Package imageparser

public class **imageparser.ImageParser**

Cette classe permet de faire des operations d'analyse sur une image

Constructors **public ImageParser()**

Methods

public int getPixel(String path, int x, int y)
Recupere un pixel
Parameters
path - chemin de l'image
x - indice x du pixel
y - indice y du pixel
Returns
le pixel au format entier
Throws
-

public byte[] getPixelBuffer(String path)
Recupere un buffer de pixel
Parameters
path - chemin de l'image
Returns
retourne un tableau de byte contenant les pixel
Throws
-

public int compareImageRatioOpti(String path1, String path2, int hCut, int wCut, int correctness, int precision)
Compare deux images et retourne un ratio de correspondance
Parameters
path1 - chemin de l'image 1
path2 - chemin de l'image 2
hCut - coupure dans la hauteur
wCut - coupure dans la largeur
correctness - pourcentage de ressemblance autorise
precision - nombre de pixel analyse dans l'image
Returns
retourne le pourcentage de ressemblance de l'image
Throws
-

public int comparePixelTolerance(int pixel1, int pixel2, int correctness)
Compare deux pixels et renvoie leur diffÃ©rence
Parameters
pixel1 - premier pixel
pixel2 - deuxieme pixel
correctness - taux de ressemblance accepter
Returns
pourcentage de ressemblance

class **imageparser.ImageParserTest**

Constructors **ImageParserTest()**

Methods	**void getPixelBufferShouldReturnTheRightArray()**
	void compareImageRatioShouldGiveACentPourCentRatio()
	void comparePixelShouldBeRight()

Package JTreeManager

public class **JTreeManager.TaggedTreeNode** extends javax.swing.tree.DefaultMutableTreeNode

Extention de DefaultMutableTreeNode permettant d'ajouter un tag au noeud

Constructors

public TaggedTreeNode(String path, String tag)
constructeur du noeud
Parameters
path - le chemin du noeud
tag - le tag du noeud

public TaggedTreeNode(String path)
constructeur du noeud
Parameters
path - le chemin du noeud

Methods

public java.lang.String getTag()
Returns
retourne le tag

public void setTag(String tag)

Fields

private static final serialVersionUID

private tag

public class **JTreeManager.OldNodePair**

Cette classe permet d'associer des noeuds entre eux ainsi que de stocker un index.

Constructors

public OldNodePair(DefaultMutableTreeNode parent, DefaultMutableTreeNode child, int index)
Constructeur
Parameters
parent - le noeud parent du noeud enfant
child - le noeud enfant
index - l'index ou est placer le noeud enfant

Methods

public javax.swing.tree.DefaultMutableTreeNode getParent()
Returns
retourne le parent

public void setParent(DefaultMutableTreeNode parent)

public javax.swing.tree.DefaultMutableTreeNode getChild()
Returns
retourne l'enfant

public void setChild(DefaultMutableTreeNode child)

public int getIndex()
Returns
retourne l'index

public void setIndex(int index)

Fields **private parent**

private child

private index

public class **JTreeManager.JTreeManager** extends javax.swing.JPanel

Constructors **public JTreeManager()**
Constructeur du JtreeManager

Methods **public void addFiltre(AbstractTreeFilter f)**
Parameters
f - filtre etant ajouter au manager

public void removeFiltre(AbstractTreeFilter f)
Parameters
f - le filtre devant etre enlever

public void setSlider(SliderDemo s)
Parameters
s - slider qui va etre ajouter

public void setTable(ViewerTable t)
Parameters
t - qui va etre ajouter

private void setText(String text)
le JTreeManager contient un text qui permet d'afficher des informations cette methode permet d'ajouter le texte qui va etre montre
Parameters
text - qui va etre montre

private void releaseText()
Permet de clear le label

private void showTree()
affiche le tree

private void showWaiting()
affiche une bar de progression d'attente

Fields **private static final serialVersionUID**

private GUIRender

private final Filtre

private root

private tree

private final rootDirectory

private final JsonTree

private slider

private table

private parserTag

private final mutex

private messageBox

private flagFilter

waitingPanel

class **JTreeManager.JTreeManager.CellRenderer** extends javax.swing.tree.DefaultTreeCellRenderer

Constructors	**CellRenderer()**
Methods	**public java.awt.Component getTreeCellRendererComponent(JTree tree, Object value, boolean isSelected, boolean expanded, boolean leaf, int row, boolean hasFocus)**
	public java.awt.Color getBackgroundNonSelectionColor()
	public java.awt.Color getBackgroundSelectionColor()
	public java.awt.Color getForeground()
	private void setElementFont()
Fields	**private elementFont**
	private elementFontSelected
	private final ALPHA_OF_ZERO

Package animalType

public final class **animalType.AnimalType** extends java.lang.Enum

Enumeration des type d'animaux possible

Constructors	**private AnimalType(String animalName)**
Methods	**public static animalType.AnimalType[] values()**
	public static animalType.AnimalType valueOf(String name)
	public java.lang.String getName()
Fields	**public static final GRENOUILLE**
	public static final CRAPAUD
	public static final TRITON
	public static final AUTRE
	private final animalName

