



## PRAXIS DER SOFTWAREENTWICKLUNG

GRUPPE 30

---

### knipsX Entwurf

---

Gruppenmitglieder:  
Borovik, Volodymyr  
Bouché, Kai  
Draxler, Benjamin  
Kaufman, David  
Zuber, Kevin

Gruppenbetreuer:  
Meder, David

16. Dezember 2009 - Revision: 1.0

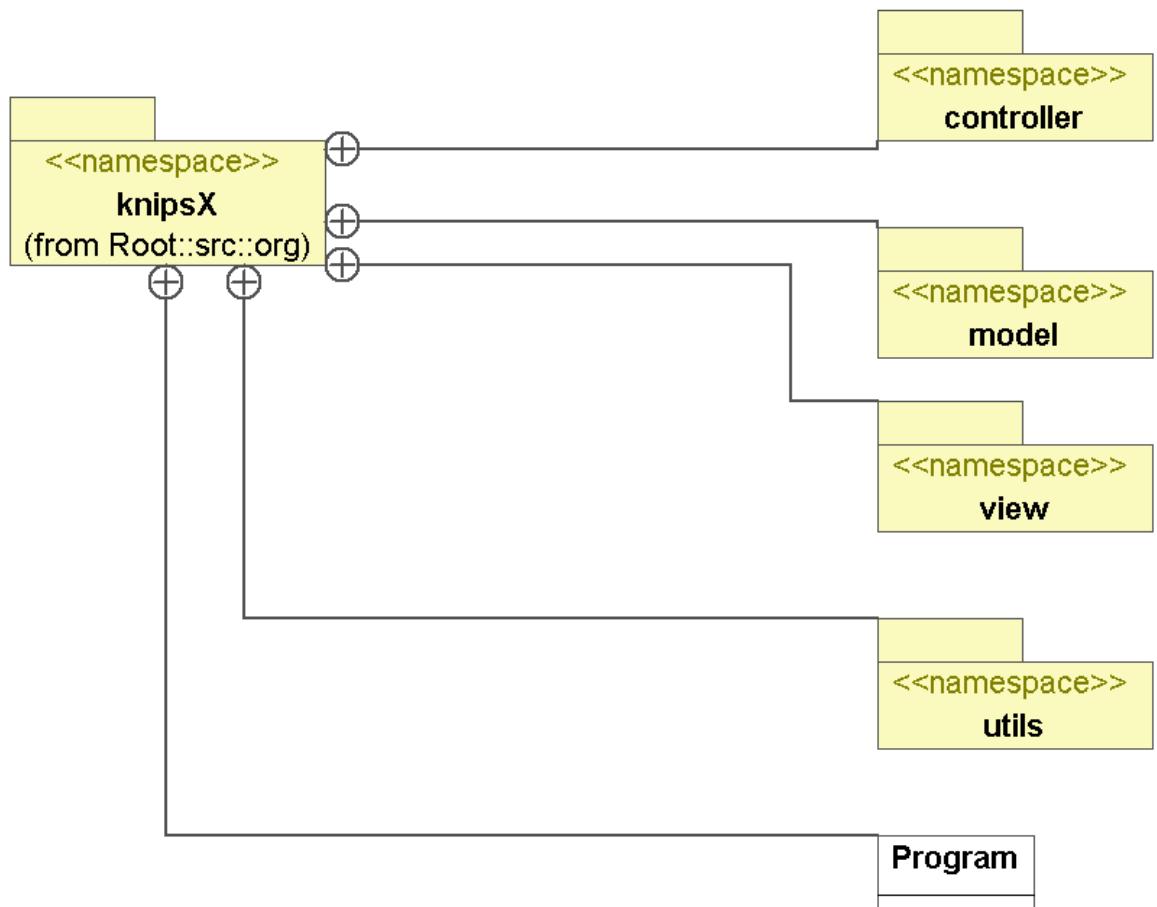
Im folgenden Dokument sehen Sie den Entwurf von knipsX, spezifiziert in UML 2.2. Auf den ersten Seiten befindet sich das Paketdiagramm, danach kommen die Klassendiagramme, die Sequenzdiagramme und noch eine Beschreibung aller Klassen mit deren Attributen und Methoden. Im Anhang finden sie noch eine Liste aller Klassen.

Den Diagrammen liegen folgende Legende zugrunde: Klassenfamilienfarben:

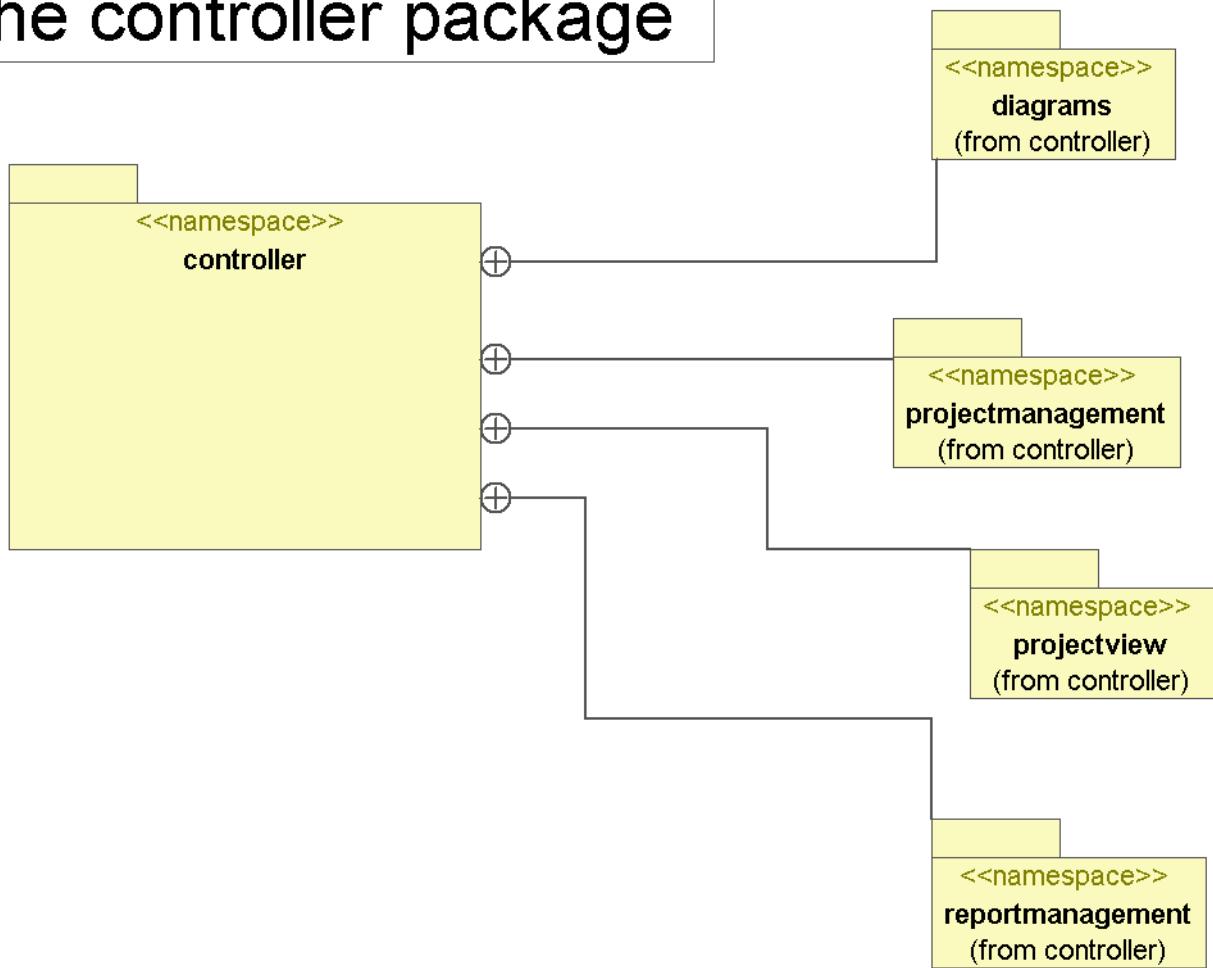
1. Rot: Externe Klassen meist Java SDK
2. Grün: Alle Klassen die zur View gehören.
3. Blau: Alle Klassen die die zugrunde liegenden Daten bilden, sprich das Model.
4. Gelb: Alle Klassen die für die Steuerung des Programms notwendig sind, sprich die Controller.

Zudem befinden sich Kommentare in den Diagrammen, diese erkennt man an der umgeschlagenen Ecke und an der gelb-orangen Färbung.

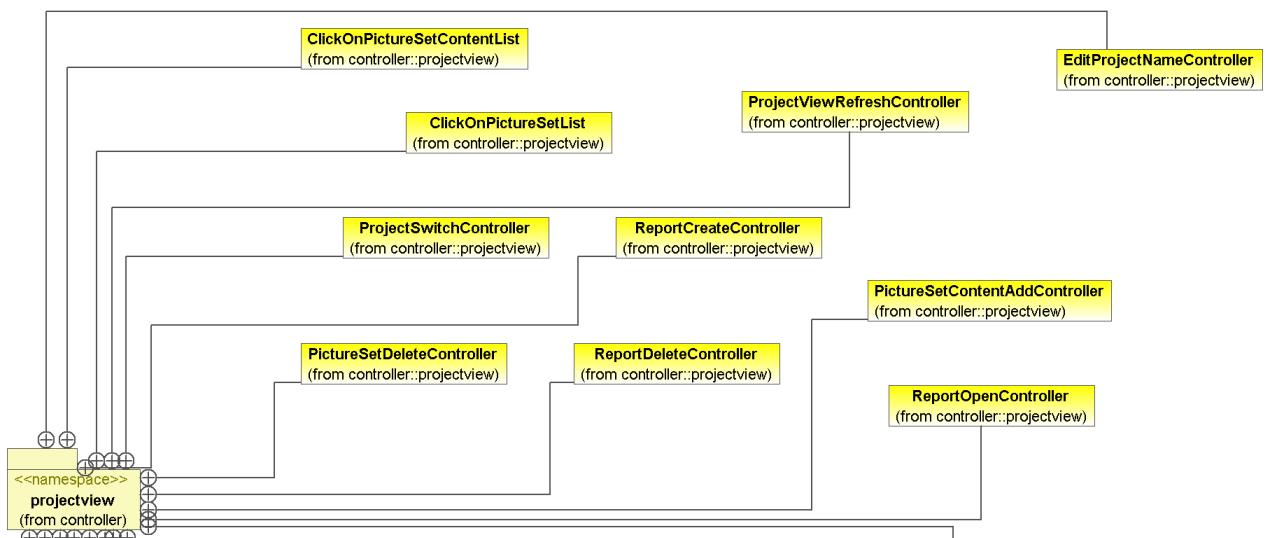
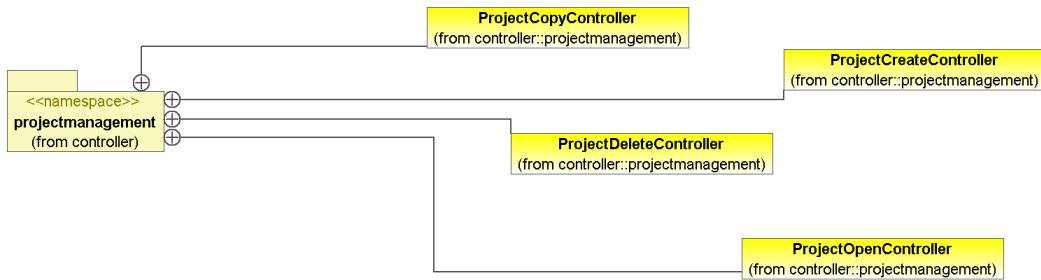
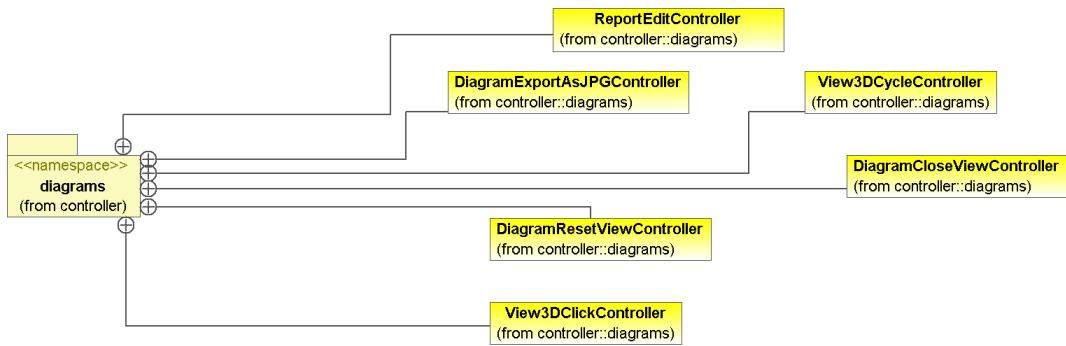
# The knipsX package

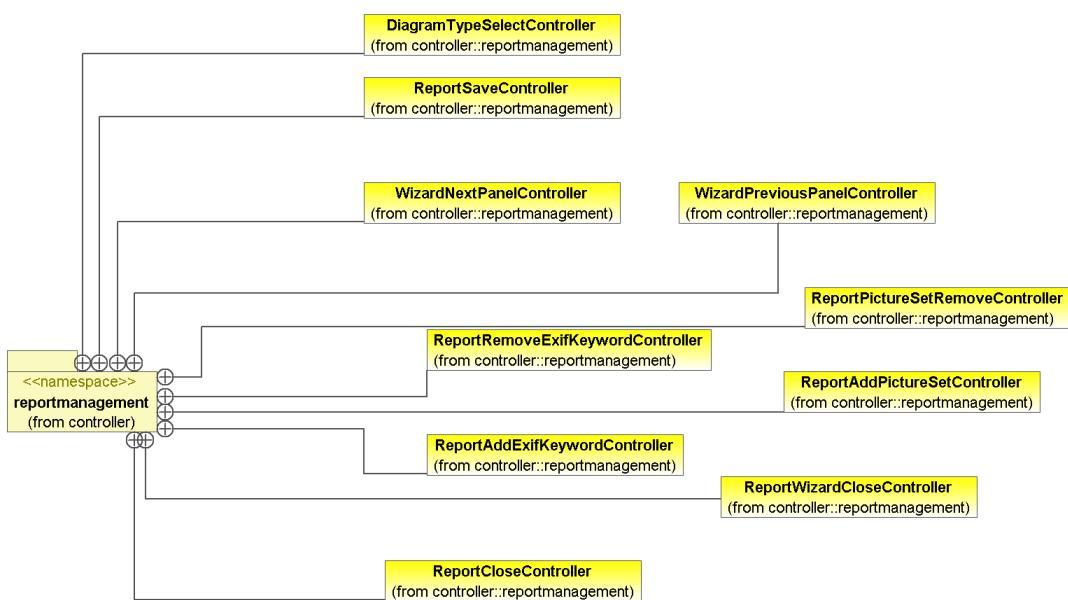
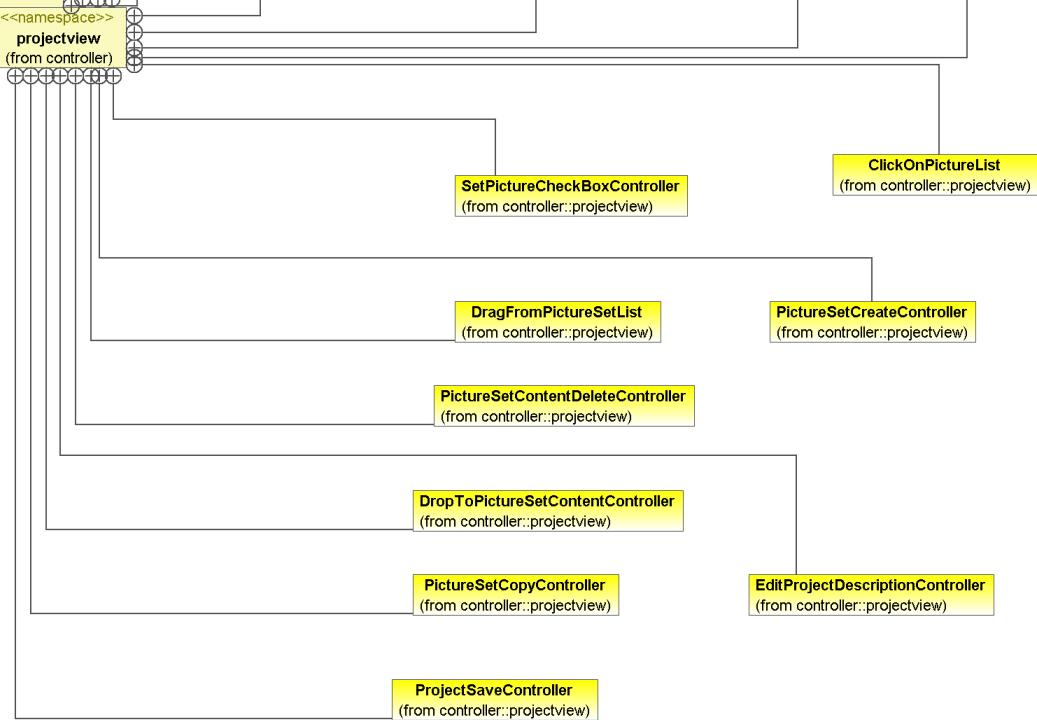


