

## Dissimulation

Generated by Doxygen 1.8.10

Sun Jan 10 2016 20:56:52



# Contents



# Chapter 1

## Namespace Index

### 1.1 Packages

Here are the packages with brief descriptions (if available):

|                                 |       |    |
|---------------------------------|-------|----|
| <a href="#">gestion_options</a> | ..... | ?? |
|---------------------------------|-------|----|



## Chapter 2

# Hierarchical Index

### 2.1 Class Hierarchy

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

|                                                           |    |
|-----------------------------------------------------------|----|
| gestion_options.BaseArgument . . . . .                    | ?? |
| gestion_options.Argument . . . . .                        | ?? |
| img_analyses.Channels . . . . .                           | ?? |
| Exception                                                 |    |
| exceptions.ArgumentAlreadyUsedException . . . . .         | ?? |
| exceptions.ChannelException . . . . .                     | ?? |
| exceptions.ImageTooShort . . . . .                        | ?? |
| exceptions.InvalidArgException . . . . .                  | ?? |
| exceptions.InvalidValueException . . . . .                | ?? |
| exceptions.MandatoryArgumentNotDefinedException . . . . . | ?? |
| exceptions.MessageContainsMagicNumberException . . . . .  | ?? |
| exceptions.TooManyValuesException . . . . .               | ?? |
| LSB.WayStrategy.Hider . . . . .                           | ?? |
| LSB.WayStrategy.BHider . . . . .                          | ?? |
| LSB.WayStrategy.GHider . . . . .                          | ?? |
| LSB.WayStrategy.RHider . . . . .                          | ?? |
| img_analyses.ImageMatrix . . . . .                        | ?? |
| Launch . . . . .                                          | ?? |
| message.Message . . . . .                                 | ?? |
| gestion_options.ProgramOptions . . . . .                  | ?? |
| action.Ways . . . . .                                     | ?? |
| LSB.WayStrategy . . . . .                                 | ?? |
| LSB.DirectWay . . . . .                                   | ?? |
| LSB.ExternSpiralWay . . . . .                             | ?? |
| LSB.InternSpiralWay . . . . .                             | ?? |
| LSB.RevertWay . . . . .                                   | ?? |





## Chapter 3

# Class Index

### 3.1 Class List

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

|                                                                 |    |
|-----------------------------------------------------------------|----|
| <a href="#">gestion_options.Argument</a>                        | ?? |
| <a href="#">exceptions.ArgumentAlreadyUsedException</a>         | ?? |
| <a href="#">gestion_options.BaseArgument</a>                    | ?? |
| <a href="#">LSB.WayStrategy.BHider</a>                          | ?? |
| <a href="#">exceptions.ChannelException</a>                     | ?? |
| <a href="#">img_analyses.Channels</a>                           | ?? |
| <a href="#">LSB.DirectWay</a>                                   | ?? |
| <a href="#">LSB.ExternSpiralWay</a>                             | ?? |
| <a href="#">LSB.WayStrategy.GHider</a>                          | ?? |
| <a href="#">LSB.WayStrategy.Hider</a>                           | ?? |
| <a href="#">img_analyses.ImageMatrix</a>                        | ?? |
| <a href="#">exceptions.ImageTooShort</a>                        | ?? |
| <a href="#">LSB.InternSpiralWay</a>                             | ?? |
| <a href="#">exceptions.InvalidArgException</a>                  | ?? |
| <a href="#">exceptions.InvalidValueException</a>                | ?? |
| <a href="#">Launch</a>                                          | ?? |
| <a href="#">exceptions.MandatoryArgumentNotDefinedException</a> | ?? |
| <a href="#">message.Message</a>                                 | ?? |
| <a href="#">exceptions.MessageContainsMagicNumberException</a>  | ?? |
| <a href="#">gestion_options.ProgramOptions</a>                  | ?? |
| <a href="#">LSB.RevertWay</a>                                   | ?? |
| <a href="#">LSB.WayStrategy.RHider</a>                          | ?? |
| <a href="#">exceptions.TooManyValuesException</a>               | ?? |
| <a href="#">action.Ways</a>                                     | ?? |
| <a href="#">LSB.WayStrategy</a>                                 | ?? |



## Chapter 4

# File Index

### 4.1 File List

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

|                                                                          |                                                                                                |    |
|--------------------------------------------------------------------------|------------------------------------------------------------------------------------------------|----|
| src/action/ <a href="#">Ways.java</a>                                    | Define the different pattern to go through the image . . . . .                                 | ?? |
| src/exceptions/ <a href="#">ArgumentAlreadyUsedException.java</a>        | Exception which is raised when an argument is specified more than one time . . . . .           | ?? |
| src/exceptions/ <a href="#">ChannelException.java</a>                    | Exception which is raised when a specified channel is wrong . . . . .                          | ?? |
| src/exceptions/ <a href="#">ImageTooShort.java</a>                       | Exception which is raised when the given image is too short to hide the message . . . . .      | ?? |
| src/exceptions/ <a href="#">InvalidArgException.java</a>                 | Exception which is raised when an argument is invalid . . . . .                                | ?? |
| src/exceptions/ <a href="#">InvalidValueException.java</a>               | Exception which is raised when a value passed is wrong . . . . .                               | ?? |
| src/exceptions/ <a href="#">MessageContainsMagicNumberException.java</a> | Exception which is raised when the message contain the magic number . . . . .                  | ?? |
| src/exceptions/ <a href="#">TooManyValuesException.java</a>              | Exception which is raised when an argument has too many passed values . . . . .                | ?? |
| src/gestion_options/ <a href="#">Argument.java</a>                       | Represent a passed argument but more specifically than it super class . . . . .                | ?? |
| src/gestion_options/ <a href="#">BaseArgument.java</a>                   | Represent a passed argument . . . . .                                                          | ?? |
| src/img_analyses/ <a href="#">Channels.java</a>                          | Define all usable channels to hide information . . . . .                                       | ?? |
| src/img_analyses/ <a href="#">ImageMatrix.java</a>                       | Represent the image as a matrix with the different action that you can perform on it . . . . . | ?? |
| src/LSB/ <a href="#">DirectWay.java</a>                                  | The direct pattern . . . . .                                                                   | ?? |
| src/message/ <a href="#">Message.java</a>                                | Prepare the message to be hidden . . . . .                                                     | ?? |



## Chapter 5

# Namespace Documentation

### 5.1 Package `gestion_options`

#### Classes

- class [Argument](#)
- class [BaseArgument](#)
- class [ProgramOptions](#)

#### 5.1.1 Detailed Description

Created by Loïc on 11/7/2015.