Package Shapes

public abstract class **Shapes.Shapes**

Classe implementant les formes

Constructors	**public Shapes(double x, double y)** Constructeur d'une forme **Parameters** x - position en x y - position en y
Methods	**public double getX()** Accesseur de x **Returns** la position x de la forme **public double getY()** Accesseur de y **Returns** la position y de la forme
Fields	**private x** **private y**

public class **Shapes.Rectangle** extends Shapes.Shapes

Classe implementant les rectangles

Constructors	**public Rectangle(double x, double y, double h, double l)** Constructeur **Parameters** x - position x y - position y h - hauteur l - largeur
Methods	**public double getWidth()** Accesseur de la largeur **Returns** la largeur du rectangle **public double getHeight()** Accesseur de la hauteur **Returns** la hauteur du rectangle
Fields	**private width** **private height**

public class **Shapes.Point** extends Shapes.Shapes

Classe implementant les points

Constructors	**public Point(double x, double y)** Constructeur **Parameters** x - position x y - position y

public class **Shapes.Cercle** extends Shapes.Shapes

Classe implementant les Cercle

Constructors	**public Cercle(double x, double y, double r)** Constructeur **Parameters** x - position x y - position y r - rayon du cercle
Methods	**public double getRadius()** Accesseur du rayon **Returns** le rayon du cercle
Fields	**private radius**

Package TagTest

class **TagTest.TagTest**

Constructors	**TagTest()**
Methods	**void shouldFormatATag()**
	void shouldFormatACircle()
	void shouldFormatAPoint()
	void shouldFormatARectangle()
	void shouldWriteAndReadImageTag()
	void shouldParseCsv()

Package searchfilters

public class **searchfilters.TemperatureTreeFilter** extends searchfilters.AbstractTreeFilter

extension de abstractTreeFilter permettant de filtrer les noeud par rapport a la tempÃ©rature

Constructors

public TemperatureTreeFilter(double tempMin, double tempMax)
Parameters
-
-

Methods

public boolean analyseNode(DefaultMutableTreeNode node)
regarde si le noeud est un noeud de date, regarde si l'heure correspond a la temperature voulue.
Parameters
node - le noeud etant analyse
Returns
si oui ou non on doit l'enleve de l'arbre

protected void filtreNode(DefaultMutableTreeNode node)

public java.lang.String toString()

Fields

private tempMin

private tempMax

private df

private dates

private ma

public class **searchfilters.TagTreeFilter** extends searchfilters.AbstractTreeFilter

extension de abstractTreeFilter permettant de filtrer les noeud par rapport aux tags

Constructors

public TagTreeFilter(boolean tagged)

Methods

public boolean analyseNode(DefaultMutableTreeNode node)
Regarde si le noeud est une image et si cette image est taggee.
Parameters
node - le noeud etant analyse
Returns
si oui ou non on doit l'enleve de l'arbre

public java.lang.String toString()

Fields

private final rootDirectory

private final tagged

public class **searchfilters.RatioTreeFilter** extends searchfilters.AbstractTreeFilter

extension de abstractTreeFilter permettant de filtrer les noeud par rapport au ratio

Constructors

public RatioTreeFilter(int min, int max, int tolerance, int precision)

Methods

public boolean analyseNode(DefaultMutableTreeNode node)
analyse les noeuds, sur les caracteristes entre les differences des images d'une sequance
Parameters
node - le noeud etant analyse
Returns
si oui ou non on doit l'enleve de l'arbre

protected void filtreNode(DefaultMutableTreeNode node)

public java.lang.String toString()

Fields

private ip

private min

private max

private tolerance

private precision

private rootDirectory

public abstract class **searchfilters.AbstractTreeFilter**

Cette classe implemente les methodes de base pour filtrer sur un jtree

Constructors

public AbstractTreeFilter(JTree tree)
Constructeur du filtre
Parameters
tree - l'arbre composer de DefaultMutableTreeNode

public AbstractTreeFilter()
Constructeur du filtre

Methods

public javax.swing.JTree getTree()
get l'arbre sur lequelle ce filtre pointe
Returns
l'arbre sur lequelle cette fonction filtre

public void setTree(JTree tree)
permet de placer sur quel arbre va filtrer le filtre
Parameters
-

protected void removeFromTree(DefaultMutableTreeNode node)
enleve un noeud de l'arbre et ces enfant
Parameters
node - le noeud a enlever

public void PopToTree()
remet le dernier element enleve dans l'arbre

public void filtreTree()
Filtre tous l'arbre en appliquant une condition de recherche

protected void filtreNode(DefaultMutableTreeNode node)
Filtre l'arbre en applicant une condition depuis la racine node.
Parameters
node - le noeud sur lequel nous filtrons

public void unfiltreTree()
remet dans l'arbre tous les noeud enleve

public abstract boolean analyseNode(DefaultMutableTreeNode node)
Parameters
node - le noeud analyser
Returns
si oui ou non le noeud doit etre enlever

public java.lang.String testToString()

public java.lang.String toString()

Fields

static counter

private num

private filtredElements

private tree

private model

private root

public class **searchfilters.MeteoTreeFilter** extends searchfilters.AbstractTreeFilter

Extention de la classe abstractTreeFilter permetant de trier les noeuds de dates en fonction de la meteo ce jour la.