# The controller package

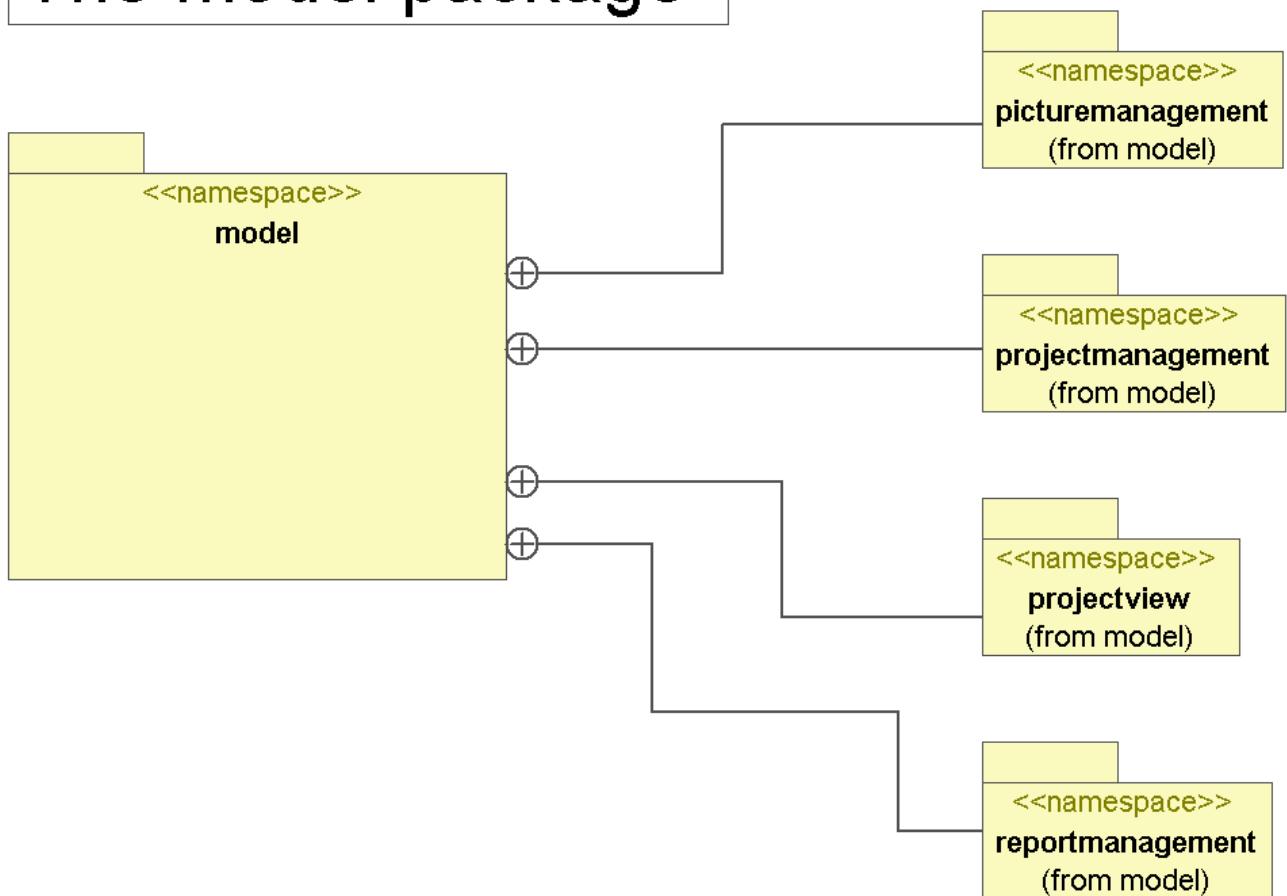


## Detailed Package - controller

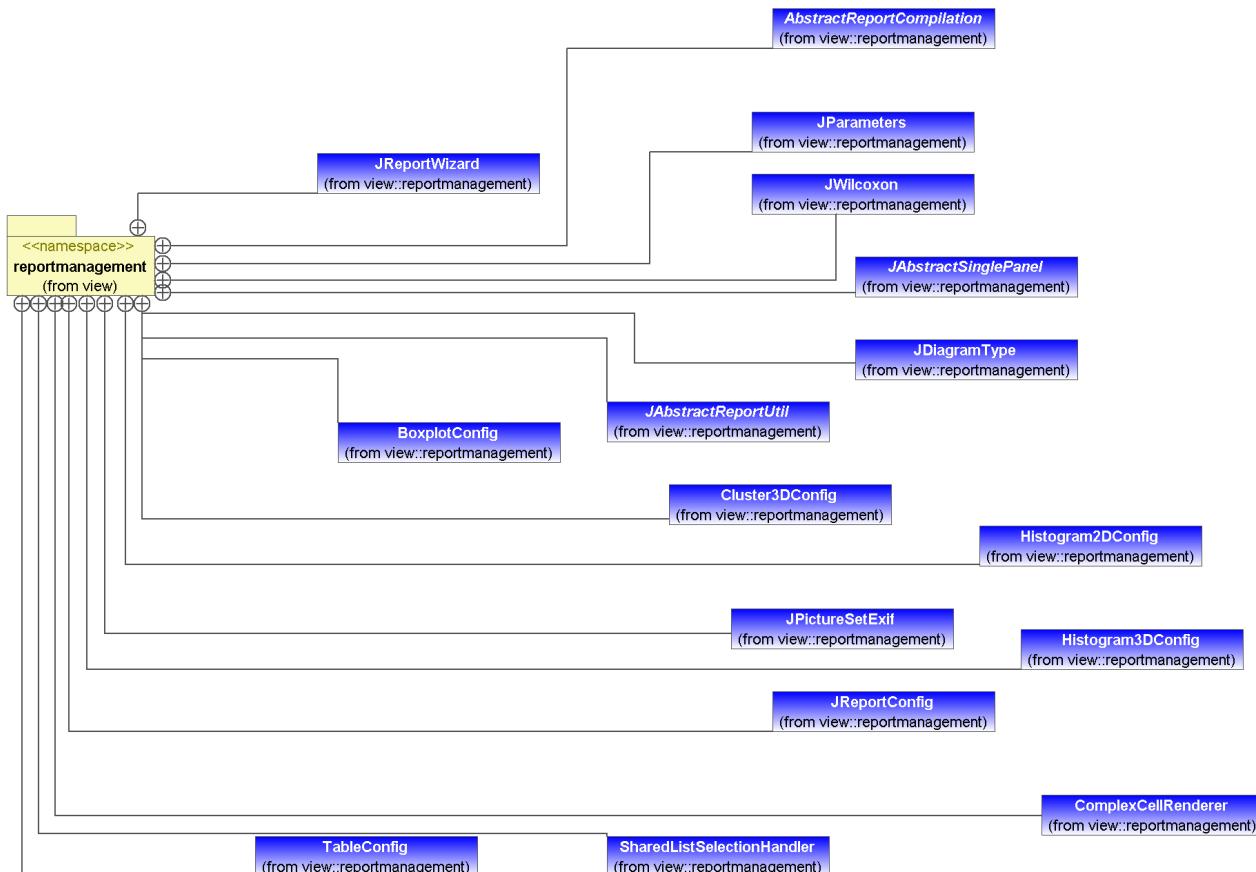
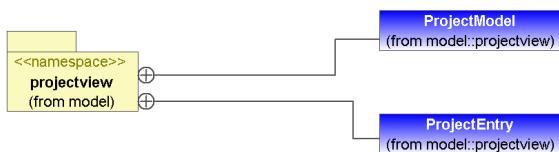
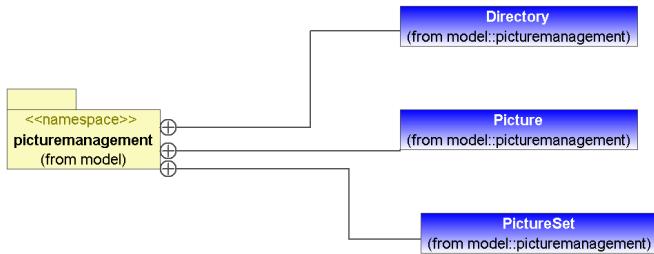




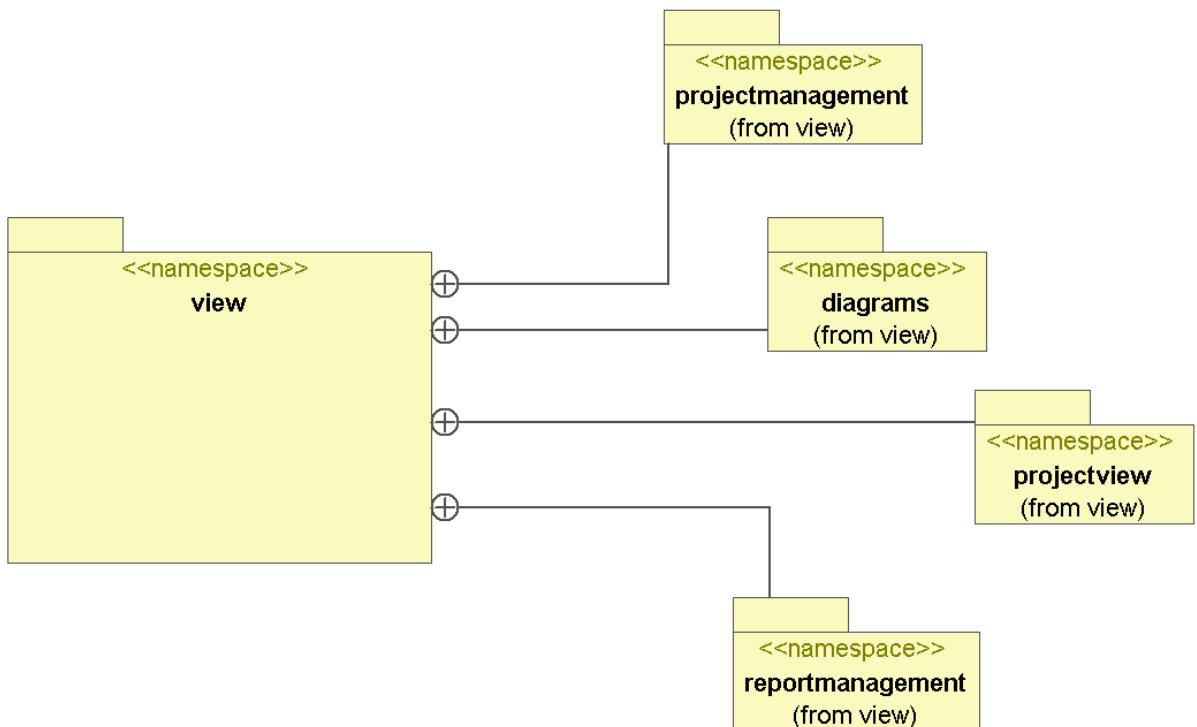
# The model package



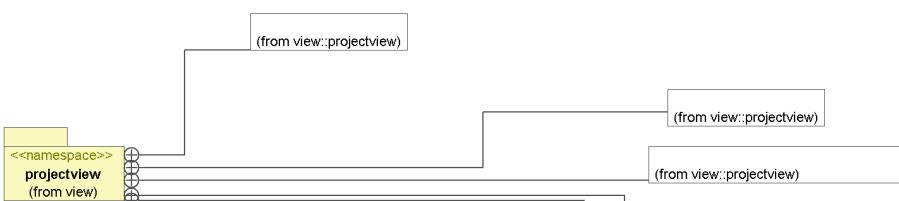
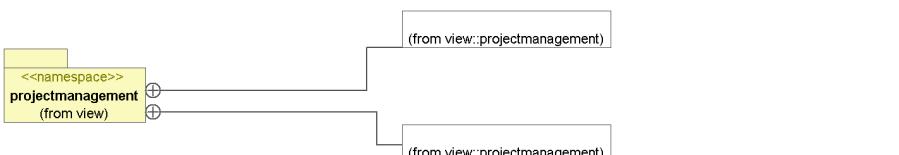
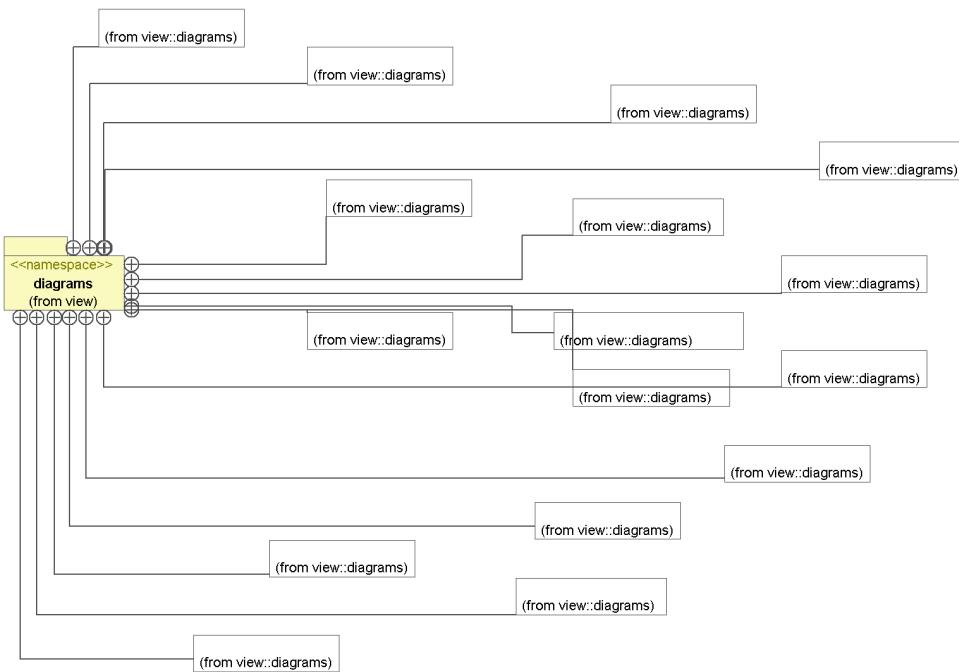
# Detailed Package - model