## Chapter 6

# Class Documentation

### 6.1 gestion\_options.Argument Class Reference

Inheritance diagram for gestion\_options.Argument:

#### Public Member Functions

- [Argument](#) (String name, boolean allowMultipleValues, String[] defaultValues, String[] validValues)  
*Constructor to create the optional argument.*
- [Argument](#) (String name, boolean allowMultipleValues, String[] validValues)  
*Constructor to create the mandatory argument.*
- ArrayList< String > [getPassedValues](#) ()  
*Get the given value of the argument.*
- ArrayList< String > [getDefaultValues](#) ()  
*Get the default value of the argument.*
- boolean [checkPassedValues](#) (ArrayList< String > passedValues) throws InvalidValueException, TooManyValuesException  
*Used to check if the values are right for the argument.*

#### Additional Inherited Members

##### 6.1.1 Constructor & Destructor Documentation

###### 6.1.1.1 gestion\_options.Argument.Argument ( String name, boolean allowMultipleValues, String[] defaultValues, String[] validValues )

Constructor to create the optional argument.

Call the super argument but specify if the values needs a validation

#### Parameters

|                            |                                                             |
|----------------------------|-------------------------------------------------------------|
| <i>name</i>                | Name of the argument                                        |
| <i>allowMultipleValues</i> | <i>boolean</i> representing if you can pass multiple values |

|                      |                                                         |
|----------------------|---------------------------------------------------------|
| <i>defaultValues</i> | Represent the default values of the argument            |
| <i>validValues</i>   | Represent the different possible values of the argument |

#### 6.1.1.2 `gestion_options.Argument.Argument ( String name, boolean allowMultipleValues, String[] validValues )`

Constructor to create the mandatory argument.

Call the super argument but specify if the values needs a validation

Parameters

|                            |                                                             |
|----------------------------|-------------------------------------------------------------|
| <i>name</i>                | Name of the argument                                        |
| <i>allowMultipleValues</i> | <i>boolean</i> representing if you can pass multiple values |
| <i>validValues</i>         | Represent the different possible values of the argument     |

### 6.1.2 Member Function Documentation

#### 6.1.2.1 `boolean gestion_options.Argument.checkPassedValues ( ArrayList<String> passedValues ) throws InvalidValueException, TooManyValuesException`

Used to check if the values are right for the argument.

Parameters

|                     |                 |
|---------------------|-----------------|
| <i>passedValues</i> | values to check |
|---------------------|-----------------|

#### 6.1.2.2 `ArrayList<String> gestion_options.Argument.getDefaultValues ( )`

Get the default value of the argument.

Returns

A `ArrayList<String>` representing the default values

#### 6.1.2.3 `ArrayList<String> gestion_options.Argument.getPassedValues ( )`

Get the given value of the argument.

Returns

A `ArrayList<String>` representing the passed values

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

- `src/gestion_options/Argument.java`

## 6.2 exceptions.ArgumentAlreadyUsedException Class Reference

Inheritance diagram for `exceptions.ArgumentAlreadyUsedException`:



## Public Member Functions

- [ArgumentAlreadyUsedException](#) ()  
*Default constructor.*
- [ArgumentAlreadyUsedException](#) (String argument)  
*Same as the default constructor but specify the argument already given.*

### 6.2.1 Constructor & Destructor Documentation

#### 6.2.1.1 exceptions.ArgumentAlreadyUsedException.ArgumentAlreadyUsedException ( )

Default constructor.

Used to raise the exception

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

- src/exceptions/[ArgumentAlreadyUsedException.java](#)

## 6.3 gestion\_options.BaseArgument Class Reference

Inheritance diagram for gestion\_options.BaseArgument:

## Public Member Functions

- [BaseArgument](#) (String name, boolean allowMultipleValues)  
*Constructor to create the mandatory argument.*
- [BaseArgument](#) (String name, boolean allowMultipleValues, String[] defaultValues)  
*Constructor to create the mandatory argument.*
- String [getName](#) ()  
*Get the name of the argument.*
- boolean [getIsMandatory](#) ()  
*Get the if the argument is mandatory.*
- boolean [getAllowMultipleValues](#) ()  
*Get if the argument allow multiple values.*
- boolean [getIsUsed](#) ()  
*Tell if the argument is used.*
- void [setIsUsed](#) (boolean value)  
*Set if the argument is used.*
- ArrayList< String > [getDefaultValues](#) ()  
*Get the default value of the argument.*
- ArrayList< String > [getFinalValues](#) ()  
*Get the final value of the argument.*
- abstract boolean [checkPassedValues](#) (ArrayList< String > values) throws InvalidValueException, TooManyValuesException  
*Used to check if the values are right for the argument.*
- String [toString](#) ()  
*Give the argument and it value as a String.*
- String [getFinalValuesString](#) ()

## Protected Member Functions

- void [addFinalValues](#) (List< String > values)  
*Add value to the argument.*

### 6.3.1 Constructor & Destructor Documentation

#### 6.3.1.1 `gestion_options.BaseArgument.BaseArgument ( String name, boolean allowMultipleValues )`

Constructor to create the mandatory argument.

Initialise the argument with it name and if it allow multiple value

Parameters

|                                  |                                                             |
|----------------------------------|-------------------------------------------------------------|
| <i>name</i>                      | Name of the argument                                        |
| <i>allowMultiple↔<br/>Values</i> | <i>boolean</i> representing if you can pass multiple values |

#### 6.3.1.2 `gestion_options.BaseArgument.BaseArgument ( String name, boolean allowMultipleValues, String[] defaultValues )`

Constructor to create the mandatory argument.

Initialise the argument with it name and if it allow multiple value

Parameters

|                                  |                                                             |
|----------------------------------|-------------------------------------------------------------|
| <i>name</i>                      | Name of the argument                                        |
| <i>allowMultiple↔<br/>Values</i> | <i>boolean</i> representing if you can pass multiple values |
| <i>defaultValues</i>             | Represent the default values of the argument                |

### 6.3.2 Member Function Documentation

#### 6.3.2.1 `void gestion_options.BaseArgument.addFinalValues ( List< String > values ) [protected]`

Add value to the argument.