Constructors

public MeteoTreeFilter(TYPEMETEO meteo)
Constructeur de filtre meteo
Parameters
meteo - la meteo voulue

Methods

public boolean analyseNode(DefaultMutableTreeNode node)
analyse les noeuds, si c'est des noeuds de date, la methode regarde si la date a la bonne meteo (pluie, beau,...)
Parameters
node - le noeud etant analyse
Returns
si oui ou non on doit l'enleve de l'arbre

protected void filtreNode(DefaultMutableTreeNode node)
modification de filtreNode pour qu'il ne parcours pas les noeud plus loin que date

public java.lang.String toString()

Fields

private meteo

private df

private dates

private ma

public class **searchfilters.DateTreeFilter** extends searchfilters.AbstractTreeFilter

extention de abstractTreeFilter permettant de filtrer les noeud par rapport au date

Constructors

public DateTreeFilter(Date startdate, Date endDate)
Constructeur du filtre
Parameters
startdate - date de debut du filtrage
endDate - date de fin du filtrage

Methods

public boolean analyseNode(DefaultMutableTreeNode node)
regarde si un noeud est un noeud de date et s'y il est compris dans les date min et max
Parameters
node - le noeud etant analyse
Returns
si oui ou non on doit l'enleve de l'arbre

protected void filtreNode(DefaultMutableTreeNode node)
modification de filtreNode pour qu'il ne parcours pas les noeud plus loin que date

public java.lang.String toString()

Fields

private startDate

private endDate

class **searchfilters.treeFilterTest**

Constructors

treeFilterTest()

Methods

void treeFilterCanAddAndRemoveNode()

void treeFilterFiltreCorrectly()

private java.lang.String printTree(DefaultMutableTreeNode node)

Package jsontreeparse

public class **jsontreeparse.JsonTreeParser**

Constructors

public JsonTreeParser()

Methods

public void parseHierarchyTag()
Cette methode parse les tag (le type de dossier de l'arborescence) Exemple un dossier tage pourrai etre un dossier de date ou d'heure. Ceci nous permet de simplifier differentes operation sur le filtrage des images.

public void createJson(File rootDirectory, int history)
Cette methode permet de generer un fichier json en explorant l'arborescence des fichiers
Parameters
rootDirectory - Le dossier racine ou l'exploration de l'arborescence s'executera

private com.google.gson.JsonArray setJson(File file, int i, int history)
Parameters
file - the file or the directory that will be explored
i - the index of the tag in the hierarchytag array
Returns
an JsonArray that containe the JPG, SubDirectory and subFile in the directory explored.
Throws
-

public javax.swing.tree.DefaultMutableTreeNode setDirectoryTree(String path)
Parameters
path - chemin du fichier json contenant l'arborescence
Returns
un arbre composer de treenode representant l'arborescance des fichiers

public void createTree(JsonArray a, TaggedTreeNode d)
Parameters
a - le json array contenant le dossier et les sousdossier
d - le noeud parent de ce dossier

Fields

private hierarchyTag

properties

private histArray

private counterhist

Package exceptionHandler

public class **exceptionHandler.LogFileWritingHandler**

Classe qui gere les exceptions

Constructors	**public LogFileWritingHandler()**
Methods	**public static void handleException(String m, StackTraceElement[] s)**

Package GUI

public class **GUI.GUIRender**

Cette classe regroupe divers parametres graphiques utilisable dans les autres classes

Constructors	**public GUIRender()**
Methods	**public static java.awt.Color getForeColor()**
	public static java.awt.Color getBackColor()
	public static java.awt.Color getButtonColor()
	public static java.awt.Color getButtonSelectedColor()
	public static java.awt.Font getSectionTitle()
	public static java.awt.Font getMainTitle()
	public static java.awt.Font getElementSelected()
	public static java.awt.Font GetElement()
Fields	**private static final CRAPA_VIOLET**
	private static final BUTTON_VIOLET
	private static final BUTTON_VIOLET_SEL
	private static mainTitle
	private static sectionTitle
	private static element
	private static elementSelected

public class **GUI.DateFiltrer** extends GUI.TreeFilter

Classe implementant l'interface pour le filtre par date

Constructors	**public DateFiltrer(JTreeManager manager)** Constructeur **Parameters** manager - Jtree de la banque d'image
Methods	**protected void specialisation()**
Fields	**private static final serialVersionUID**

public class **GUI.TemperatureFilter** extends GUI.TreeFilter

Classe implementant l'interface pour le fitre meteo

Constructors **public TemperatureFilter(JTreeManager manager)**
Constructeur
Parameters
manager - JTreeManager sur lequelle les filtres vont filtrer

Methods **protected void specialisation()**
specialisation des filtres pour la temperature