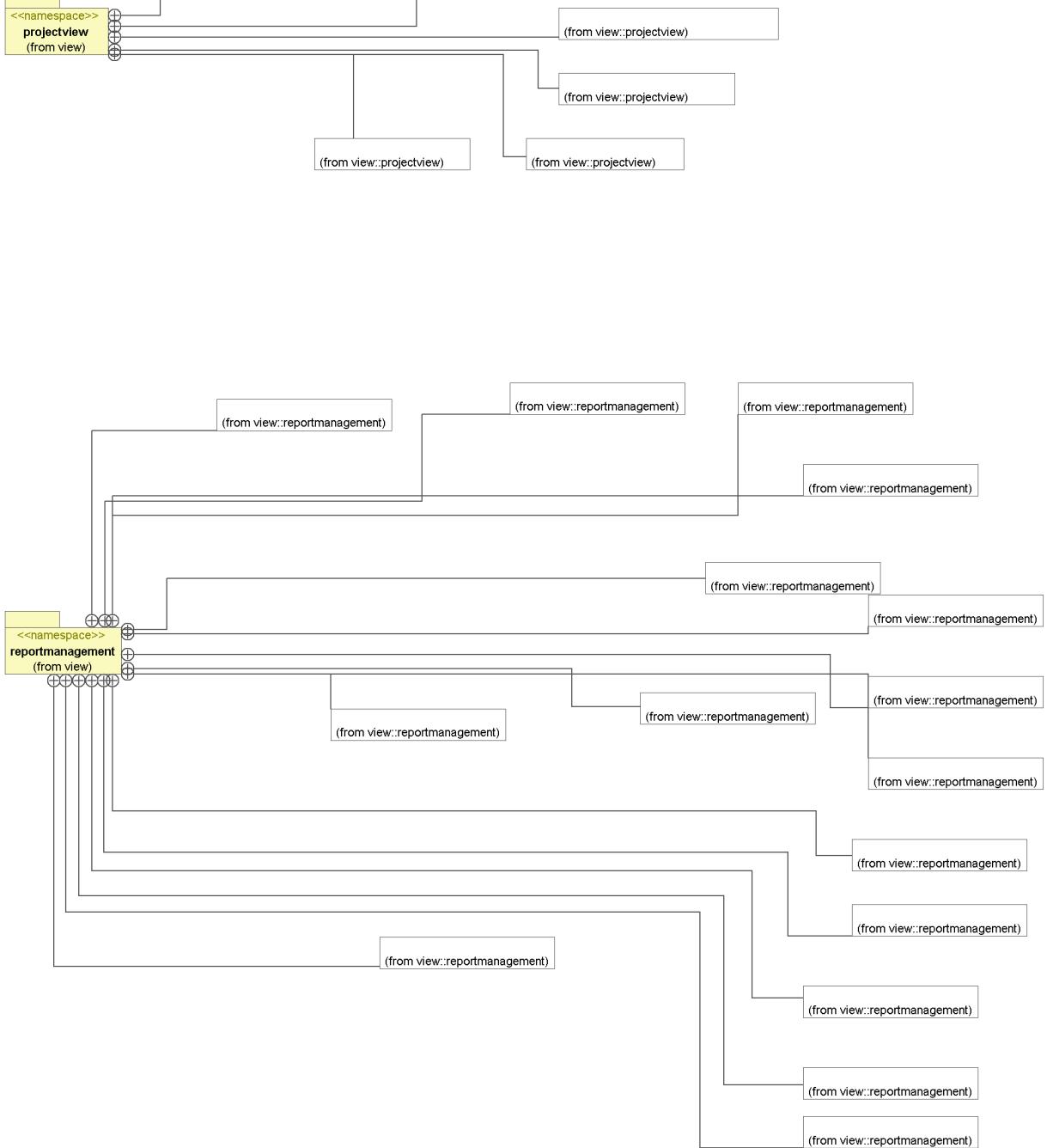


# The view package

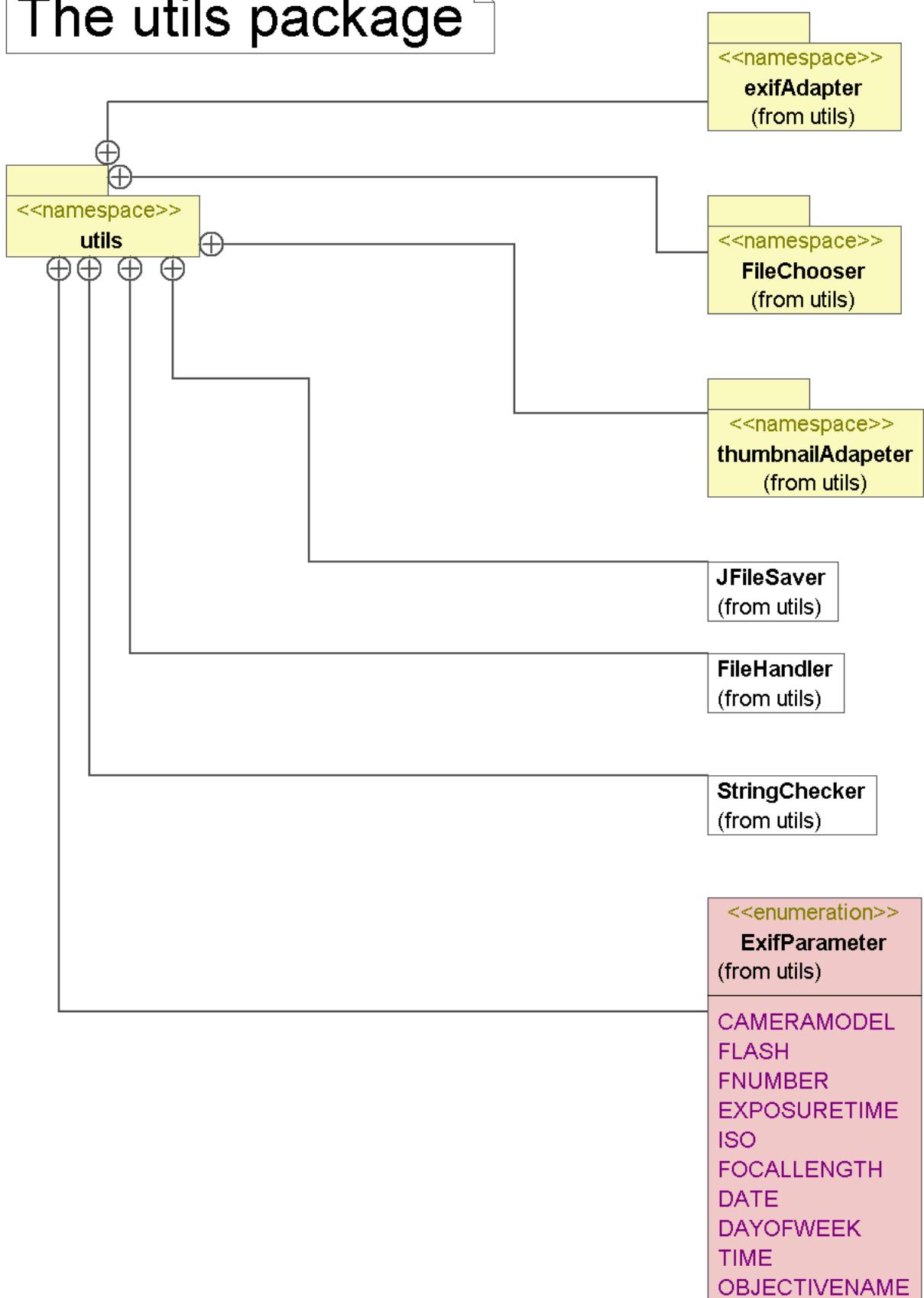


## Detailed Package - view

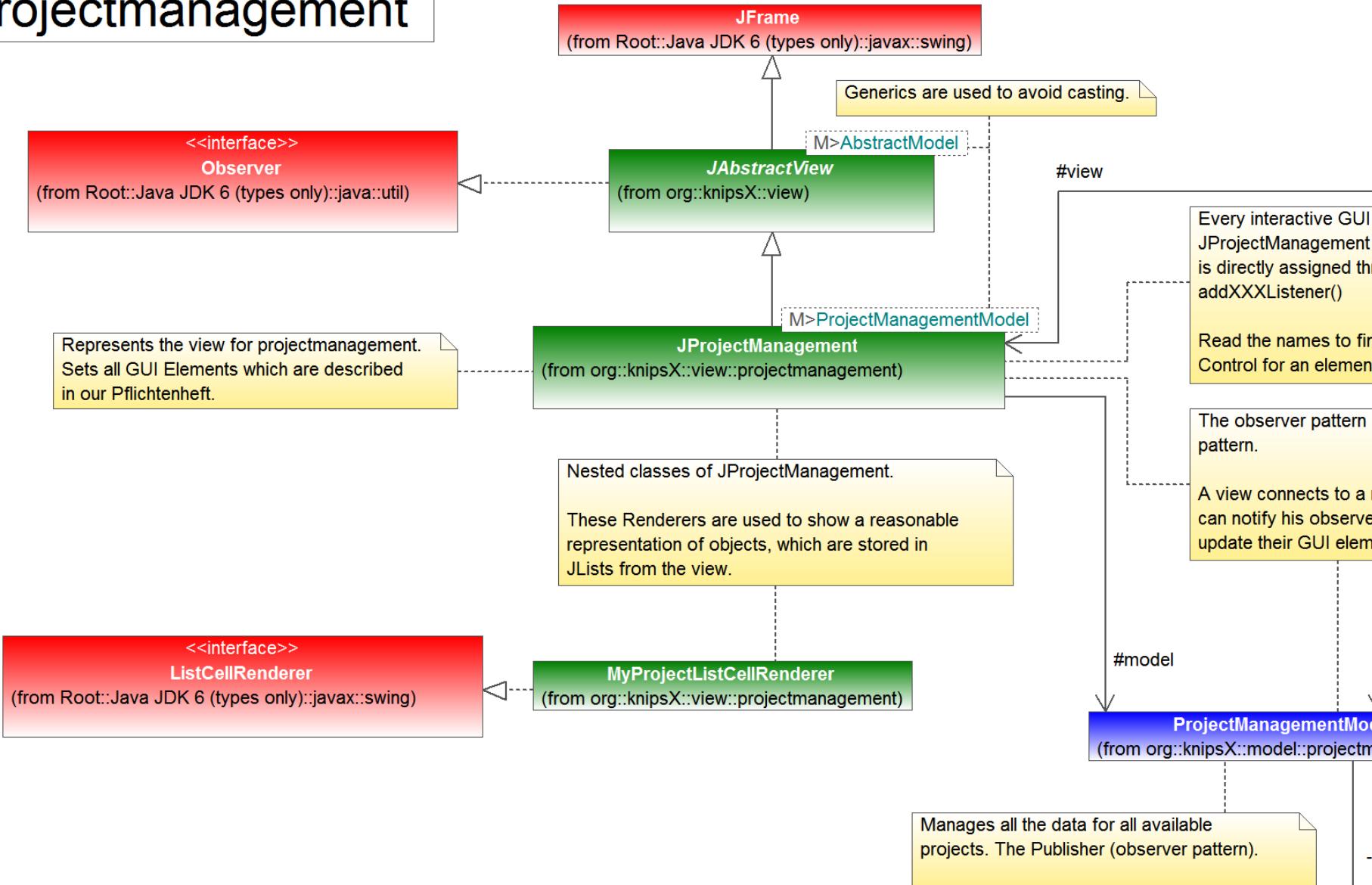


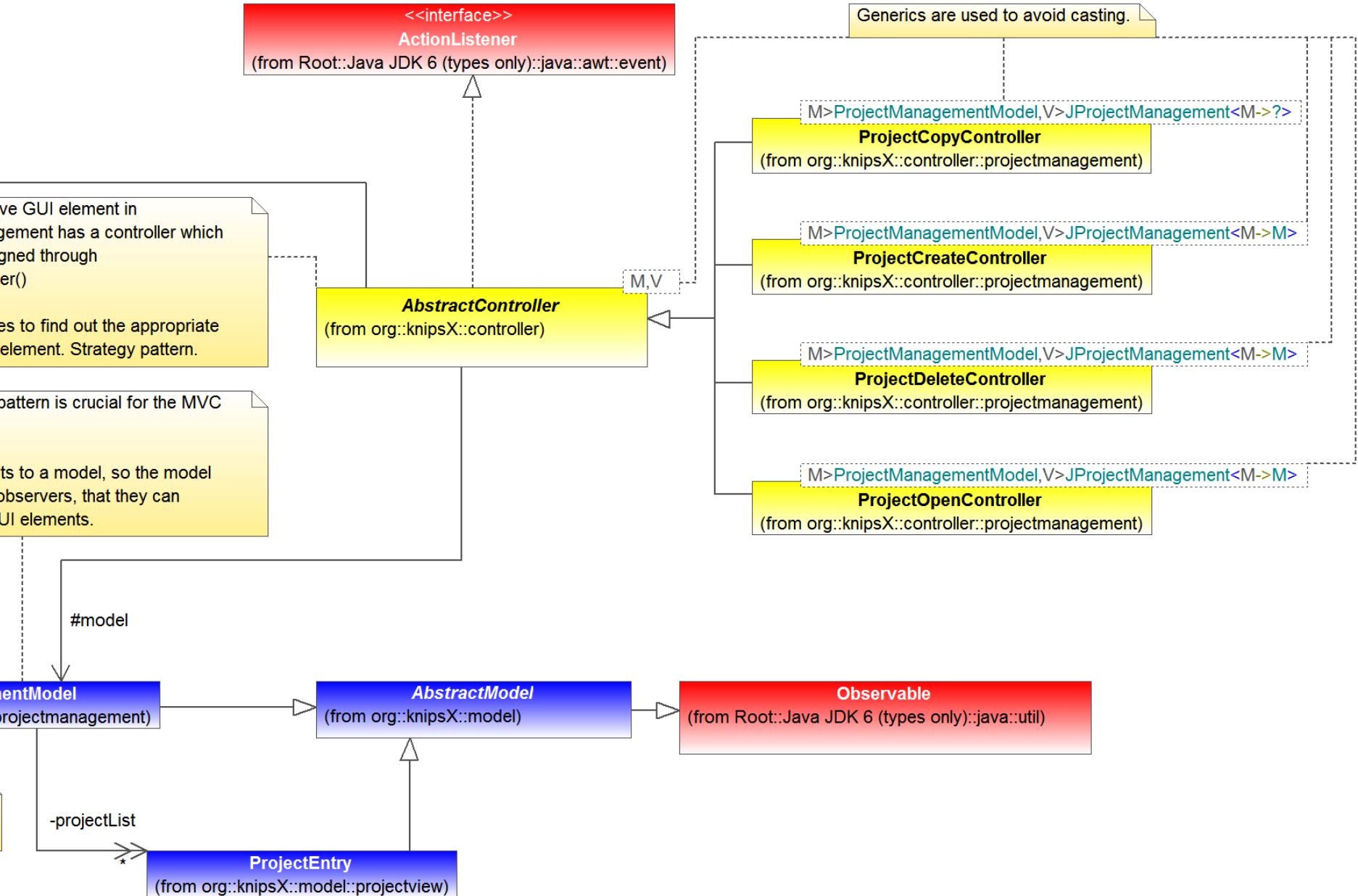


# The utils package

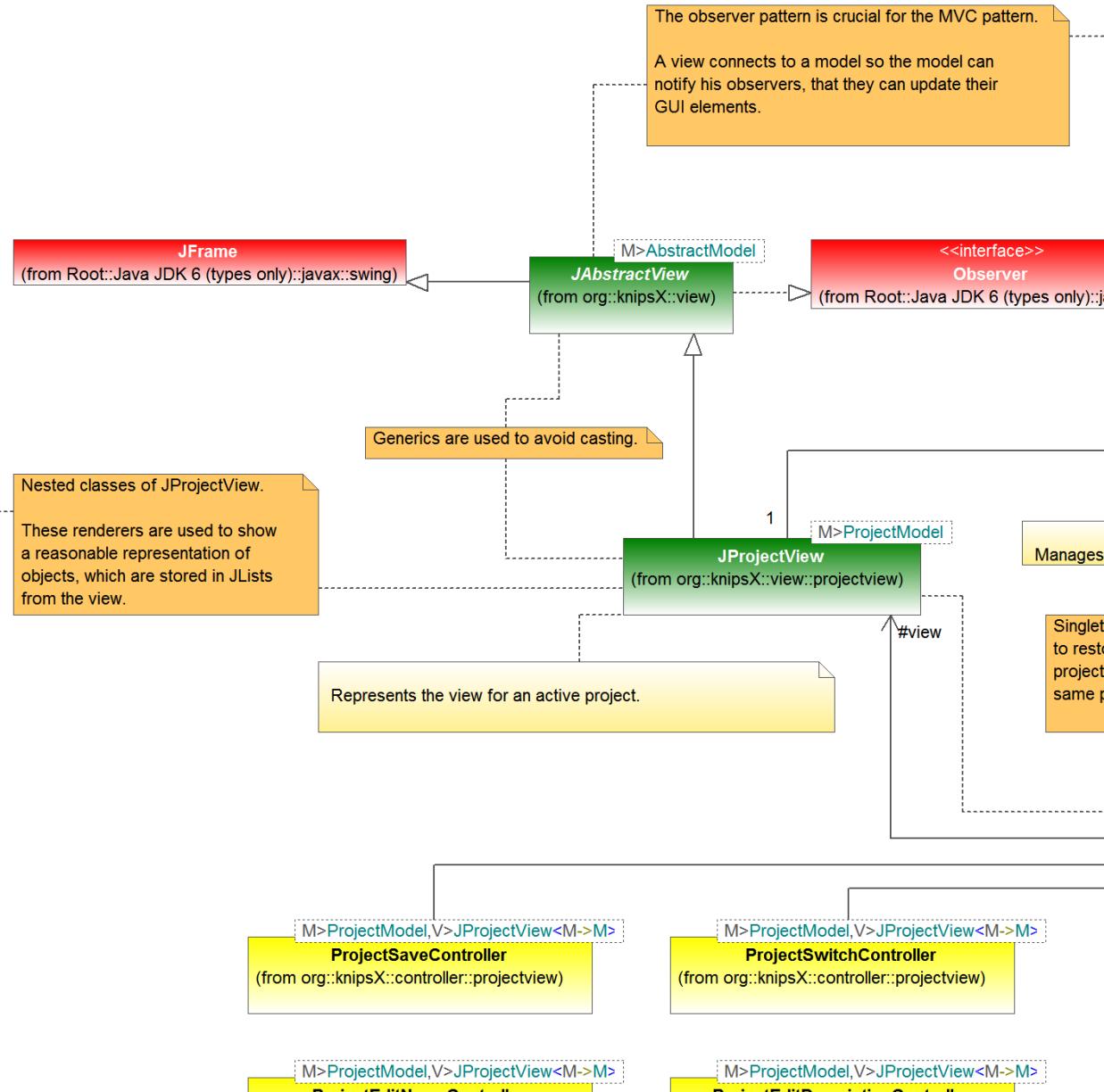
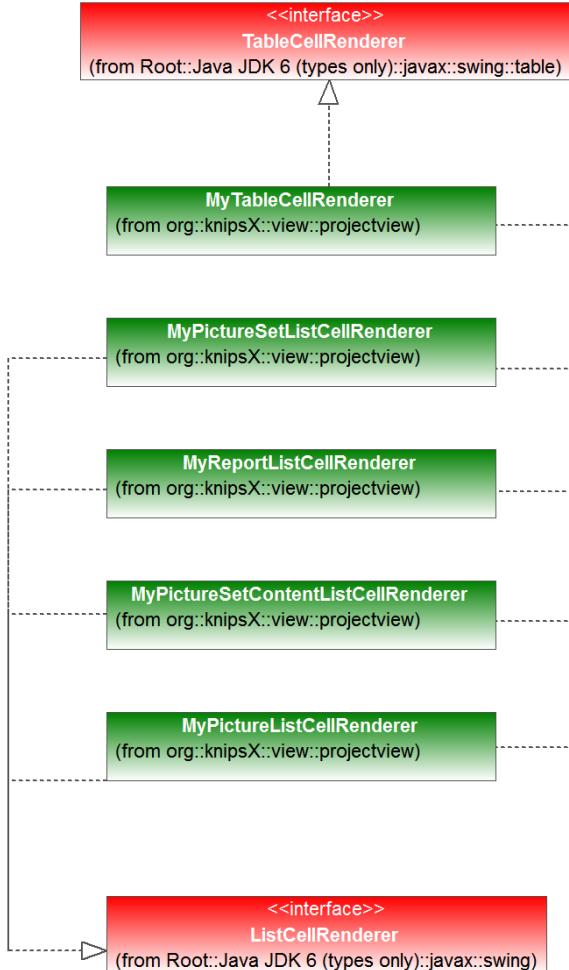