Parameters

|               |                               |
|---------------|-------------------------------|
| <i>values</i> | values to aad to the argument |
|---------------|-------------------------------|

#### 6.3.2.2 `abstract boolean gestion_options.BaseArgument.checkPassedValues ( ArrayList< String > values ) throws InvalidValueException, TooManyValuesException [abstract]`

Used to check if the values are right for the argument.

Parameters

|                     |                 |
|---------------------|-----------------|
| <i>passedValues</i> | values to check |
|---------------------|-----------------|

#### 6.3.2.3 `boolean gestion_options.BaseArgument.getAllowMultipleValues ( )`

Get if the argument allow multiple values.

**Returns**

A *boolean* telling if you can pass multiple values

**6.3.2.4 `ArrayList<String> gestion_options.BaseArgument.getDefaultValues ( )`**

Get the default value of the argument.

**Returns**

A *ArrayList<String>* representing the default values

**6.3.2.5 `ArrayList<String> gestion_options.BaseArgument.getFinalValues ( )`**

Get the final value of the argument.

**Returns**

A *ArrayList<String>* representing the final values

**6.3.2.6 `boolean gestion_options.BaseArgument.getIsMandatory ( )`**

Get the if the argument is mandatory.

**Returns**

A *boolean* representing the importance of the argument

**6.3.2.7 `boolean gestion_options.BaseArgument.getIsUsed ( )`**

Tell if the argument is used.

**Returns**

A *boolean* representing if the argument is used

**6.3.2.8 `String gestion_options.BaseArgument.getName ( )`**

Get the name of the argument.

**Returns**

A *String* representing the name of the argument

**6.3.2.9 `void gestion_options.BaseArgument.setIsUsed ( boolean value )`**

Set if the argument is used.

## Parameters

|              |                |
|--------------|----------------|
| <i>value</i> | value of using |
|--------------|----------------|

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

- [src/gestion\\_options/BaseArgument.java](#)

## 6.4 LSB.WayStrategy.BHider Class Reference

Inheritance diagram for LSB.WayStrategy.BHider:

### Public Member Functions

- void [hide](#) ([ImageMatrix](#) imageMatrix, int i, int j, int mask, String msg)  
*Hide information in a pixel and blue channel.*

### 6.4.1 Member Function Documentation

6.4.1.1 void [LSB.WayStrategy.BHider.hide](#) ( [ImageMatrix](#) *imageMatrix*, int *i*, int *j*, int *mask*, String *msg* )

Hide information in a pixel and blue channel.

## Parameters

|                    |                                                  |
|--------------------|--------------------------------------------------|
| <i>imageMatrix</i> | The matrix to use                                |
| <i>i</i>           | x position of the pixel                          |
| <i>j</i>           | y position of the pixel                          |
| <i>mask</i>        | Represent the mask to applicate before each hide |
| <i>msg</i>         | The message to hide                              |
| <i>hider</i>       | Represent in which channel we are going to hide  |

Implements [LSB.WayStrategy.Hider](#).

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

- [src/LSB/WayStrategy.java](#)

## 6.5 exceptions.ChannelException Class Reference

Inheritance diagram for exceptions.ChannelException:

### Public Member Functions

- [ChannelException](#) ()  
*Default constructor.*

### 6.5.1 Constructor & Destructor Documentation

#### 6.5.1.1 exceptions.ChannelException.ChannelException ( )

Default constructor.

Used to raise the exception

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

- src/exceptions/[ChannelException.java](#)

## 6.6 img\_analyses.Channels Enum Reference

### Public Attributes

- RED
- GREEN
- BLUE
- ALPHA
- GREY

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

- src/img\_analyses/[Channels.java](#)

## 6.7 LSB.DirectWay Class Reference

Inheritance diagram for LSB.DirectWay:

### Protected Member Functions

- int [Browse](#) ([ImageMatrix](#) imageMatrix, int nbBytes, int begin, String msg, int mask, [Hider](#) hider)  
*Run the pattern algorithm and apply the LSB.*

### Additional Inherited Members

#### 6.7.1 Member Function Documentation

##### 6.7.1.1 int LSB.DirectWay.Browse ( [ImageMatrix](#) imageMatrix, int nbBytes, int begin, String msg, int mask, [Hider](#) hider ) [protected]

Run the pattern algorithm and apply the LSB.

#### Parameters

|                    |                   |
|--------------------|-------------------|
| <i>imageMatrix</i> | The matrix to use |
|--------------------|-------------------|

|                |                                                  |
|----------------|--------------------------------------------------|
| <i>nbBytes</i> | number of bits in each pixel                     |
| <i>begin</i>   | Where to begin to hide in the message            |
| <i>msg</i>     | The message to hide                              |
| <i>mask</i>    | Represent the mask to applicate before each hide |
| <i>hider</i>   | represent in which channel we are going to hide  |

#### Returns

An *int* representing the remaining part of the message to hide

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

- src/LSB/[DirectWay.java](#)

## 6.8 LSB.ExternSpiralWay Class Reference

Inheritance diagram for LSB.ExternSpiralWay:

#### Protected Member Functions

- `int Browse (ImageMatrix imageMatrix, int nbBytes, int begin, String msg, int mask, Hider hider)`  
Run the pattern algorithm and apply the LSB.

#### Additional Inherited Members

#### 6.8.1 Member Function Documentation

6.8.1.1 `int LSB.ExternSpiralWay.Browse ( ImageMatrix imageMatrix, int nbBytes, int begin, String msg, int mask, Hider hider )` [protected]

Run the pattern algorithm and apply the LSB.

#### Parameters

|                    |                                                  |
|--------------------|--------------------------------------------------|
| <i>imageMatrix</i> | The matrix to use                                |
| <i>nbBytes</i>     | number of bits in each pixel                     |
| <i>begin</i>       | Where to begin to hide in the message            |
| <i>msg</i>         | The message to hide                              |
| <i>mask</i>        | Represent the mask to applicate before each hide |
| <i>hider</i>       | represent in which channel we are going to hide  |

#### Returns

An *int* representing the remaining part of the message to hide

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

- src/LSB/ExternSpiralWay.java

## 6.9 LSB.WayStrategy.GHider Class Reference

Inheritance diagram for LSB.WayStrategy.GHider:

### Public Member Functions

- void [hide](#) ([ImageMatrix](#) imageMatrix, int i, int j, int mask, String msg)  
*Hide information in a pixel and green channel.*

### 6.9.1 Member Function Documentation

#### 6.9.1.1 void LSB.WayStrategy.GHider.hide ( [ImageMatrix](#) imageMatrix, int i, int j, int mask, String msg )

Hide information in a pixel and green channel.