Fields **private static final serialVersionUID**

public class **GUI.ViewerTable** extends javax.swing.JPanel

Classe implementant l'interface du tableau pour les tags

Constructors **public ViewerTable()**
Constructeur

Methods **public void addRow()**
Ajoute un ligne au tableau

public void delRow()
Supprime une ligne au tableau

public void clear()
Vide le tableau

public java.util.ArrayList getData()
Retourne la valeur des cases du tableau
Returns
Valeurs des cases du tableau

public void setUpAnimalColumn(JTable table, TableColumn animalColumn)
Set la combobox
Parameters
table - Tableau
animalColumn - Colonne contenant la combobox

public void setTags(ArrayList tags)
Remplis le tableau avec les tags presents sur l'image
Parameters
tags - Tags de l'image

Fields **private static final serialVersionUID**

private final table

private columnNames

private rowData

private final row

model

public class **GUI.WeatherFilter** extends GUI.TreeFilter

Classe implementant l'interface pour le fitre meteo

Constructors	**public WeatherFilter(JTreeManager manager)** Constructeur **Parameters** manager - jtree de la banque d'image
Methods	**protected void specialisation()**
Fields	**private static final serialVersionUID**

private class **GUI.WeatherFilter.ComboBoxRenderer** extends javax.swing.JLabel implements javax.swing.ListCellRenderer

Constructors	**public ComboBoxRenderer()**
Methods	**public java.awt.Component getListCellRendererComponent(JList list, Object value, int index, boolean isSelected, boolean cellHasFocus)**

public class **GUI.SliderDemo** extends javax.swing.JPanel implements java.awt.event.ActionListener, java.awt.event.WindowListener

Classe implementant le slider d'images

Constructors	**public SliderDemo()** Constructeur
Methods	**void addWindowListener(Window w)** Add a listener for window events. **public void windowIconified(WindowEvent e)** **public void windowDeiconified(WindowEvent e)** **public void windowOpened(WindowEvent e)** **public void windowClosing(WindowEvent e)** **public void windowClosed(WindowEvent e)** **public void windowActivated(WindowEvent e)** **public void windowDeactivated(WindowEvent e)** **public void startAnimation()** Demarre l'animation **public void stopAnimation()** Stoppe l'animation **public void actionPerformed(ActionEvent e)**

public void nextPicture()
Update the label to display the image for the current frame.

public void prevPicture()
Update the label to display the image for the current frame.

protected static javax.swing.ImageIcon createImageIcon(String path)
Returns an ImageIcon, or null if the path was invalid.
Parameters
path - Chemin de l'image
Returns
Icone de l'image

private java.lang.String getFileExtension(File file)
Retourne l'extension d'un fichier
Parameters
file - fichier
Returns
extension du fichier

public void addImage(String path)
Ajoute les images au slider
Parameters
path - chemin de l'image

public java.lang.String getImage()
Retourne le chemin de l'image actuelle
Returns
chemin de l'image actuelle

public java.lang.String getDirectory()
Retourne le chemin du dossier de l'image actuelle
Returns
chemin du dossier de l'image actuelle

public void setTable(ViewerTable table)
Setteur du tableau de tags
Parameters
table - tableau de tags

Fields

private static final serialVersionUID

private static final FPS_INIT

private frameNumber

private images

private final delay

private final timer

private directory

private final imagesPath

private final parserTag

private table

picture

public class **GUI.FiltersPanel** extends javax.swing.JPanel

Classe implementant l'interface pour les filtres

Constructors	**public FiltersPanel()** Constructeur
Methods	**public void setManager(JTreeManager manager)** **Parameters** manager - le JTreeManager sur lequelle les filtres doivent filtrer
Fields	**private static final serialVersionUID**
	private final panel

public class **GUI.TagFilter** extends GUI.TreeFilter

Classe implementant l'interface pour le filtre par tag

Constructors	**public TagFilter(JTreeManager manager)** Constructeur **Parameters** manager - jtree de la banque d'image
Methods	**protected void specialisation()**
Fields	**private static final serialVersionUID**
	private tagChecked

public class **GUI.ChangeFilter** extends GUI.TreeFilter

Implementation de l'interface pour le filtre par changement d'image

Constructors	**public ChangeFilter(JTreeManager manager)** Constructeur **Parameters** manager - Jtree de la banque d'image
Methods	**protected void specialisation()**
Fields	**private static final serialVersionUID**

public abstract class **GUI.TreeFilter** extends javax.swing.JPanel

Cette classe est une representation graphique des filtres sur le JTree

Constructors **protected TreeFilter()**
Constructeur

public TreeFilter(JTreeManager manager)
Constructeur avec manager
Parameters
manager - jtree de la banque d'image

Methods **protected abstract void specialisation()**
Cet methode doit etre implementee dans les sous classe extendant celle-ci Elle permet d'ajouter des elements graphiques differents entre les filtres

private void common()
Methode qui cree les elements graphiques de base commun a chaque filtre