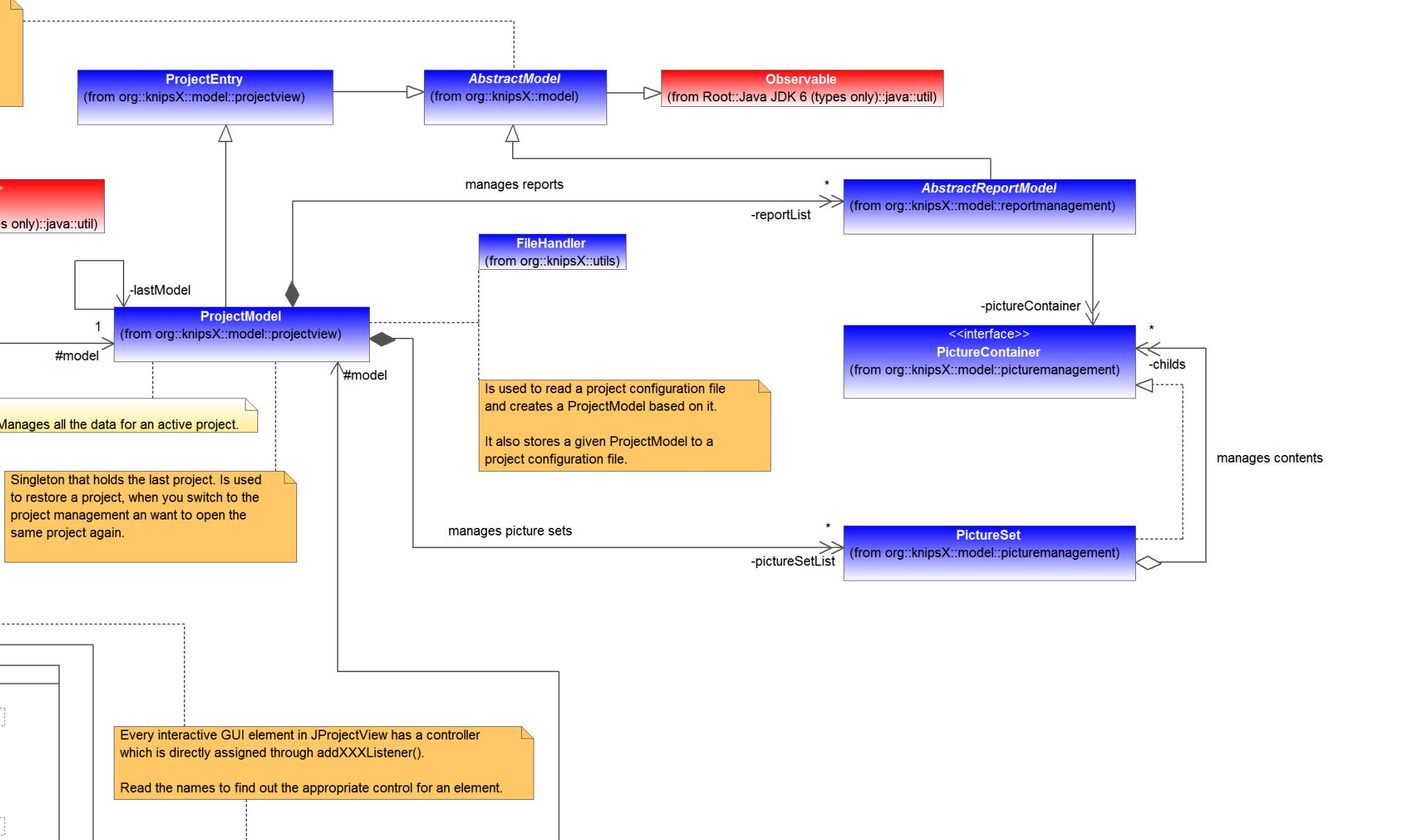
# Projectmanagement

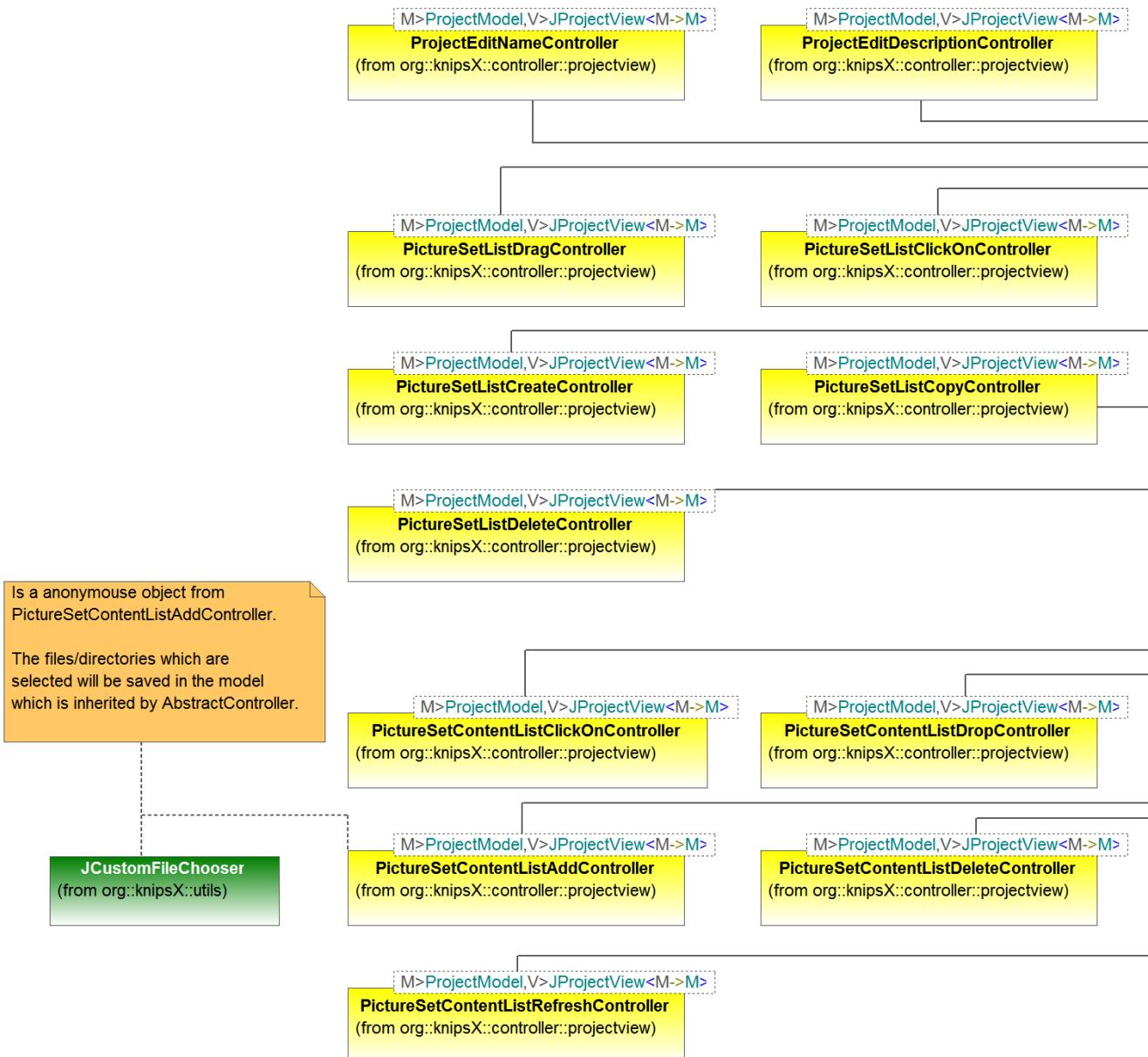


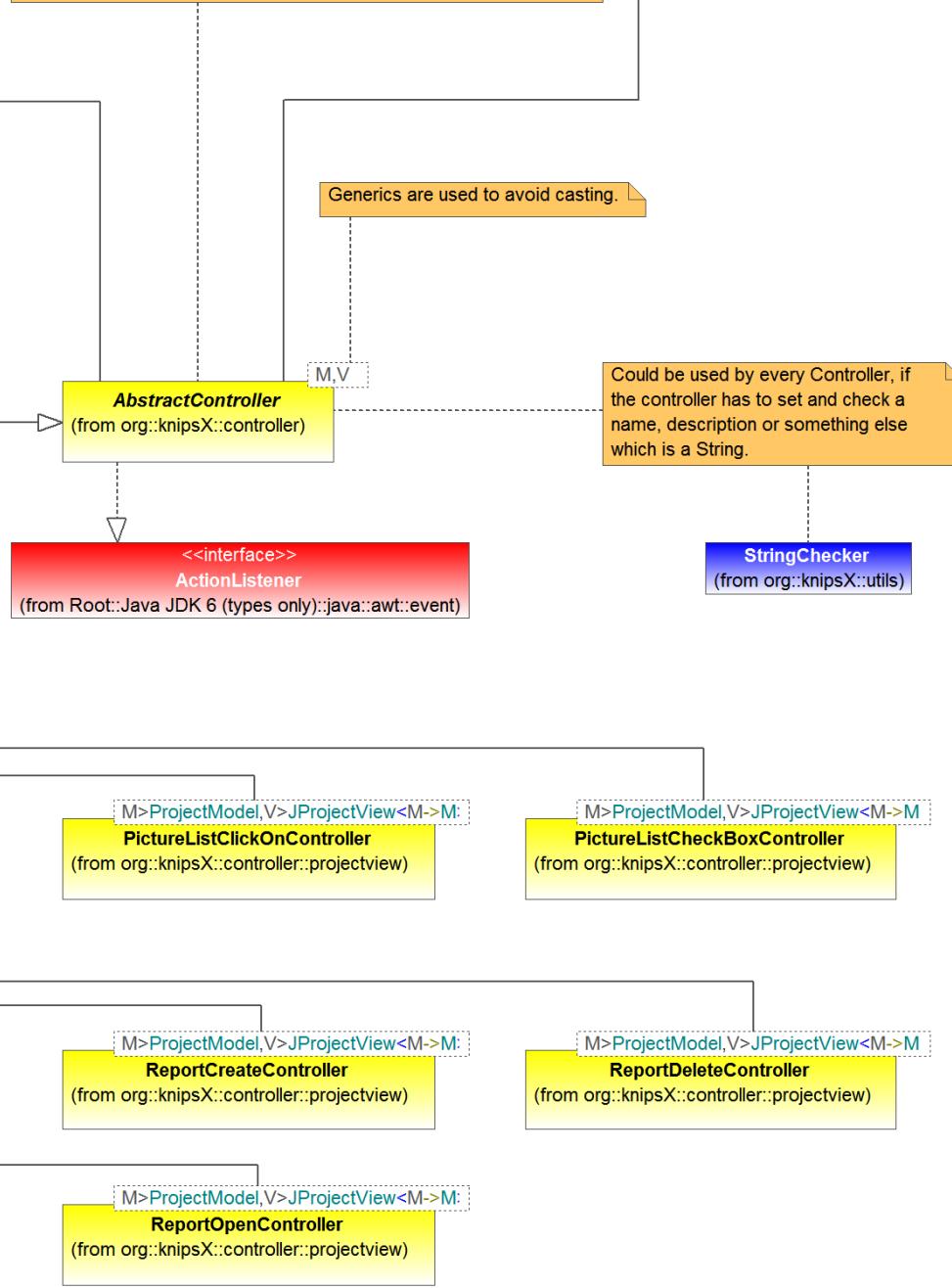


# MVC for the project view

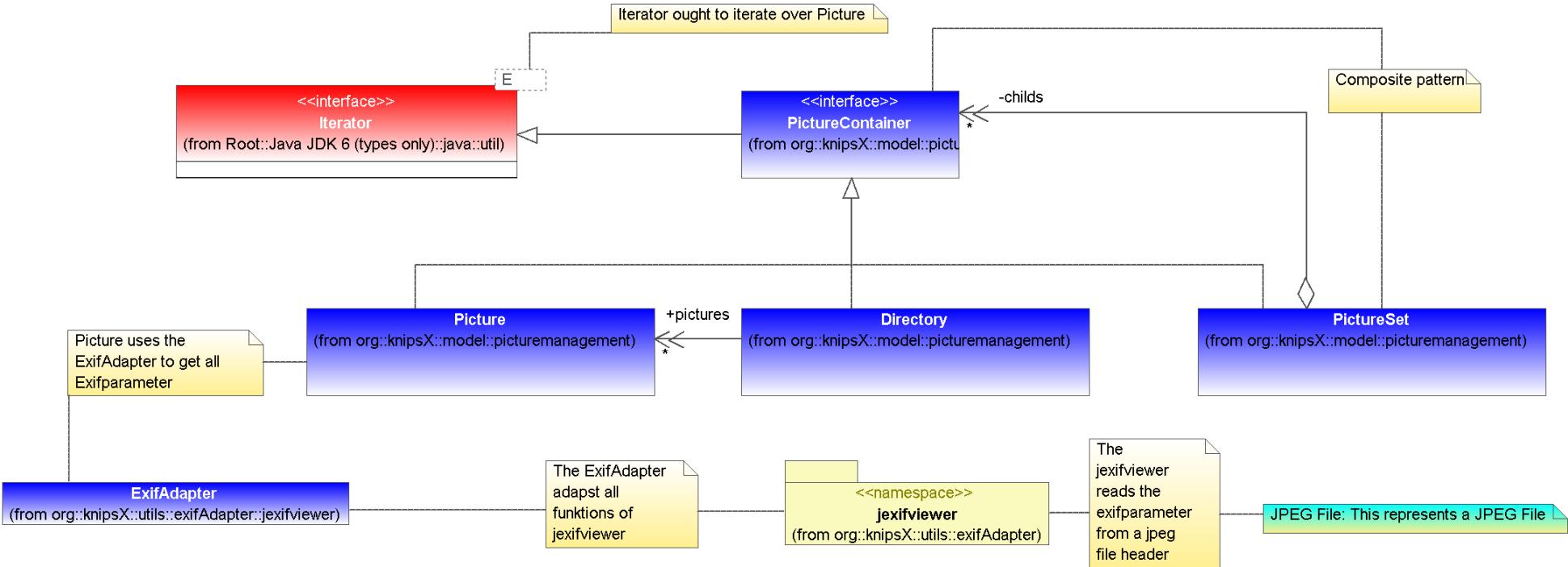




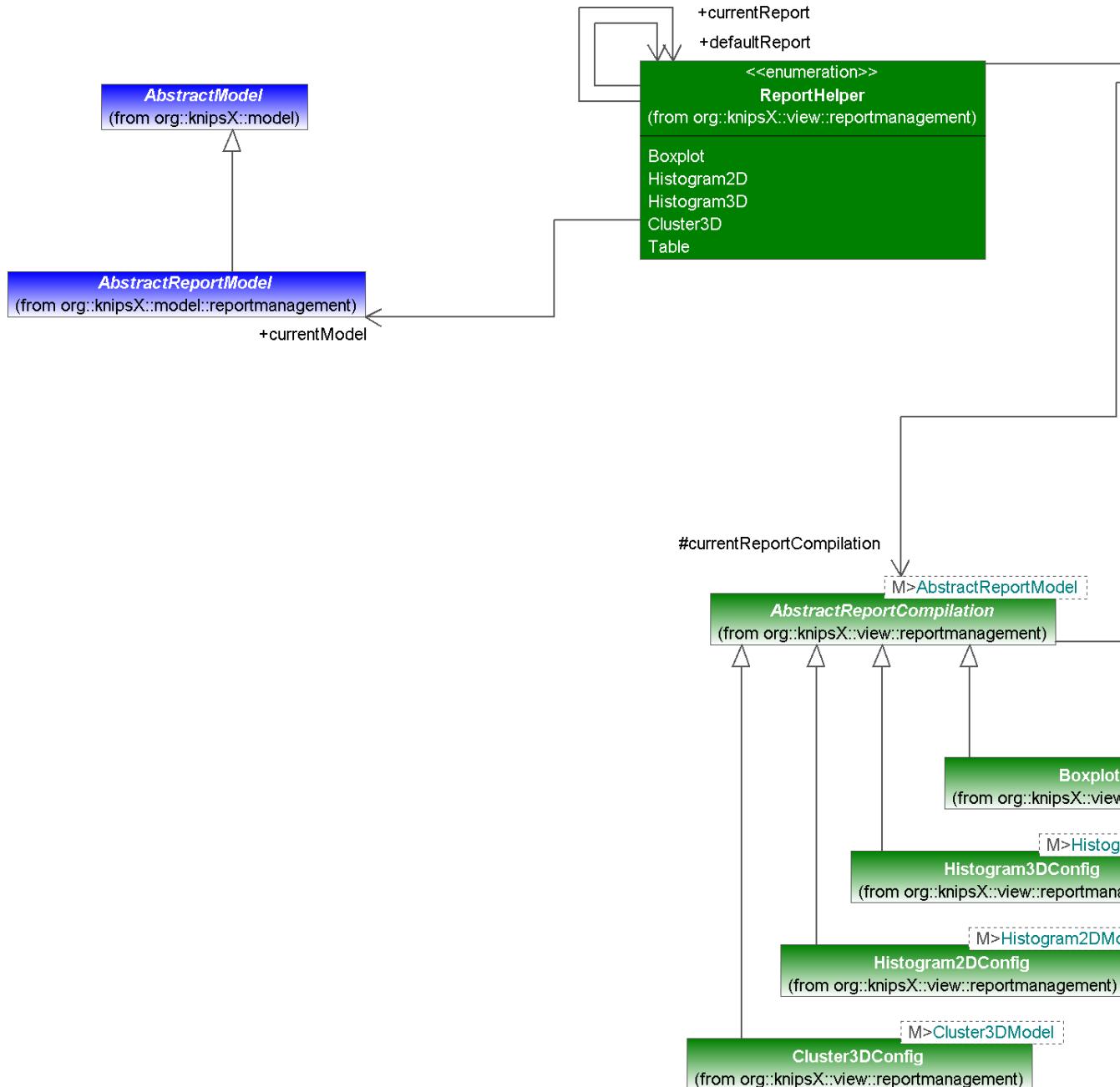


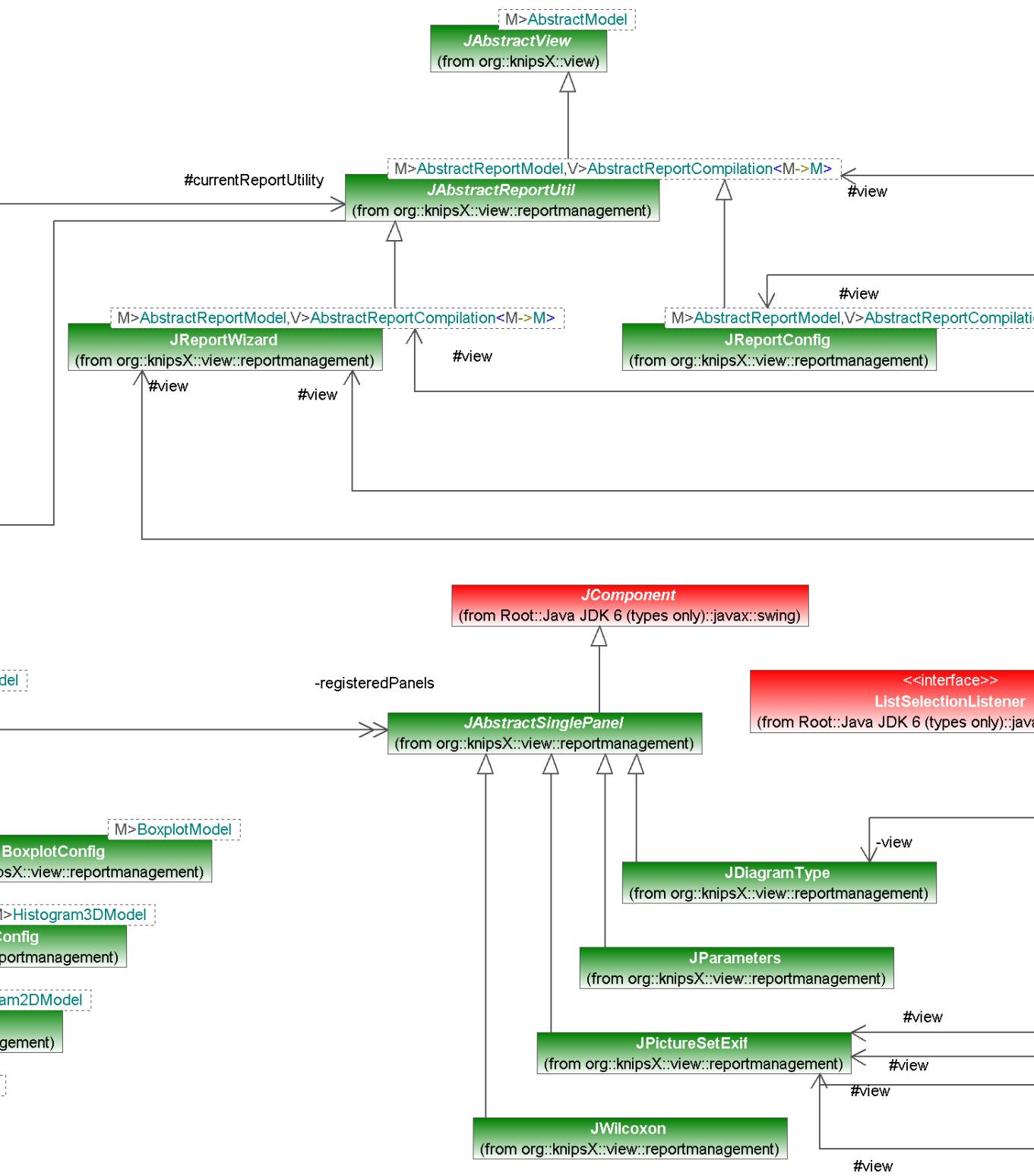