##### Parameters

|                    |                                                  |
|--------------------|--------------------------------------------------|
| <i>imageMatrix</i> | The matrix to use                                |
| <i>i</i>           | x position of the pixel                          |
| <i>j</i>           | y position of the pixel                          |
| <i>mask</i>        | Represent the mask to applicate before each hide |
| <i>msg</i>         | The message to hide                              |
| <i>hider</i>       | Represent in which channel we are going to hide  |

Implements [LSB.WayStrategy.Hider](#).

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

- src/LSB/WayStrategy.java

## 6.10 LSB.WayStrategy.Hider Interface Reference

Inheritance diagram for LSB.WayStrategy.Hider:

### Public Member Functions

- void [hide](#) ([ImageMatrix](#) imageMatrix, int i, int j, int mask, String msg)  
*Hide information in a pixel and a channel.*

### 6.10.1 Member Function Documentation

#### 6.10.1.1 void LSB.WayStrategy.Hider.hide ( [ImageMatrix](#) imageMatrix, int i, int j, int mask, String msg )

Hide information in a pixel and a channel.

## Parameters

|                    |                                                  |
|--------------------|--------------------------------------------------|
| <i>imageMatrix</i> | The matrix to use                                |
| <i>i</i>           | x position of the pixel                          |
| <i>j</i>           | y position of the pixel                          |
| <i>mask</i>        | Represent the mask to applicate before each hide |
| <i>msg</i>         | The message to hide                              |
| <i>hider</i>       | Represent in which channel we are going to hide  |

Implemented in [LSB.WayStrategy.BHider](#), [LSB.WayStrategy.GHider](#), and [LSB.WayStrategy.RHider](#).

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

- `src/LSB/WayStrategy.java`

## 6.11 img\_analyses.ImageMatrix Class Reference

### Public Member Functions

- [ImageMatrix](#) (String imagePath) throws IOException  
*Constructor of the matrix.*
- boolean [checkSize](#) (int msgSize, int nbChannel, int nbBits, int nbPixel)  
*Check if the image is large enough to hide the message.*
- int [getNbPixelsImpacted](#) ()  
*Get the number of impacted pixels.*
- ArrayList< Integer > [getNbChannelsImpacted](#) ()  
*Get the number of channels impacted.*
- boolean [isGreyscale](#) ()  
*Get the image is in greyscale.*
- [WayStrategy](#) [getWay](#) ()  
*Get the pattern algorithm used to hide the image.*
- void [setWay](#) ([WayStrategy](#) way)  
*Set the pattern algorithm used to hide the image.*
- Color [getColor](#) (int i, int j)  
*Get a pixel.*
- int [getHeight](#) ()  
*Get the height of the image.*
- int [getWidth](#) ()  
*Get the width of the image.*
- boolean [getAsAlpha](#) ()  
*Get if the image as an alpha channel.*
- List< List< Color > > [getImage](#) ()  
*Get the matrix of the image.*
- int [getSize](#) ()  
*Get the number of pixel of the image.*
- boolean [LSB](#) (String msg, String magicWord, int bitPerPixel, [Channels](#)[] chan, [Ways](#) w) throws Channel↵Exception, ImageTooShort  
*Apply the lsb to the matrix.*
- void [saveImage](#) (String name, String type)  
*Save the image.*



## 6.11.1 Constructor & Destructor Documentation

### 6.11.1.1 `img_analyses.ImageMatrix.ImageMatrix ( String imagePath )` throws `IOException`

Constructor of the matrix.

## Parameters

|                  |  |
|------------------|--|
| <i>imagePath</i> |  |
|------------------|--|

Build the matrix, and take some information about the image

## Parameters

|                  |                             |
|------------------|-----------------------------|
| <i>imagePath</i> | The path of the given image |
|------------------|-----------------------------|

## 6.11.2 Member Function Documentation

### 6.11.2.1 boolean `img_analyses.ImageMatrix.checkSize ( int msgSize, int nbChannel, int nbBits, int nbPixel )`

Check if the image is large enough to hide the message.

## Parameters

|                  |                                                |
|------------------|------------------------------------------------|
| <i>msgSize</i>   | Size of the message                            |
| <i>nbChannel</i> | Number of channel used to hide the information |
| <i>nbBits</i>    | Number of bit used per pixel                   |
| <i>nbPixel</i>   | Number of pixel in the image                   |

### 6.11.2.2 boolean `img_analyses.ImageMatrix.getAsAlpha ( )`

Get if the image as an alpha channel.

## Returns

A *boolean* representing the presence of an alpha channel

### 6.11.2.3 Color `img_analyses.ImageMatrix.getColor ( int i, int j )`

Get a pixel.

## Parameters

|          |  |
|----------|--|
| <i>i</i> |  |
|----------|--|

## Returns

Color at the index asked  
A *Color* representing a pixel

### 6.11.2.4 int `img_analyses.ImageMatrix.getHeight ( )`

Get the height of the image.

## Returns

An *int* representing the height of the image

### 6.11.2.5 List<List<Color> > `img_analyses.ImageMatrix.getImage ( )`

Get the matrix of the image.

**Returns**

Matrix of color, representing the image  
A *List<List<Color>>* representing the matrix of the image

**6.11.2.6   *ArrayList<Integer>* img\_analyses.ImageMatrix.getNbChannelsImpacted ( )**

Get the number of channels impacted.

**Returns**

A *ArrayList<Integer>* representing the number of channels impacted

**6.11.2.7   *int* img\_analyses.ImageMatrix.getNbPixelsImpacted ( )**

Get the number of impacted pixels.

**Returns**

An *int* representing the number of impacted pixels

**6.11.2.8   *int* img\_analyses.ImageMatrix.getSize ( )**

Get the number of pixel of the image.

**Returns**

A *int* representing the number of pixel of the image

**6.11.2.9   *WayStrategy* img\_analyses.ImageMatrix.getWay ( )**

Get the pattern algorithm used to hide the image.