Fields **protected panel**

protected specialistationPanel

protected label

private static final serialVersionUID

private final GUIRender

private delete

protected filter

protected manager

protected currentFilter

private class **GUI.TreeFilter.SelButton** extends javax.swing.JButton

Constructors **public SelButton(String text)**

public class **GUI.statisticsPage** extends javax.swing.JFrame

Cette classe construit une fenetre de statistiques sur la banque d'images

Constructors **public statisticsPage()**
Constructeur

Methods **public void initAndShowGUI()**
Fonction de creation de l'interface graphique

private javafx.scene.Scene createScene()
Fonction de creation de la Scene JaxaFX
Returns
la scene cree

public javafx.scene.Group createDynamGroup()
Cree un Group JavaFX avec la zone de generation dynamique, a savoir les graphes par mois

et par jour
Returns
le groupe cree

public void populateDayLineChart(LineChart dayLineChart)
Population du graphique animaux totaux par jour
Parameters
dayLineChart - la chart a peupler

public void populateMonthLineChart(LineChart monthLineChart)
Population du graphique animaux totaux par mois
Parameters
monthLineChart - la chart a peupler

public void populateMonthBarChart(StackedBarChart sbcMonth)
Population du graphique animaux par type et par mois
Parameters
sbcMonth - la chart a peupler

public void populateDayBarChart(StackedBarChart sbcDay)
Population du graphique animaux par type et par jour
Parameters
sbcDay - la chart a peupler

public javafx.scene.Group createMainPieChart()
Creation d'un groupe avec la pie chart
Returns
le group cree

public javafx.scene.Group createLineChart()
Cree un groupe avec la line chart par annee
Returns
le group cree

public javafx.scene.Group createBarChart()
Cree un groupe avec la bar chart par annee
Returns
le groupe cree

public javafx.scene.Group createSideInfosZone()
Cree un groupe avec la zone d'infos droite
Returns
le groupe cree

private void initFX(JFXPanel fxPanel)
Initialisation du FxPanel avec la Scene
Parameters
fxPanel - le FxPanel a initialiser

public void statistics()
Main de JavaFX

Fields

private parser

private statHandler

private dayConfig

private monthConfig

public class **GUI.crapau** extends javax.swing.JFrame

Interface graphique du projet

Constructors

public crapau()
Creates new form crapau

Methods

private void initComponents()
This method is called from within the constructor to initialize the form. WARNING: Do NOT modify this code. The content of this method is always regenerated by the Form Editor.

private void navigate_labelMouseClicked(MouseEvent evt)

private void play_buttonActionPerformed(ActionEvent evt)

private void undo_buttonActionPerformed(ActionEvent evt)

private void reload_buttonActionPerformed(ActionEvent evt)

private void exit_buttonActionPerformed(ActionEvent evt)

private void exit_buttonMouseClicked(MouseEvent evt)

private void pause_buttonActionPerformed(ActionEvent evt)

private void next_buttonActionPerformed(ActionEvent evt)

private void add_rowActionPerformed(ActionEvent evt)

private void save_buttonActionPerformed(ActionEvent evt)

private void prev_buttonActionPerformed(ActionEvent evt)

private void formWindowOpened(WindowEvent evt)

private void settings_labelMouseClicked(MouseEvent evt)

private void save_button1ActionPerformed(ActionEvent evt)

private void save_button1MouseClicked(MouseEvent evt)

public static void main(String[] args)
Parameters
args - the command line arguments

Fields

private static final serialVersionUID

private final GUIRender

private tagManager

private stats

private BackGround

private Edit

private Filters

private Galerie

private Navigate

private Navigate_Titre

private Settings

private Side

private Titre

private add_row

private edit_label

private exit_button

private filtersPanel1

private filters_label

private jLabel1

private jScrollPane1

private jTreeManager1

private navigate_label

private next_button

private pause_button

private play_button

private prev_button

private reload_button

private save_button

private save_button1

private settings_label

private sliderDemo1

private tag_label

private undo_button

private viewerTable

Package Statistics.components

public class **Statistics.components.Tag**

Represente un tag sur une image

Constructors	**public Tag()** Constructeur
Methods	**public void setTypeAnimal(String animalType)**
	public java.lang.String getAnimalType()
	public void setIsMale(boolean isMale)
	public void setSize(double size)
	public void setIsEnteringTunnel(boolean isEnteringTunnel)
	public java.lang.String toString()
Fields	**private animalType**
	private isMale
	private size
	private isEnteringTunnel

public final class **Statistics.components.Month** extends java.lang.Enum

Enum representant les mois de l'annee avec leur nombre de jours

Constructors	**private Month(String name, String abbreviation, int nbDays)** Constructeur **Parameters** name - le nom du mois abbreviation - son abreviation nbDays - le nombre de jours qu'il a
Methods	**public static Statistics.components.Month[] values()**
	public static Statistics.components.Month valueOf(String name)
	public java.lang.String getName()
	public java.lang.String getAbbreviation()
	public int getNbDays()
Fields	**public static final JAN**
	public static final FEB
	public static final MAR
	public static final APR

public static final MAY

public static final JUN

public static final JUL

public static final AUG

public static final SEP

public static final OCT

public static final NOV

public static final DEC

private final name

private final abbreviation

private final nbDays

public class **Statistics.components.Image**

Represente une image de la banque de donnee avec tous ses attributs

Constructors

public Image(String path, ArrayList tags)
Constructeur
Parameters
path - le chemin d'acces de l'image
tags - la liste des tags enregistres sur l'image