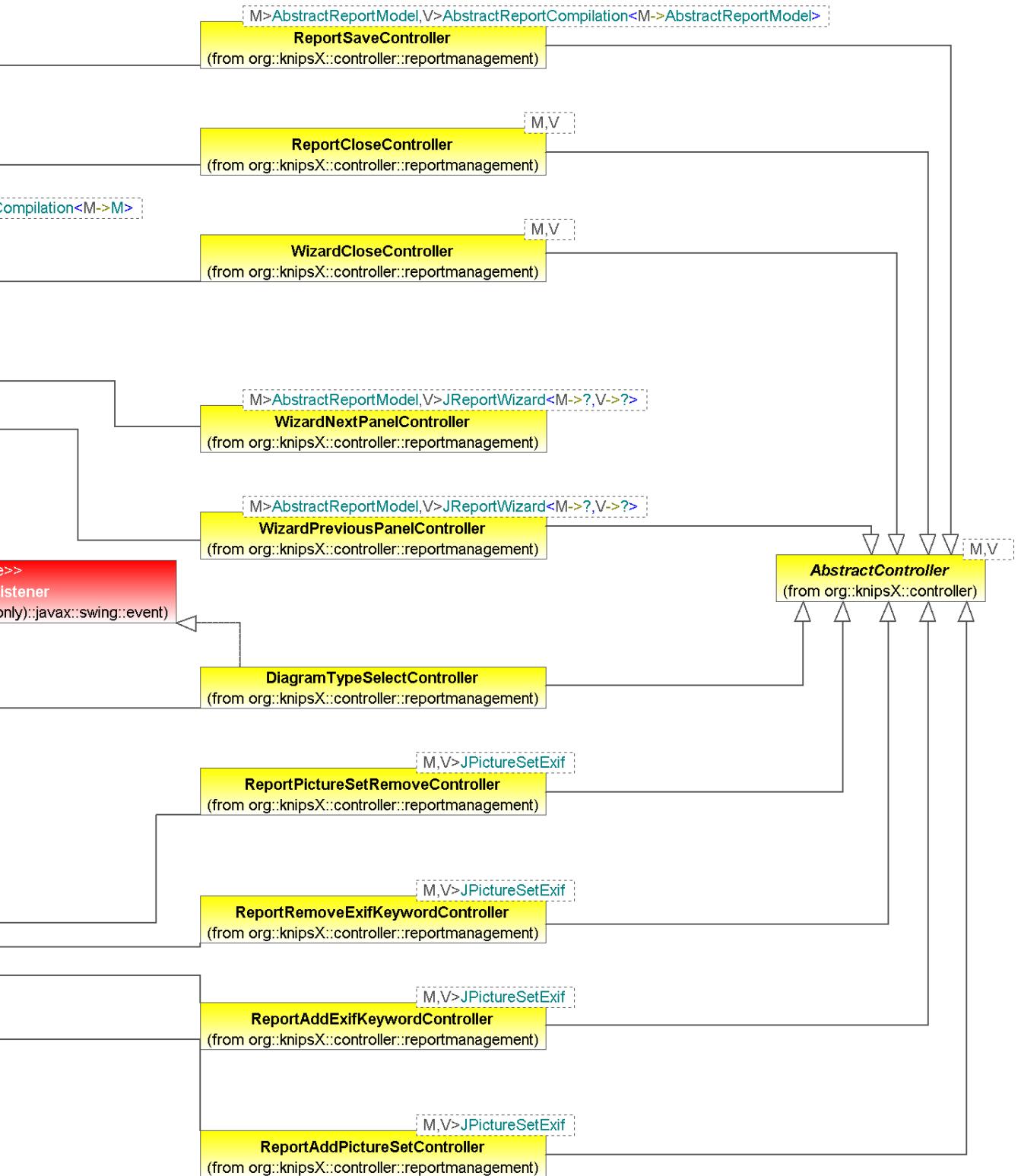
# Picturemanagement



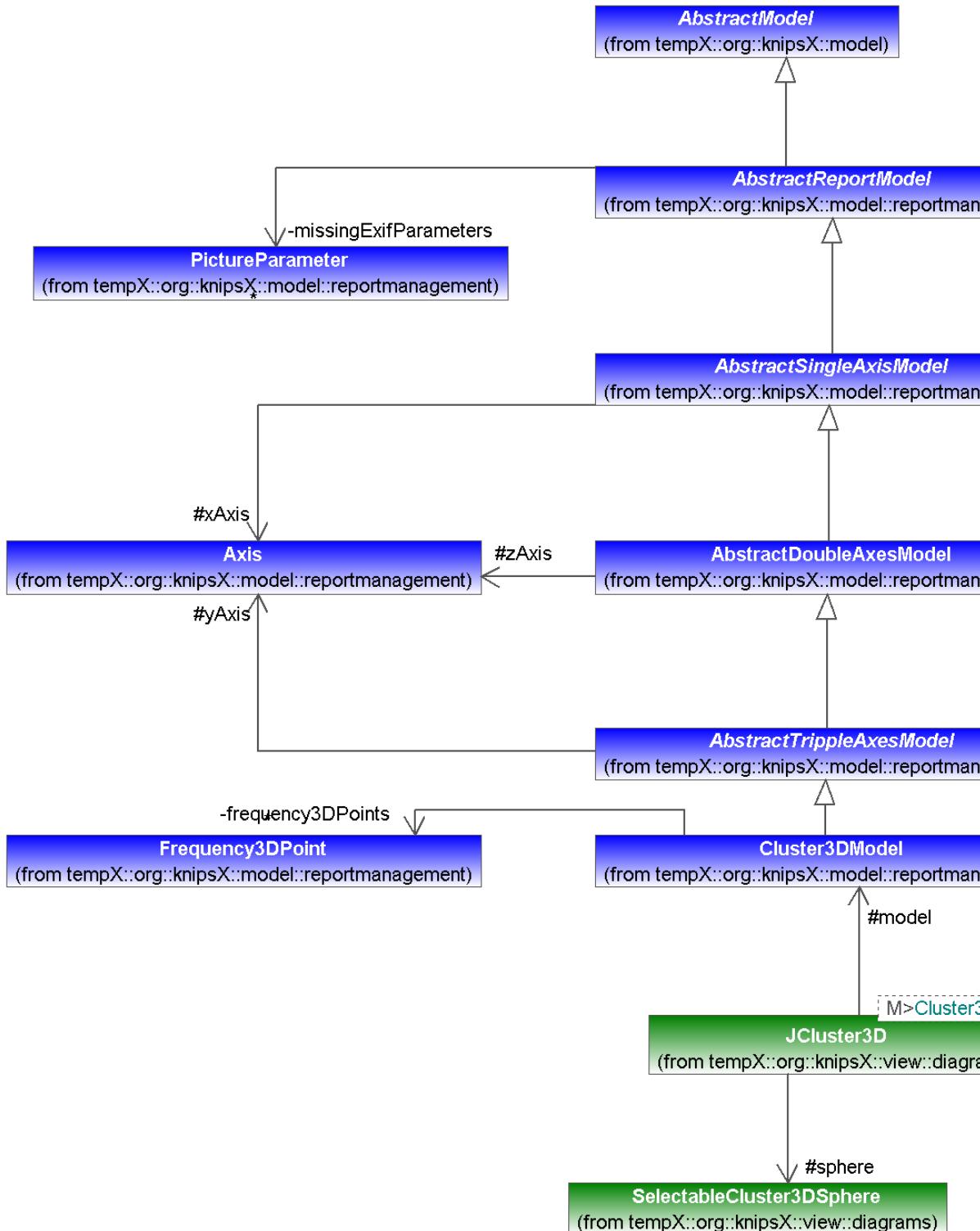
# MVC for the report management

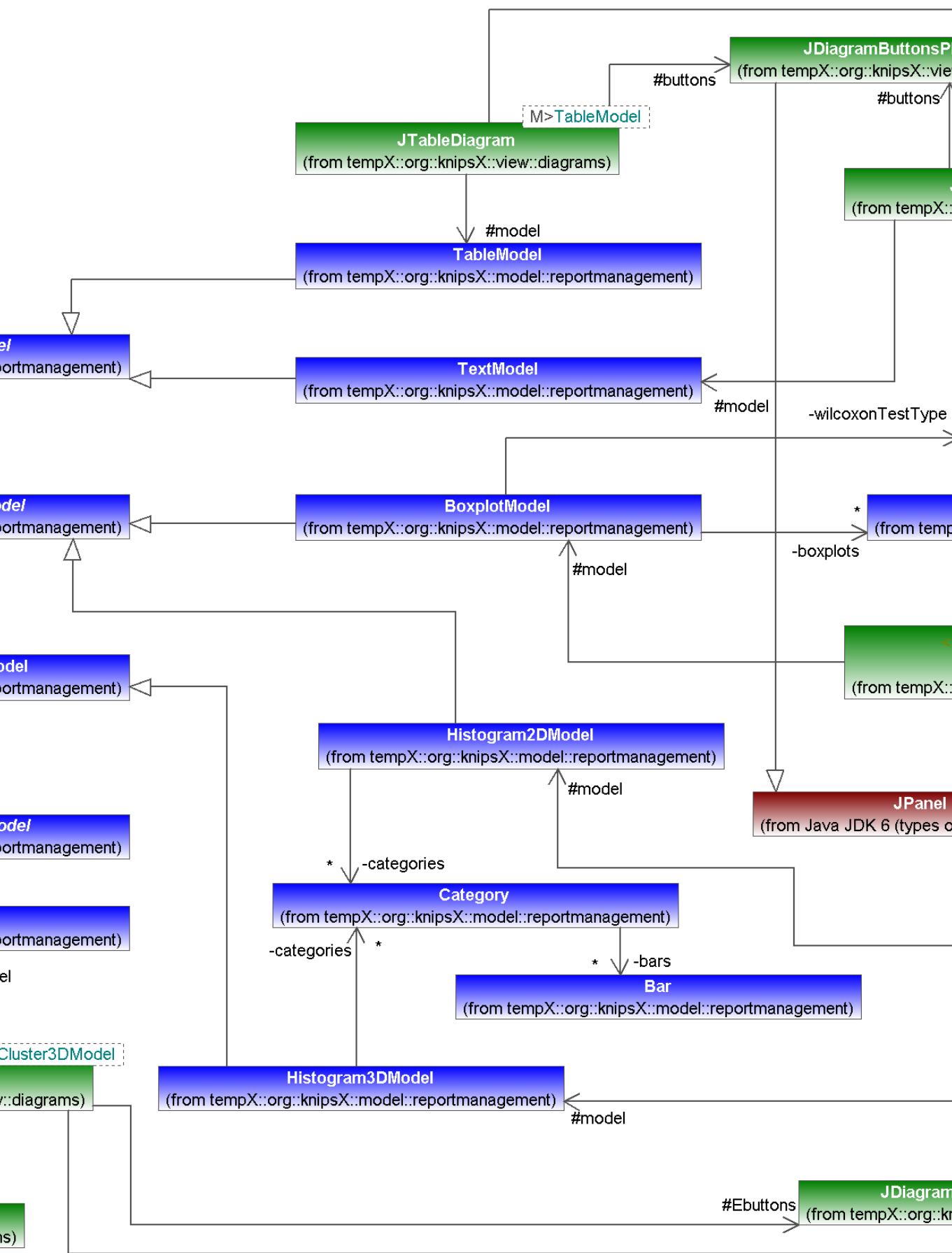


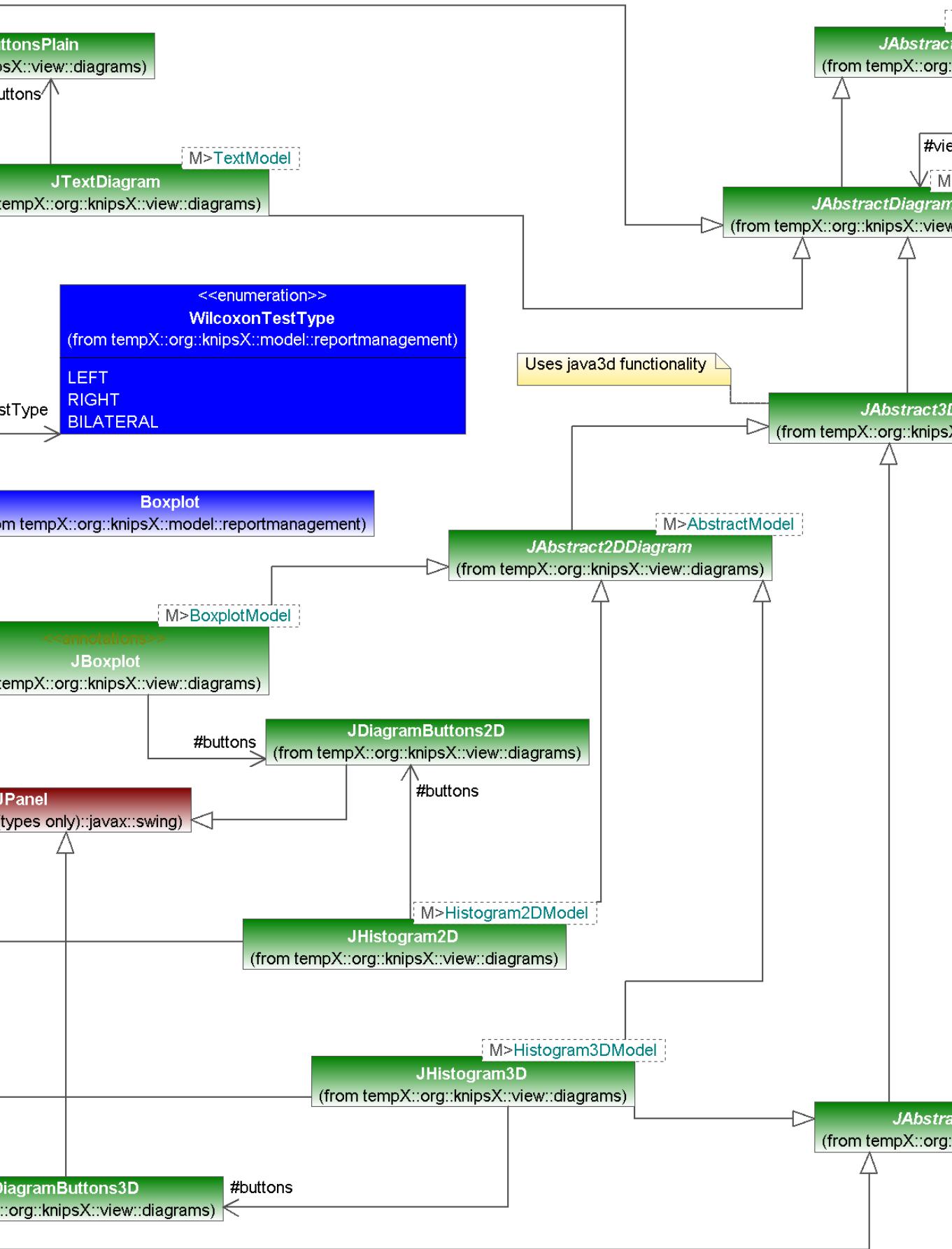


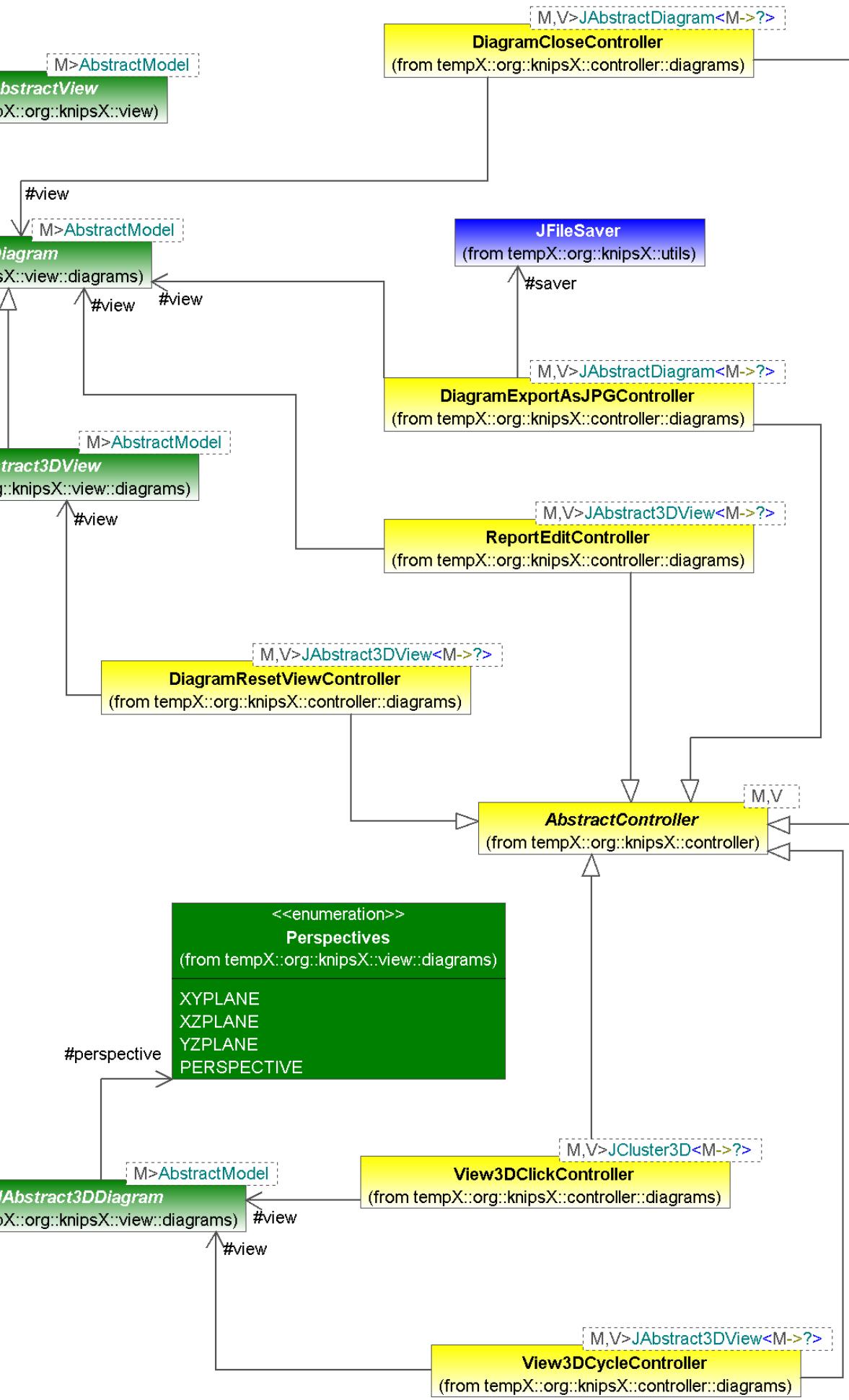