**Returns**

A *WayStrategy* representing the pattern used to hide the image

**6.11.2.10   *int* img\_analyses.ImageMatrix.getWidth ( )**

Get the width of the image.

**Returns**

An *int* representing the width of the image

**6.11.2.11   *boolean* img\_analyses.ImageMatrix.isGreyscale ( )**

Get the image is in greyscale.

**Returns**

A *boolean* representing if the image is in greyscale

6.11.2.12 `boolean img_analyses.ImageMatrix.LSB ( String msg, String magicWord, int bitPerPixel, Channels[] chan, Ways w )` throws `ChannelException`, `ImageTooShort`

Apply the lsb to the matrix.

## Parameters

|                    |                                       |
|--------------------|---------------------------------------|
| <i>msg</i>         | The message to hide                   |
| <i>magicWord</i>   | The magic word to use                 |
| <i>bitPerPixel</i> | Number of bit used in each pixel      |
| <i>chan</i>        | <a href="#">Channels</a> used to hide |
| <i>way</i>         | The way through the image             |

## Returns

A *boolean* representing the success of the operation

## 6.11.2.13 void img\_analyses.ImageMatrix.saveImage ( String name, String type )

Save the image.

## Parameters

|             |                          |
|-------------|--------------------------|
| <i>name</i> | Name of the saved file   |
| <i>type</i> | Format of the saved file |

## 6.11.2.14 void img\_analyses.ImageMatrix.setWay ( WayStrategy way )

Set the pattern algorithm used to hide the image.

## Returns

ways Represent the pattern to set

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

- src/img\_analyses/[ImageMatrix.java](#)

## 6.12 exceptions.ImageTooShort Class Reference

Inheritance diagram for exceptions.ImageTooShort:

### Public Member Functions

- [ImageTooShort](#) ()  
*Default constructor.*

### 6.12.1 Constructor & Destructor Documentation

## 6.12.1.1 exceptions.ImageTooShort.ImageTooShort ( )

Default constructor.

Used to raise the exception

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

- src/exceptions/[ImageTooShort.java](#)

## 6.13 LSB.InternSpiralWay Class Reference

Inheritance diagram for LSB.InternSpiralWay:

### Protected Member Functions

- `int Browse (ImageMatrix imageMatrix, int nbBytes, int begin, String msg, int mask, Hider hider)`  
*Run the pattern algorithm and apply the LSB.*

### Additional Inherited Members

#### 6.13.1 Member Function Documentation

6.13.1.1 `int LSB.InternSpiralWay.Browse ( ImageMatrix imageMatrix, int nbBytes, int begin, String msg, int mask, Hider hider )` [protected]

Run the pattern algorithm and apply the LSB.

#### Parameters

|                    |                                                  |
|--------------------|--------------------------------------------------|
| <i>imageMatrix</i> | The matrix to use                                |
| <i>nbBytes</i>     | number of bits in each pixel                     |
| <i>begin</i>       | Where to begin to hide in the message            |
| <i>msg</i>         | The message to hide                              |
| <i>mask</i>        | Represent the mask to applicate before each hide |
| <i>hider</i>       | represent in which channel we are going to hide  |

#### Returns

An *int* representing the remaining part of the message to hide

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

- `src/LSB/InternSpiralWay.java`

## 6.14 exceptions.InvalidArgException Class Reference

Inheritance diagram for exceptions.InvalidArgException:

### Public Member Functions

- `InvalidArgException ()`  
*Default constructor.*
- `InvalidArgException (String argument)`  
*Same as the default constructor, but specify which argument is invalid.*

### 6.14.1 Constructor & Destructor Documentation

#### 6.14.1.1 exceptions.InvalidArgException.InvalidArgException ( )

Default constructor.

Used to raise the exception

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

- src/exceptions/[InvalidArgException.java](#)

## 6.15 exceptions.InvalidValueException Class Reference

Inheritance diagram for exceptions.InvalidValueException:

### Public Member Functions

- [InvalidValueException](#) ()  
*Default constructor.*
- [InvalidValueException](#) (String value)  
*Same as the default constructor but specify the invalid value.*

### 6.15.1 Constructor & Destructor Documentation

#### 6.15.1.1 exceptions.InvalidValueException.InvalidValueException ( )

Default constructor.

Used to raise the exception

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

- src/exceptions/[InvalidValueException.java](#)

## 6.16 Launch Class Reference

### Static Public Member Functions

- static void **main** (String[] args) throws ImageTooShort, ChannelException, IOException, InvalidArgException, MessageContainsMagicNumberException

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

- src/Launch.java

## 6.17 exceptions.MandatoryArgumentNotDefinedException Class Reference

Inheritance diagram for exceptions.MandatoryArgumentNotDefinedException:

## Public Member Functions

- [MandatoryArgumentNotDefinedException](#) ()  
*Default constructor.*
- [MandatoryArgumentNotDefinedException](#) (String argument)  
*Same as the default constructor but specify the missing argument.*

### 6.17.1 Constructor & Destructor Documentation

#### 6.17.1.1 exceptions.MandatoryArgumentNotDefinedException.MandatoryArgumentNotDefinedException ( )

Default constructor.

Used to raise the exception

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

- src/exceptions/MandatoryArgumentNotDefinedException.java

## 6.18 message.Message Class Reference

### Public Member Functions

- [Message](#) (String [Message](#), String MagicWord, int bitPerPixel)  
*Constructor of the message to make it ready to hide.*
- String [getFullMessage](#) ()  
*Give back the message + the magic word.*
- String [getBinString](#) ()  
*Give back the message + the magic word in binary.*

### 6.18.1 Constructor & Destructor Documentation

#### 6.18.1.1 message.Message.Message ( String Message, String MagicWord, int bitPerPixel )

Constructor of the message to make it ready to hide.

Parameters

|                         |                              |
|-------------------------|------------------------------|
| <a href="#">Message</a> | The message to hide          |
| <i>MagicWord</i>        | The used magic word          |
| <i>bitPerPixel</i>      | Number of bit used per pixel |

### 6.18.2 Member Function Documentation

#### 6.18.2.1 String message.Message.getBinString ( )

Give back the message + the magic word in binary.

Returns

A *String* representing the full binary message



### 6.18.2.2 String message.Message.getFullMessage ( )

Give back the message + the magic word.