Methods

private static java.lang.String[] splitPath(String pathString)
Renvoie une representation du path sous forme d'un tableau, chaque element du tableau est un sous-dossier
Parameters
pathString - le chemin d'acces a separer
Returns
le tableau genere

private void pathDecomposition(String path)
Enregistrement des attributs du path dans les attributs de la classe Image
Parameters
path - le chemin d'acces a decortiquer

public java.util.ArrayList getTags()

public java.lang.String getCamera()

public java.lang.String getDate()

public java.lang.String getSequence()

public Statistics.components.Month getMonth()

public int getDay()

public int getHour()

public boolean hasTags()

public java.lang.String toString()

Fields

private camera

private date

private sequence

private month

private day

private hour

private final tags

Package properties

public class **properties.PropertiesHandler**

Cette classe permet de parser le fichier conf.properties

Constructors	**public PropertiesHandler()**
Methods	**public static java.util.Properties parseProperties()** parse les proprietes du fichier. **Returns** les proprietes du fichier

Package meteoAPI

public final class **meteoAPI.TYPEMETEO** extends java.lang.Enum

Enumeration representant les differentes conditions meteorologique possible

Constructors

private TYPEMETEO(String meteo)
Constructeur de l'enum
Parameters
-

Methods

public static meteoAPI.TYPEMETEO[] values()

public static meteoAPI.TYPEMETEO valueOf(String name)

public java.lang.String toString()
Retourne les conditions en string
Returns
meteo

public static meteoAPI.TYPEMETEO getTypeByString(String condition)
Retourne les bytes de la condition fournie en parametre
Parameters
-
Returns
weather

Fields

public static final DEGAGE

public static final PEUNUAGUEX

public static final TRESNUAGEUX

public static final OVERCAST

public static final Rain

public static final LightRain

public static final Snow

public static final LightSleet

public static final Foggy

private meteo

public class **meteoAPI.MeteoPerDay**

Classe regroupant les informations meteo par jour

Constructors

public MeteoPerDay(String date)
Constructeur de la classe lorsque la date est donnee en string
Parameters
-

public MeteoPerDay(Date date)

Constructeur de la classe lorsque la date est donnee en format Date
Parameters
-

Methods

public java.util.Date getDate()
Methode retournant la date de l'objet
Returns
date

public java.util.List getMeteo()
Methode retournant la liste des condition meteorologique pour chaque heure
Returns
meteo

public void addMeteo(Object summary)
Methode permettant d'ajouter une condition meteorologique Ã l'objet
Parameters
-

public java.util.List getTemperature()
Methode retournant la liste des temperature pour chaque heure
Returns

public void addTemperature(Object temperature)
Methode permettant d'ajouter une temperature Ã l'objet
Parameters
-

public void setProper()
Initialise toutes les cases de la liste a la valeur par dÃ©faut

Fields

private date

private Meteo

private Temperature

public class **meteoAPI.MeteoAPI**

Class MeteoAPI parser de fichier Json, elle parcourt toutes les dates des images, recupere les informations necessaire afin de creer des objets representant les differentes donnees voulues (MeteoPerDay).

Constructors

public MeteoAPI()

Methods

public java.util.List getList()
Methode recuperant tout les informations meteo fournies par l'API
Returns
listMeteoPerDay

public java.util.List getListFiltreSummary(TYPEMETEO filtre)
Methode recuperant tout les informations meteo fournies par l'API en lui appliquant un filtre avec une condition meteorologique
Returns
listMeteoPerDay

public java.util.List getListFiltreTemperature(double min, double max)
Methode recuperant tout les informations meteo fournies par l'API en lui appliquant un filtre avec des temperatures

Returns
listMeteoPerDay

private void parseDateObject(JsonObject date)
Methode recuperant toutes informations sur le Json pour une date
Parameters
-

private void parseHourObject(MeteoPerDay met, JsonObject hour)
Methode recuperant pour une date donnee, toutes les informations pour chaque heure du jour
Parameters
-
-

Fields **private listMeteoPerDay**

public class **meteoAPI.meteoAPITest**

Constructors **public meteoAPITest()**

Methods **public void recuperationFichier()**
Throws
-

public void lectureFichier()
Throws
-

public void creationdelameteodu23fevrier()

public void recuperationDeToutesLesDateOuIlFaisaitBeau()

public void recuperationDeToutesLesTemperatures()

Package Tag

public class **Tag.Tag**

Classe servant a formater les tags au format CSV (;) et a les envoyer au parser.

Constructors

public Tag()

Methods

public void setTag(ArrayList tag, Shapes shape)
Ajout d'un tag a la liste des tags en le formatant
Parameters
tag - Tag a formater
shape - Forme a formater

public java.lang.String formatTag(ArrayList tag)
Format un tag en ligne csv
Parameters
tag - Tag Ã formater
Returns
Tag formate

public java.lang.String formatShape(Shapes shape)
Formate une forme en ligne CSV
Parameters
shape - Forme Ã formter
Returns
Forme formatee

public void saveTags(ArrayList tags, String imagesPath)
Formate la liste de tags et l'envoie au parser pour les sauver sur l'image ou les images de la sequence
Parameters
tags - Liste des tags
imagesPath - Chemin de l'image ou du dossier

Fields

private tags

private parser

public class **Tag.Parser**

Classe de gestion des metadatas d'images avec l'ajout et la lecture des tags.