# MVC for the Diagrams

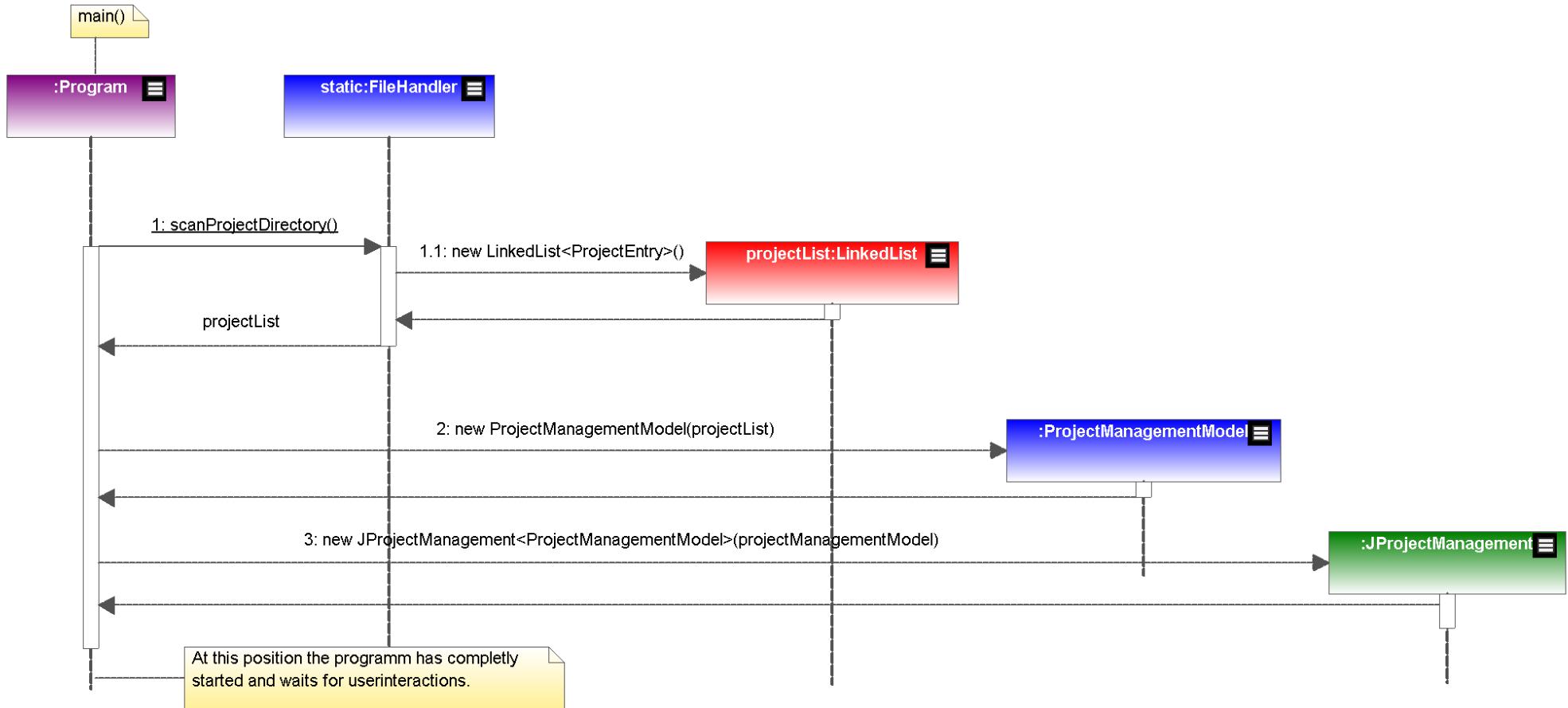




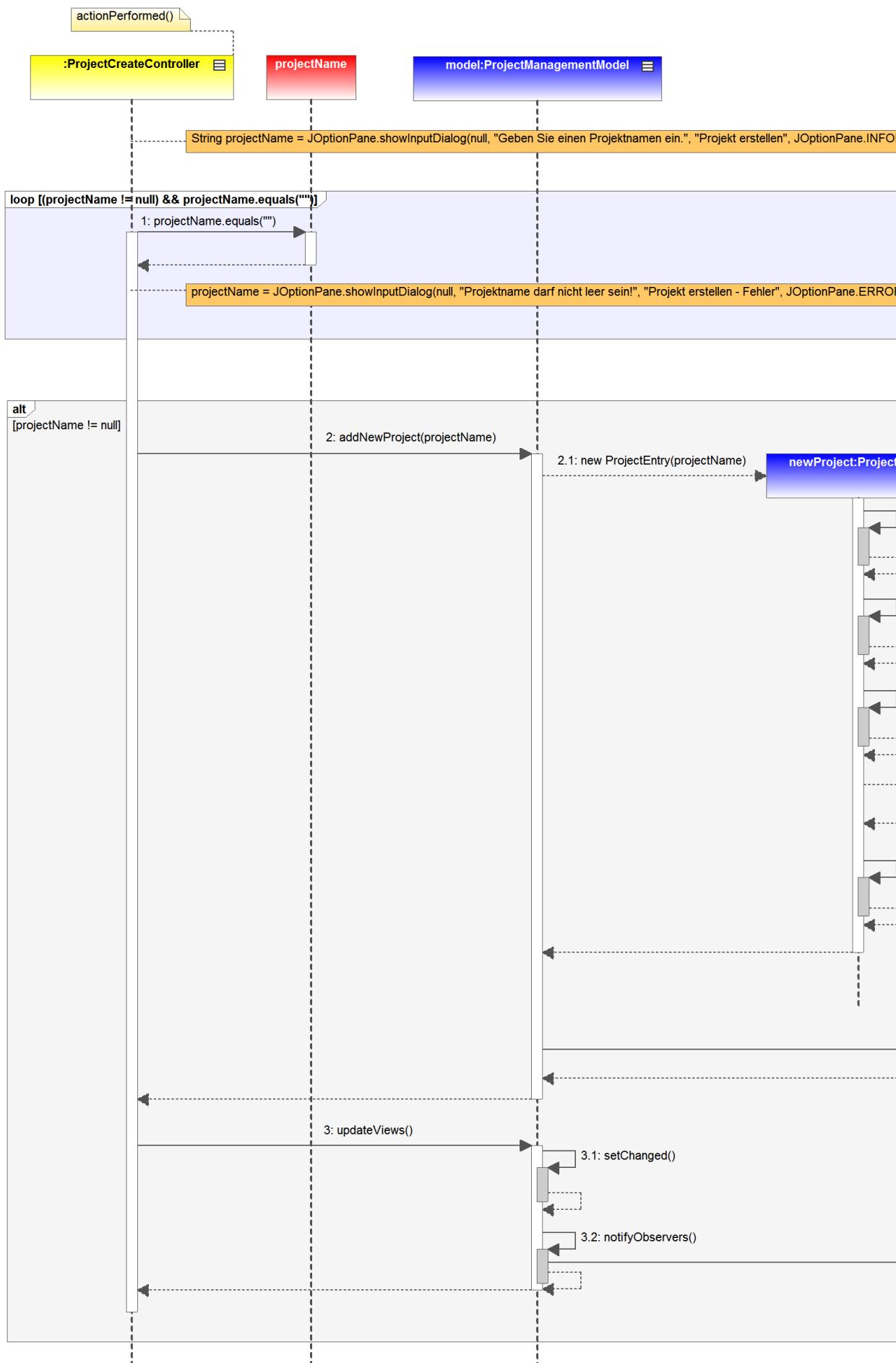




# Sequence diagram of the program start



# Sequence diagram for the action "create project"



pane.INFORMATION\_MESSAGE);

ne.ERROR\_MESSAGE)

ct:ProjectEntry