#### Returns

A *String* representing the full message

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

- src/message/[Message.java](#)

## 6.19 exceptions.MessageContainsMagicNumberException Class Reference

Inheritance diagram for exceptions.MessageContainsMagicNumberException:

### Public Member Functions

- [MessageContainsMagicNumberException](#) ()  
*Default constructor.*

### 6.19.1 Constructor & Destructor Documentation

#### 6.19.1.1 exceptions.MessageContainsMagicNumberException.MessageContainsMagicNumberException ( )

Default constructor.

Used to raise the exception

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

- src/exceptions/[MessageContainsMagicNumberException.java](#)

## 6.20 gestion\_options.ProgramOptions Class Reference

### Public Member Functions

- String [getInputFormat](#) ()  
*Get the format of the input file.*
- void [setInputFormat](#) (String value)  
*Set the format of the input file.*
- String [getOutputFormat](#) ()  
*Get the format of the output file.*
- void [setOutputFormat](#) (String value)  
*Set the format of the output file.*
- String [getInputFile](#) ()  
*Get the path/name of the input file.*
- void [setInputFile](#) (String value)  
*Set the path/name of the input file.*
- String [getOutputFile](#) ()

- Get the path/name of the output file.*

  - void [setOutputFile](#) (String value)

*Set the path/name of the output file.*
- String [getMessage](#) ()

*Get the message to hide.*
- void [setMessage](#) (String value)

*Set the message to hide.*
- [Channels](#)[] [getChannels](#) ()

*Get the channel(s) used to hide the message.*
- void [setChannels](#) (String value) throws `InvalidArgException`

*Set the channel(s) used to hide the message.*
- [Ways](#) [getPattern](#) ()

*Get the pattern used to go through the image matrix.*
- void [setPattern](#) (String value) throws `InvalidArgException`

*Set the pattern used to go through the image matrix.*
- String [getNumberBits](#) ()

*Get the number of bits used to hide the message in each pixel for each channel.*
- void [setNumberBits](#) (String value)

*Set the number of bits used to hide the message in each pixel for each channel.*
- String [getMagicNumber](#) ()

*Get the magic number.*
- void [setMagicNumber](#) (String value)

*Set the magic number.*
- String [getMetrics](#) ()

*Get the metrics.*
- void [setMetrics](#) (String value)

*Set the metrics.*
- String [toString](#) ()

*Give the program options and their values as a String.*

### 6.20.1 Detailed Description

Created by Loic Rose on 11/17/2015.

### 6.20.2 Member Function Documentation

#### 6.20.2.1 [Channels](#)[] [gestion\\_options.ProgramOptions.getChannels](#) ( )

Get the channel(s) used to hide the message.

##### Returns

A [Channels](#)[] representing the channel(s) used to hide the message

#### 6.20.2.2 String [gestion\\_options.ProgramOptions.getInputFile](#) ( )

Get the path/name of the input file.

##### Returns

A *String* representing the path/name of the input file

**6.20.2.3 String gestion\_options.ProgramOptions.getInputFormat ( )**

Get the format of the input file.

Returns

A *String* representing the format of the input file

**6.20.2.4 String gestion\_options.ProgramOptions.getMagicNumber ( )**

Get the magic number.

Returns

A *String* representing the magic number

**6.20.2.5 String gestion\_options.ProgramOptions.getMessage ( )**

Get the message to hide.

Returns

A *String* representing the message to hide

**6.20.2.6 String gestion\_options.ProgramOptions.getMetrics ( )**

Get the metrics.

Returns

A *String* representing the metrics

**6.20.2.7 String gestion\_options.ProgramOptions.getNumberBits ( )**

Get the number of bits used to hide the message in each pixel for each channel.

Returns

A *String* representing the number of bits used to hide the message in each pixel for each channel

**6.20.2.8 String gestion\_options.ProgramOptions.getOutputFile ( )**

Get the path/name of the output file.

Returns

A *String* representing the path/name of the output file

**6.20.2.9 String gestion\_options.ProgramOptions.getOutputFormat ( )**

Get the format of the output file.

Returns

A *String* representing the format of the output file

**6.20.2.10    Ways   `gestion_options.ProgramOptions.getPattern ( )`**

Get the pattern used to go through the image matrix.

**Returns**

A Ways the pattern used to go through the image matrix

**6.20.2.11    void   `gestion_options.ProgramOptions.setChannels ( String value )` throws `InvalidArgException`**

Set the channel(s) used to hide the message.

**Parameters**

|              |                                                   |
|--------------|---------------------------------------------------|
| <i>value</i> | Represent the channel(s) used to hide the message |
|--------------|---------------------------------------------------|

**6.20.2.12    void   `gestion_options.ProgramOptions.setInputFile ( String value )`**

Set the path/name of the input file.

**Parameters**

|              |                                           |
|--------------|-------------------------------------------|
| <i>value</i> | Represent the path/name of the input file |
|--------------|-------------------------------------------|

**6.20.2.13    void   `gestion_options.ProgramOptions.setInputFormat ( String value )`**

Set the format of the input file.

**Parameters**

|              |                                        |
|--------------|----------------------------------------|
| <i>value</i> | Represent the format of the input file |
|--------------|----------------------------------------|

**6.20.2.14    void   `gestion_options.ProgramOptions.setMagicNumber ( String value )`**

Set the magic number.

**Parameters**

|              |                            |
|--------------|----------------------------|
| <i>value</i> | represent the magic number |
|--------------|----------------------------|

**6.20.2.15    void   `gestion_options.ProgramOptions.setMessage ( String value )`**

Set the message to hide.

**Parameters**

|              |                               |
|--------------|-------------------------------|
| <i>value</i> | Represent the message to hide |
|--------------|-------------------------------|

**6.20.2.16    void   `gestion_options.ProgramOptions.setMetrics ( String value )`**

Set the metrics.

## Parameters

|              |                       |
|--------------|-----------------------|
| <i>value</i> | Represent the metrics |
|--------------|-----------------------|

6.20.2.17 void `gestion_options.ProgramOptions.setNumberBits ( String value )`

Set the number of bits used to hide the message in each pixel for each channel.