Constructors

public Parser()

Methods

public void setTags(ArrayList tags, String imagesPath)
Ajoute les tags sur une image ou sur toutes les images d'un repertoire
Parameters
tags - Liste de tags
imagesPath - Chemin d'une image ou d'un repertoire

private void writeTag(File file, ArrayList tags)
Ecrit les tags sur une image
Parameters
file - Image Ã modifier
tags - Liste de tags

private java.lang.String getFileExtension(File file)
Recupere l'extension d'un fichier
Parameters
file - Fichier
Returns
Retourne l'extension

private void addTextEntry(IIOMetadata metadata, String key, ArrayList value)
Modifie les metadatas en ajoutant les tags. Si il y a deja des tags presents sur l'image on les supprimes et on ajoute les nouveaux.
Parameters
metadata - Metadata de l'image
key - Valeur unique definissant un tag
value - Liste de tags
Throws
-
-

public java.util.ArrayList getTextEntry(IIOMetadata metadata, String key)
Recupere les tags present sur l'image
Parameters
metadata - Metadata de l'image
key - Valeur unique definissant un tag
Returns
Liste des tags presents

private boolean findTag(IIOMetadata metadata, String key)
Trouve la prÃ©sence de tags sur l'image
Parameters
metadata - Metadata de l'image
key - Valeur unique definissant un tag
Returns
True si il y des tags sinon false

public boolean isTagged(File file)
Trouve la presence de tags sur l'image
Parameters
file - Image Ã tester
Returns
True si il y des tags sinon false
Throws
-

public java.util.ArrayList getTag(String path)
Recupere les tags d'une image
Parameters
path - Chemin de l'image
Returns
Liste des tags presents sur l'image
Throws
-

public class **Tag.CsvParser**

Classe decoupant les tags en liste de String

Constructors	**public CsvParser()**
Methods	**public static java.util.ArrayList getTag(ArrayList tags)**

DÃ©coupe les tags en liste de String

Parameters

tags - Tags de l'image

Returns

Liste des tags decoupes

public class **Tag.TagHistory**

Implemente les fonctions pour creer un fichier JSON contenant les tags ajoutes aux images pour pouvoir effectuer les statistiques sans parcourir toutes les images Ã chaque fois

Constructors

public TagHistory()

Methods

public static void saveTag(ArrayList tags, String imagesPath)

Enregistre ou met Ã jour les tags pour une ou plusieurs images dans un fichier history.json

Parameters

tags - Liste des tags

imagesPath - Chemin de l'image ou du dossier

private static void createHistory(File history, ArrayList tags, String imagesPath)

Cree la base du fichier json avec les premier tags

Parameters

history - Fichier d'historique

tags - Liste des tags

imagesPath - Chemin de l'image

private static void updateTags(File history, ArrayList tags, String imagesPath, JsonElement json)

Met a jour les tags pour une image deja enregistree

Parameters

history - fichier d'historique

tags - liste des tags

imagesPath - chemin de l'image

json - element racine ou ajouter les tags

private static java.lang.String getFileExtension(File file)

Recupere l'extension d'un fichier

Parameters

file - Fichier

Returns

Retourne l'extension

public static java.lang.String getRelativePath(String path)

Retourne le chemin relatif d'une image

Parameters

path - chemin absolu de l'image

Returns

chemin relatif

public static boolean findTag(String path)

public static void getPaths()

Fields

private static paths

Package Statistics.handler

public class **Statistics.handler.StatisticsHandler**

Sert a enregistrer toutes les donnees necessaires a la generation des statistiques

Constructors

public StatisticsHandler()
Constructeur

Methods

private void initiasize()
Initialisation des attributs et des mappages Certains mappages et attributs necessitent la presence d'une valeur minimale (0) afin de peupler toutes les chartes graphiques, cette fonction intialise ces mappages et attributs

private java.util.List createListOfAnimals()
Enregistre combien d'animaux il y a pour chaque type dans une liste
Returns
la liste generee

private java.lang.Integer findLimitValueInMap(Map map, String type)
Cherche la valeur minimale ou maximale de la map
Parameters
map - la map a sonder
type - min ou max
Returns
la valeur minimale ou maximale

private java.util.List findLimitKeysInMap(Map map, int value)
Cherche la cle donnant la valeur minimale ou maximale de la map
Parameters
map - la map a sonder
value - la valeur min ou max de la map
Returns
la liste des cles trouvees

private java.lang.String returnDataString(List list)

private void countInMap(Map map, Object key, int value)
Insere une valeur et potentiellement sa cle dans une map Permet de peupler facilement tous les mappages non initialises
Parameters
map - la map ou faire l'insertion
key - la cle
value - la valeur

public void countCameraObservation(String cameraName, int numberOfTagsForAnImage)
Compte les observations pour une camera
Parameters
cameraName - le nom de la camera
numberOfTagsForAnImage - la valeur a mettre dans la cle de la camera

public void countDaysObservation(String dayName, int numberOfTagsForAnImage)
Compte les observations pour un jour
Parameters
dayName - le nom du jour
numberOfTagsForAnImage - la valeur a mettre dans la cle du jour

public void countSequenceObservation(String sequenceName, int numberOfTagsForAnImage)
Compte les observations pour une sequence