2.1.1: generateFreeProjectId()

int id

2.1.2: this.projectName = projectName

2.1.3: this.projectDescription = ""

2.1.4: new GregorianCalendar()

creationDate:GregorianCalendar

2.1.5: getPathForProject()

String path

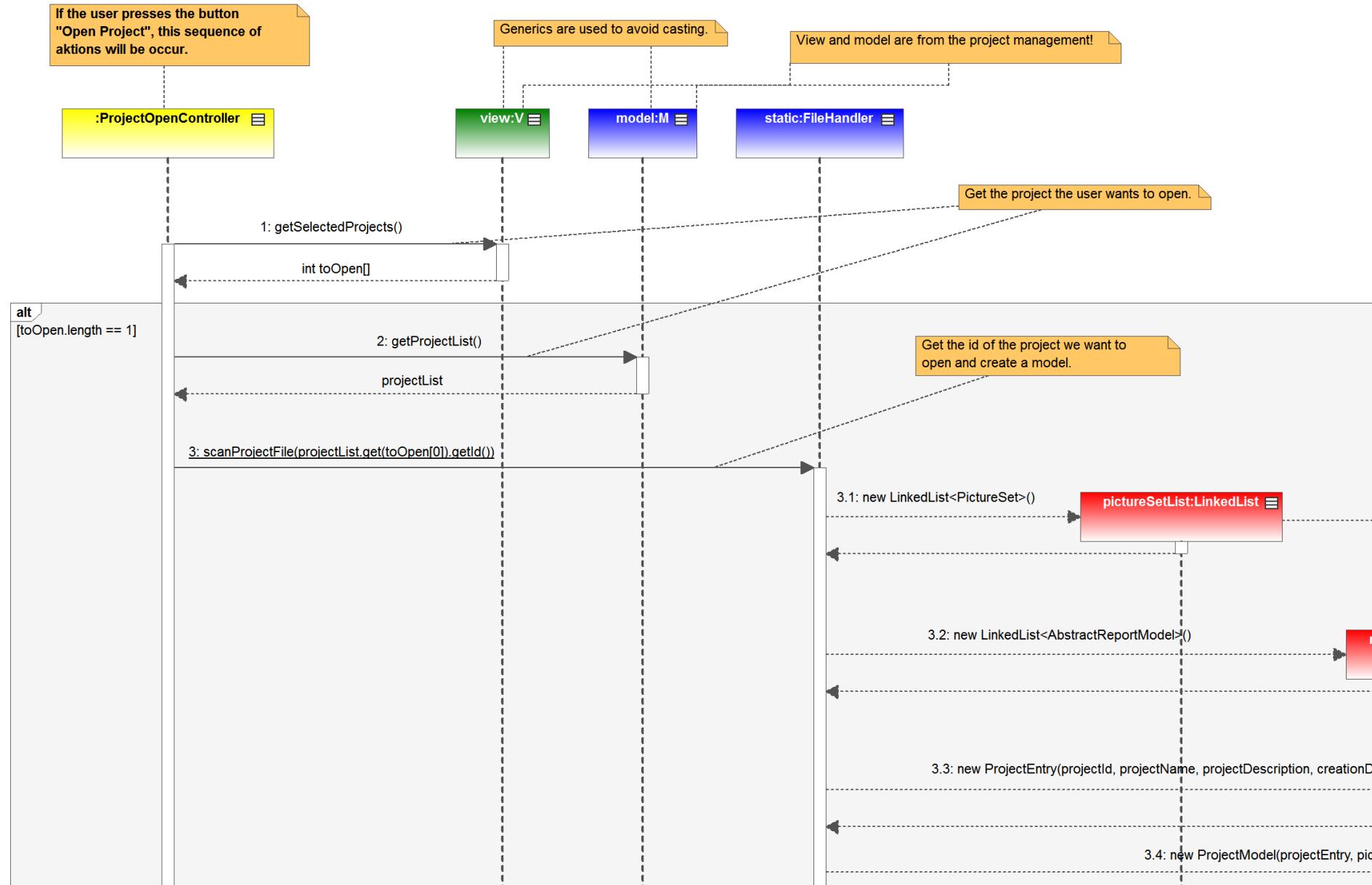
2.2: createNewProjectFile(newProject)