## Parameters

|              |                                                                                      |
|--------------|--------------------------------------------------------------------------------------|
| <i>value</i> | Represent the number of bits used to hide the message in each pixel for each channel |
|--------------|--------------------------------------------------------------------------------------|

6.20.2.18 void `gestion_options.ProgramOptions.setOutputFile ( String value )`

Set the path/name of the output file.

## Parameters

|              |                                            |
|--------------|--------------------------------------------|
| <i>value</i> | Represent the path/name of the output file |
|--------------|--------------------------------------------|

6.20.2.19 void `gestion_options.ProgramOptions.setOutputFormat ( String value )`

Set the format of the output file.

## Parameters

|              |                                         |
|--------------|-----------------------------------------|
| <i>value</i> | Represent the format of the output file |
|--------------|-----------------------------------------|

6.20.2.20 void `gestion_options.ProgramOptions.setPattern ( String value )` throws `InvalidArgException`

Set the pattern used to go through the image matrix.

## Parameters

|              |                                                           |
|--------------|-----------------------------------------------------------|
| <i>value</i> | represent the pattern used to go through the image matrix |
|--------------|-----------------------------------------------------------|

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

- `src/gestion_options/ProgramOptions.java`

## 6.21 LSB.RevertWay Class Reference

Inheritance diagram for LSB.RevertWay:

### Protected Member Functions

- `int Browse (ImageMatrix imageMatrix, int nbBytes, int begin, String msg, int mask, Hider hider)`

*Run the pattern algorithm and apply the LSB.*

## Additional Inherited Members

### 6.21.1 Member Function Documentation

6.21.1.1 `int LSB.RevertWay.Browse ( ImageMatrix imageMatrix, int nbBytes, int begin, String msg, int mask, Hider hider )`  
[protected]

Run the pattern algorithm and apply the LSB.

#### Parameters

|                    |                                                  |
|--------------------|--------------------------------------------------|
| <i>imageMatrix</i> | The matrix to use                                |
| <i>nbBytes</i>     | number of bits in each pixel                     |
| <i>begin</i>       | Where to begin to hide in the message            |
| <i>msg</i>         | The message to hide                              |
| <i>mask</i>        | Represent the mask to applicate before each hide |
| <i>hider</i>       | represent in which channel we are going to hide  |

#### Returns

An *int* representing the remaining part of the message to hide

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

- `src/LSB/RevertWay.java`

## 6.22 LSB.WayStrategy.RHider Class Reference

Inheritance diagram for `LSB.WayStrategy.RHider`:

### Public Member Functions

- void `hide (ImageMatrix imageMatrix, int i, int j, int mask, String msg)`  
*Hide information in a pixel and red channel.*

### 6.22.1 Member Function Documentation

6.22.1.1 `void LSB.WayStrategy.RHider.hide ( ImageMatrix imageMatrix, int i, int j, int mask, String msg )`

Hide information in a pixel and red channel.

#### Parameters

|                    |                                                  |
|--------------------|--------------------------------------------------|
| <i>imageMatrix</i> | The matrix to use                                |
| <i>i</i>           | x position of the pixel                          |
| <i>j</i>           | y position of the pixel                          |
| <i>mask</i>        | Represent the mask to applicate before each hide |
| <i>msg</i>         | The message to hide                              |

|                       |                                                 |
|-----------------------|-------------------------------------------------|
| <a href="#">hider</a> | Represent in which channel we are going to hide |
|-----------------------|-------------------------------------------------|

Implements [LSB.WayStrategy.Hider](#).

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

- [src/LSB/WayStrategy.java](#)

## 6.23 exceptions.[TooManyValuesException](#) Class Reference

Inheritance diagram for exceptions.[TooManyValuesException](#):

### Public Member Functions

- [TooManyValuesException](#) ()  
*Default constructor.*
- [TooManyValuesException](#) (String arg)  
*Same as the default constructor but specify which argument has too many values.*

### 6.23.1 Constructor & Destructor Documentation

#### 6.23.1.1 exceptions.[TooManyValuesException](#).[TooManyValuesException](#) ( )

Default constructor.

Used to raise the exception

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

- [src/exceptions/\[TooManyValuesException.java\]\(#\)](#)

## 6.24 action.[Ways](#) Enum Reference

### Public Attributes

- **DIRECT**
- **REVERSE**
- **INTSPI**
- **EXTSPI**

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

- [src/action/\[Ways.java\]\(#\)](#)

## 6.25 [LSB.WayStrategy](#) Class Reference

Inheritance diagram for [LSB.WayStrategy](#):

## Classes

- class [BHider](#)
- class [GHider](#)
- interface [Hider](#)
- class [RHider](#)

## Public Member Functions

- boolean [LSB](#) ([ImageMatrix](#) imageMatrix, int nbBytes, [Channels](#)[] channels, String message) throws [ChannelException](#)  
*Apply the LSB.*

## Protected Member Functions

- abstract int [Browse](#) ([ImageMatrix](#) imageMatrix, int nbBytes, int begin, String msg, int mask, [Hider](#) hider)  
*Run the pattern algorithm and apply the LSB.*

### 6.25.1 Detailed Description

Created by aelar on 02/12/15.

### 6.25.2 Member Function Documentation

6.25.2.1 abstract int [LSB.WayStrategy.Browse](#) ( [ImageMatrix](#) *imageMatrix*, int *nbBytes*, int *begin*, String *msg*, int *mask*, [Hider](#) *hider* ) [abstract], [protected]

Run the pattern algorithm and apply the LSB.

#### Parameters

|                    |                                                  |
|--------------------|--------------------------------------------------|
| <i>imageMatrix</i> | The matrix to use                                |
| <i>nbBytes</i>     | number of bits in each pixel                     |
| <i>begin</i>       | Where to begin to hide in the message            |
| <i>msg</i>         | The message to hide                              |
| <i>mask</i>        | Represent the mask to applicate before each hide |
| <i>hider</i>       | represent in which channel we are going to hide  |

#### Returns

An *int* representing the remaining part of the message to hide

6.25.2.2 boolean [LSB.WayStrategy.LSB](#) ( [ImageMatrix](#) *imageMatrix*, int *nbBytes*, [Channels](#)[] *channels*, String *message* ) throws [ChannelException](#)

Apply the LSB.