Parameters

sequenceName - le nom de la sequence
numberOfTagsForAnImage - la valeur a mettre dans la cle de la sequence

public void countMonthlyObservation(Month month, int numberOfTagsForAnImage)
Compte les observations pour un mois
Parameters

month - le nom du mois
numberOfTagsForAnImage - la valeur a mettre dans la cle du mois

public void countDailyObservation(Month month, int day, int numberOfTagsForAnImage)
Compte les observations pour un jour
Parameters

month - le nom du mois
day - le jour
numberOfTagsForAnImage - la valeur a mettre dans la cle du jour

public void countHourlyObservation(Month month, int day, int hour, int numberOfTagsForAnImage)
Compte les observations pour une heure
Parameters

month - le nom du mois
day - le jour
hour - l'heure
numberOfTagsForAnImage - la valeur a mettre dans la cle de l'heure

public void countTotalObservationsByAnimalType(String animal)
Compte le nombre d'observations totales d'un animal donne (peuple une map)
Parameters

animal - l'animal a traiter

public void countMonthlyObservationsByAnimalType(String animal, Month month)
Compte le nombre d'observations d'un animal donne (peuple une map) pour un mois donne
Parameters

animal - l'animal a traiter
month - le mois a traiter

public void countDailyObservationsByAnimalType(String animal, Month month, int day)
Compte le nombre d'observations d'un animal donne (peuple une map) pour un jour donne
Parameters

animal - l'animal a traiter
month - le mois a traiter
day - le jour a traiter

public void countHourlyObservationsByAnimalType(String animal, Month month, int day, int hour)
Compte le nombre d'observations d'un animal donne (peuple une map) pour une heure donnee
Parameters

animal - l'animal a traiter
month - le mois a traiter
day - le jour a traiter
hour - l'heure a traiter

public void analyzeData()
Genere les observations min et max pour les cameras, les sequences et les jours Si aucune image possede un tag, un message alternatif est affiche

public void addImage(Image image)

public void addNbAnimals(int n)

public void addNbImages(int n)

public void addNbSequences(int n)

public void addNbTaggedSequences(int n)

public int getNbImages()

public int getNbTaggedImages()

public int getNbSequences()

public int getNbTaggedSequences()

public int getNbUntaggedSequences()

public int getNbUntaggedImages()

public java.util.Map getAnimalTypeCounter()

public void setTotalNbOfAnimals(int n)

public int getTaggedAnimals()

public int getTaggedImages()

public java.lang.String getMostUsedCamera()

public java.lang.String getLeastUsedCamera()

public java.lang.String getMostFrequentDate()

public java.lang.String getLeastFrequentDate()

public java.lang.String getMostTaggedSequence()

public java.lang.String getLeastTaggedSequence()

public int getAnimalNbByMonth(Month month)

public java.util.Map getAnimalNbByHourMap(Month month, int day)

public java.util.Map getAnimalNbByDayMap(Month month)

public java.util.List getAnimalNbByMonthByType(Month month)

public java.util.Map getAnimalTypeByDayMap(Month month)

public java.util.Map getAnimalTypeByHourMap(Month month, int day)

Fields **private static final HOURSINADAY**

private static final MIN

private static final MAX

private images

private meteo

private animalTypeCounter

private cameraObservations

private dateObservations

private sequenceObservations

private monthlyObservations

private monthlyObservationsByAnimalType

private dailyObservations

private dailyObservationsByAnimalType

private hourlyObservations

private hourlyObservationsByAnimalType

private nbImages

private nbSequences

private nbTaggedImages

private nbTaggedSequences

private nbAnimals

cameraMaxKeys

cameraMinKeys

dateMaxKeys

dateMinKeys

sequenceMaxKeys

sequenceMinKeys

Package Statistics.parser

public class **Statistics.parser.StatParser**

Parse le fichier history qui contient les informations json de la banque d'images

Constructors	**public StatParser(StatisticsHandler statHandler)**
Methods	**public com.google.gson.JsonParser parseFile()** Parsage du fichier history **Returns** un objet JsonParse avec les informations generees **private void parseImageContent(JsonArray content)** Parse le contenu d'un JsonArray et remplit l'objet statHandler qui contiendra les informations necessaires a la generation des graphes **Parameters** content - le JsonArray a parser **private java.util.ArrayList parseContentTags(JsonArray tags)** Parse le JsonArray des tags **Parameters** tags - le JsonArray des tags **Returns** un ArrayList des tags recuperes
Fields	**private static final HISTORIC** **private final statHandler** **private currentSequence**

Package jsonparser

public class **jsonparser.JsonTreeParserTest**

Constructors	**public JsonTreeParserTest()**
Methods	**void getPixelBufferShouldReturnTheRightArray()**
	private java.lang.String printTree(DefaultMutableTreeNode node)