  
    daysDiff, [13](#)  
    hourDiffTest, [13](#)  
    medicPayTest, [14](#)  
    totalPayTest, [14](#)  
totalPay  
    util::utilTest, [20](#)  
totalPayTest  
    test::testClass, [14](#)  
  
util::Pair, [9](#)  
    getX, [10](#)  
    getY, [10](#)  
    Pair, [10](#)  
    setX, [10](#)  
    setY, [10](#)  
util::utilTest, [15](#)  
    checkDate, [15](#)  
    checkDni, [16](#)  
    checkHour, [16](#)  
    checkString, [17](#)  
    daysDiff, [17](#)  
    hourDiff, [18](#)  
    medicPay, [19](#)  
    totalPay, [20](#)