#### Parameters

|                    |                   |
|--------------------|-------------------|
| <i>imageMatrix</i> | The matrix to use |
|--------------------|-------------------|



|                 |                              |
|-----------------|------------------------------|
| <i>nbBytes</i>  | number of bits in each pixel |
| <i>channels</i> | The channels used to hide    |
| <i>message</i>  | The message to hide          |

**Returns**

A *boolean* representing the remaining part of the message to hide

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

- src/LSB/WayStrategy.java



## Chapter 7

# File Documentation

### 7.1 `src/action/Ways.java` File Reference

Define the different pattern to go through the image.

#### Classes

- enum [action.Ways](#)

#### 7.1.1 Detailed Description

Define the different pattern to go through the image.

Created by aelar on 02/12/15.

#### Author

Benjamin PAIT

#### Version

1.0

#### Date

02 December 2015

### 7.2 `src/exceptions/ArgumentAlreadyUsedException.java` File Reference

Exception which is raised when an argument is specified more than one time.

#### Classes

- class [exceptions.ArgumentAlreadyUsedException](#)

### 7.2.1 Detailed Description

Exception which is raised when an argument is specified more than one time.

Created by Loc on 11/15/2015.

**Author**

Loïc ROSE

**Version**

1.0

**Date**

15 November 2015

## 7.3 `src/exceptions/ChannelException.java` File Reference

Exception which is raised when a specified channel is wrong.

**Classes**

- class [exceptions.ChannelException](#)

### 7.3.1 Detailed Description

Exception which is raised when a specified channel is wrong.

Created by aelar on 16/11/15.

**Author**

Benjamin PIAT

**Version**

1.0

**Date**

16 November 2015

## 7.4 `src/exceptions/ImageTooShort.java` File Reference

Exception which is raised when the given image is too short to hide the message.

**Classes**

- class [exceptions.ImageTooShort](#)

### 7.4.1 Detailed Description

Exception which is raised when the given image is too short to hide the message.

Created by aelar on 12/12/15.

**Author**

Benjamin PIAT

**Version**

1.0

**Date**

12 December 2015

## 7.5 src/exceptions/InvalidArgException.java File Reference

Exception which is raised when an argument is invalid.

**Classes**

- class [exceptions.InvalidArgException](#)

### 7.5.1 Detailed Description

Exception which is raised when an argument is invalid.

Created by Loc on 11/15/2015.

**Author**

Loïc ROSE

**Version**

1.0

**Date**

15 November 2015

## 7.6 src/exceptions/InvalidValueException.java File Reference

Exception which is raised when a value passed is wrong.

**Classes**

- class [exceptions.InvalidValueException](#)

### 7.6.1 Detailed Description

Exception which is raised when a value passed is wrong.

Created by Loc on 11/15/2015.

**Author**

Loïc ROSE

**Version**

1.0

**Date**

15 November 2015

## 7.7 `src/exceptions/MessageContainsMagicNumberException.java` File Reference

Exception which is raised when the message contain the magic number.

**Classes**

- class [exceptions.MessageContainsMagicNumberException](#)

### 7.7.1 Detailed Description

Exception which is raised when the message contain the magic number.

Created by Kevin on 09/01/2016.

**Author**

Kevin DUGLUE

**Version**

1.0

**Date**

09 January 2016

## 7.8 `src/exceptions/TooManyValuesException.java` File Reference

Exception which is raised when an argument has too many passed values.

**Classes**

- class [exceptions.TooManyValuesException](#)

### 7.8.1 Detailed Description

Exception which is raised when an argument has too many passed values.

Created by Loïc on 11/15/2015.

**Author**

Loïc ROSE

**Version**

1.0

**Date**

15 November 2015

## 7.9 src/gestion\_options/Argument.java File Reference

Represent a passed argument but more specifically than it super class.

**Classes**

- class [gestion\\_options.Argument](#)

**Packages**

- package [gestion\\_options](#)

### 7.9.1 Detailed Description

Represent a passed argument but more specifically than it super class.

**Author**

Loïc ROSE

**Version**

1.0

**Date**

07 November 2016

Specify if the argument is optional, if it accept multiple values, and if values needs a validation

## 7.10 src/gestion\_options/BaseArgument.java File Reference

Represent a passed argument.

## Classes

- class [gestion\\_options.BaseArgument](#)

## Packages

- package [gestion\\_options](#)

### 7.10.1 Detailed Description

Represent a passed argument.

#### Author

Loïc ROSE

#### Version

1.0

#### Date

07 November 2016

## 7.11 [src/img\\_analyses/Channels.java](#) File Reference

Define all usable channels to hide information.

## Classes

- enum [img\\_analyses.Channels](#)

### 7.11.1 Detailed Description

Define all usable channels to hide information.

Created by aelar on 02/12/15.

#### Author

Benjamin PIAT

#### Version

1.0

#### Date

02 December 2015

## 7.12 [src/img\\_analyses/ImageMatrix.java](#) File Reference

Represent the image as a matrix with the different action that you can perform on it.



## Classes

- class [img\\_analyses.ImageMatrix](#)

### 7.12.1 Detailed Description

Represent the image as a matrix with the different action that you can perform on it.

Created by aelar on 10/11/15.

#### Author

Benjamin PIAT

#### Version

1.0

#### Date

10 November 2015

## 7.13 src/LSB/DirectWay.java File Reference

The direct pattern.

## Classes

- class [LSB.DirectWay](#)

### 7.13.1 Detailed Description

The direct pattern.

The revert pattern.

The intern spiral pattern.

The extern spiral pattern.

Created by aelar on 02/12/15.

#### Author

Benjamin PIAT

#### Version

1.0

#### Date

02 December 2015

Go through the image from top left to bottom right

Created by aelar on 05/12/15.

**Author**

Benjamin PIAT

**Version**

1.0

**Date**

02 December 2015

Go through the image from top left in a spiral to the center

Created by aelar on 08/12/15.

**Author**

Benjamin PIAT

**Version**

1.0

**Date**

02 December 2015

Go through the image from the center in a spiral to extern pixels

Created by aelar on 02/12/15.

**Author**

Benjamin PIAT

**Version**

1.0

**Date**

02 December 2015

Go through the image from bottom right to top left

## 7.14 `src/message/Message.java` File Reference

Prepare the message to be hidden.

**Classes**

- class [message.Message](#)

### 7.14.1 Detailed Description

Prepare the message to be hidden.

Created by aelar on 02/12/15.

**Author**

Benjamin PIAT

**Version**

1.0

**Date**

02 December 2015