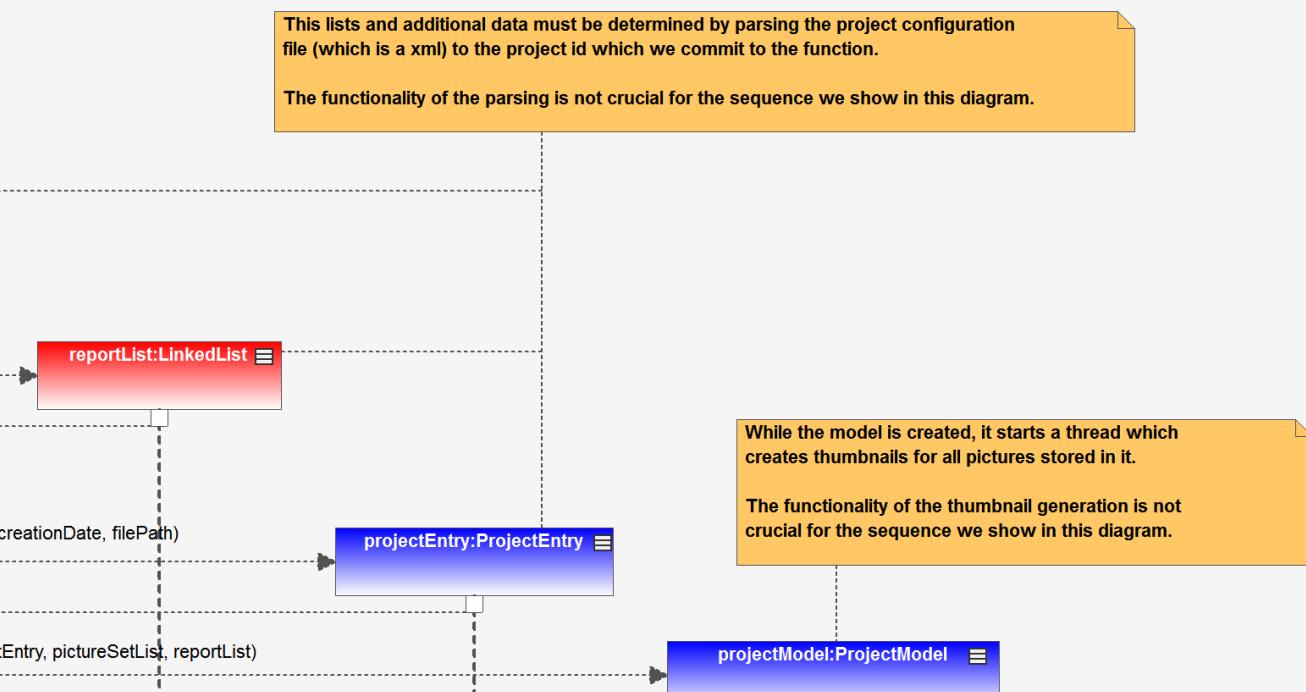
static:FileHandler

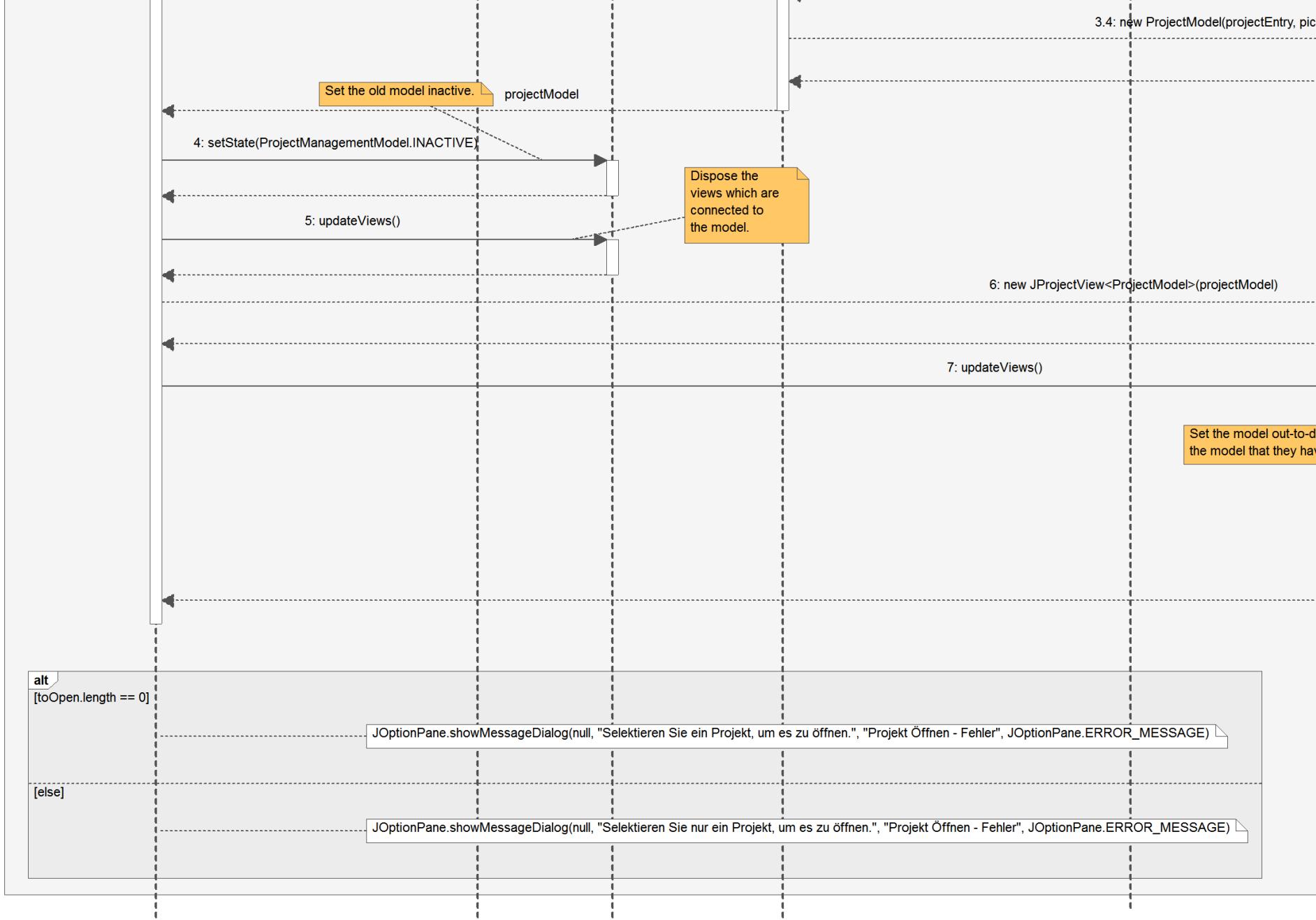
:JProjectManagement

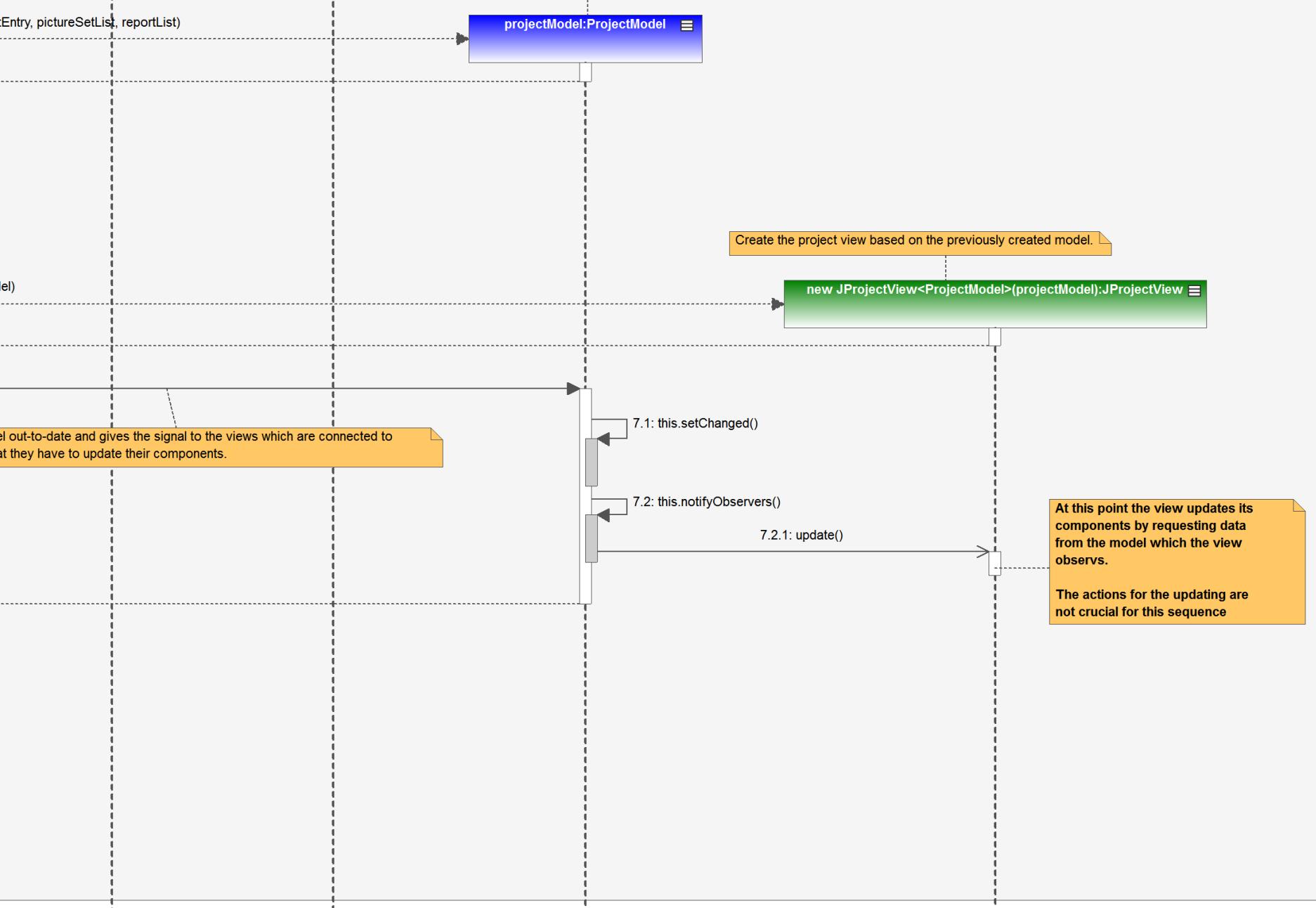
3.2.1: update()

# Sequence diagram for the action "project view opens"

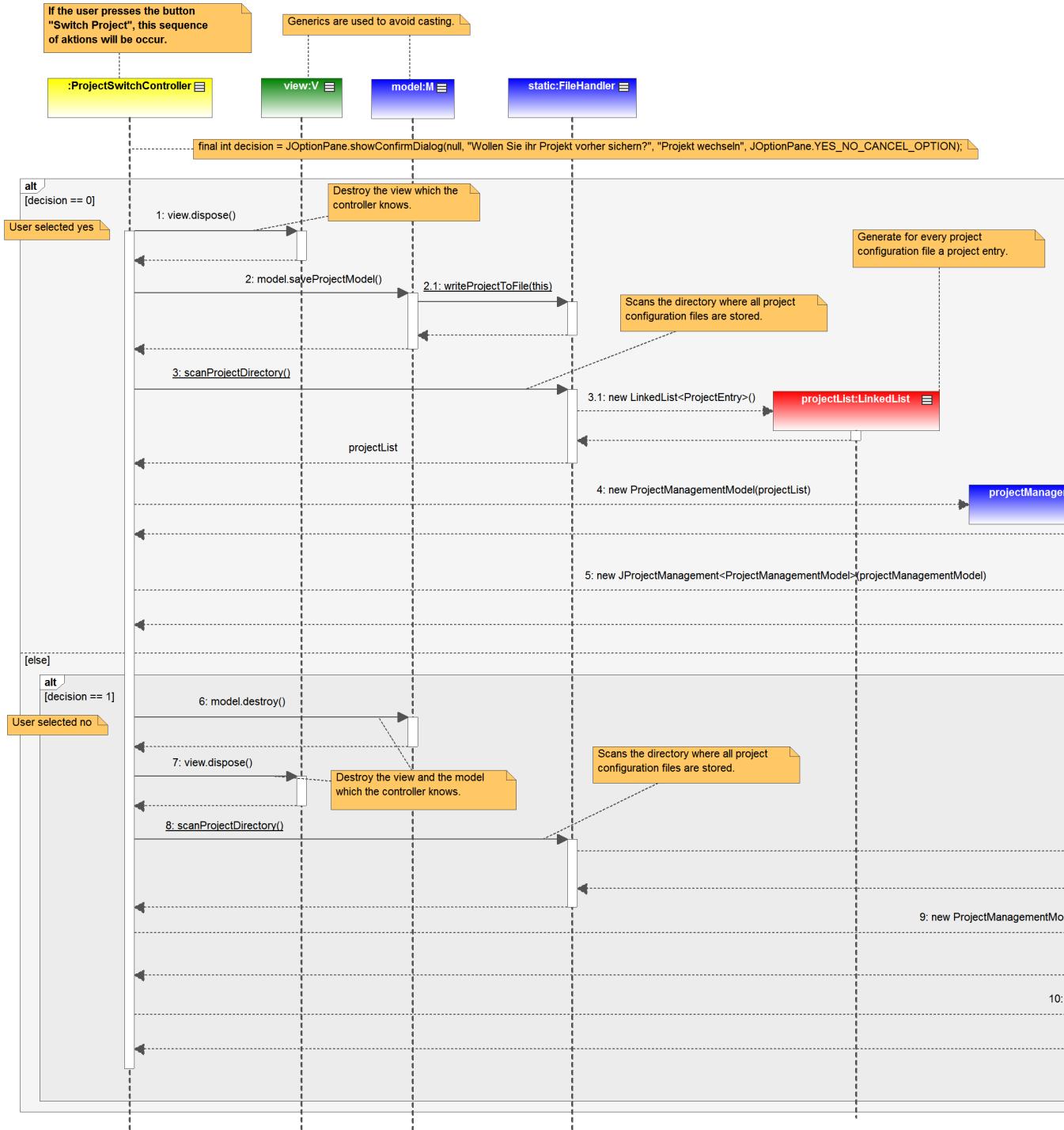


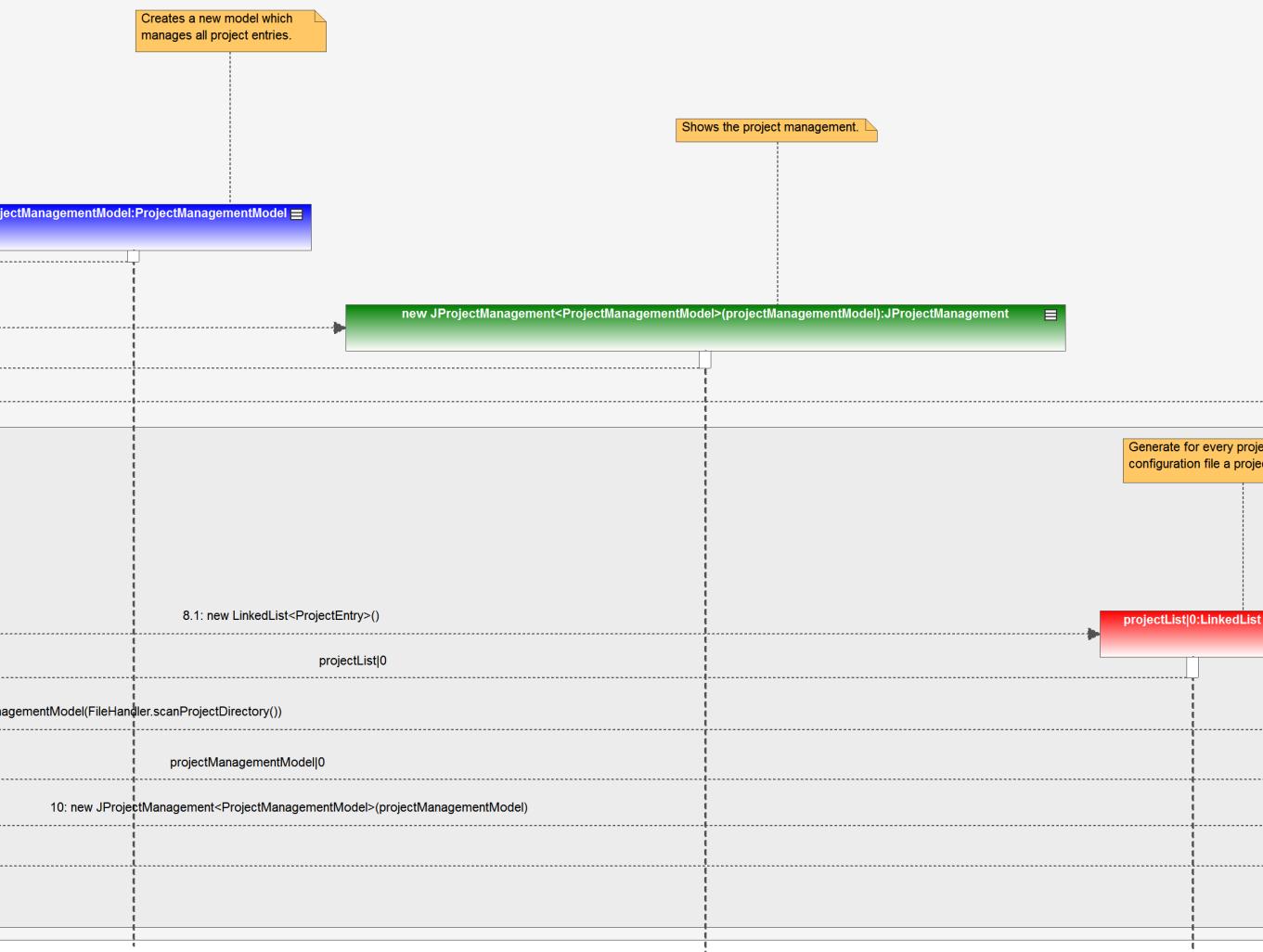


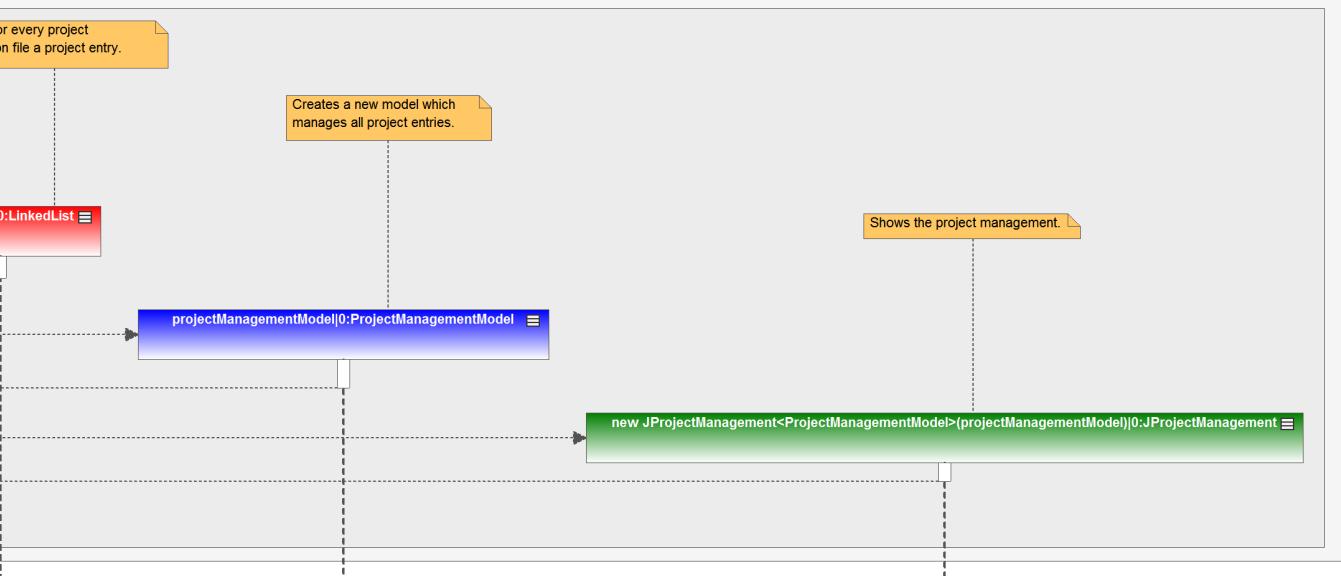




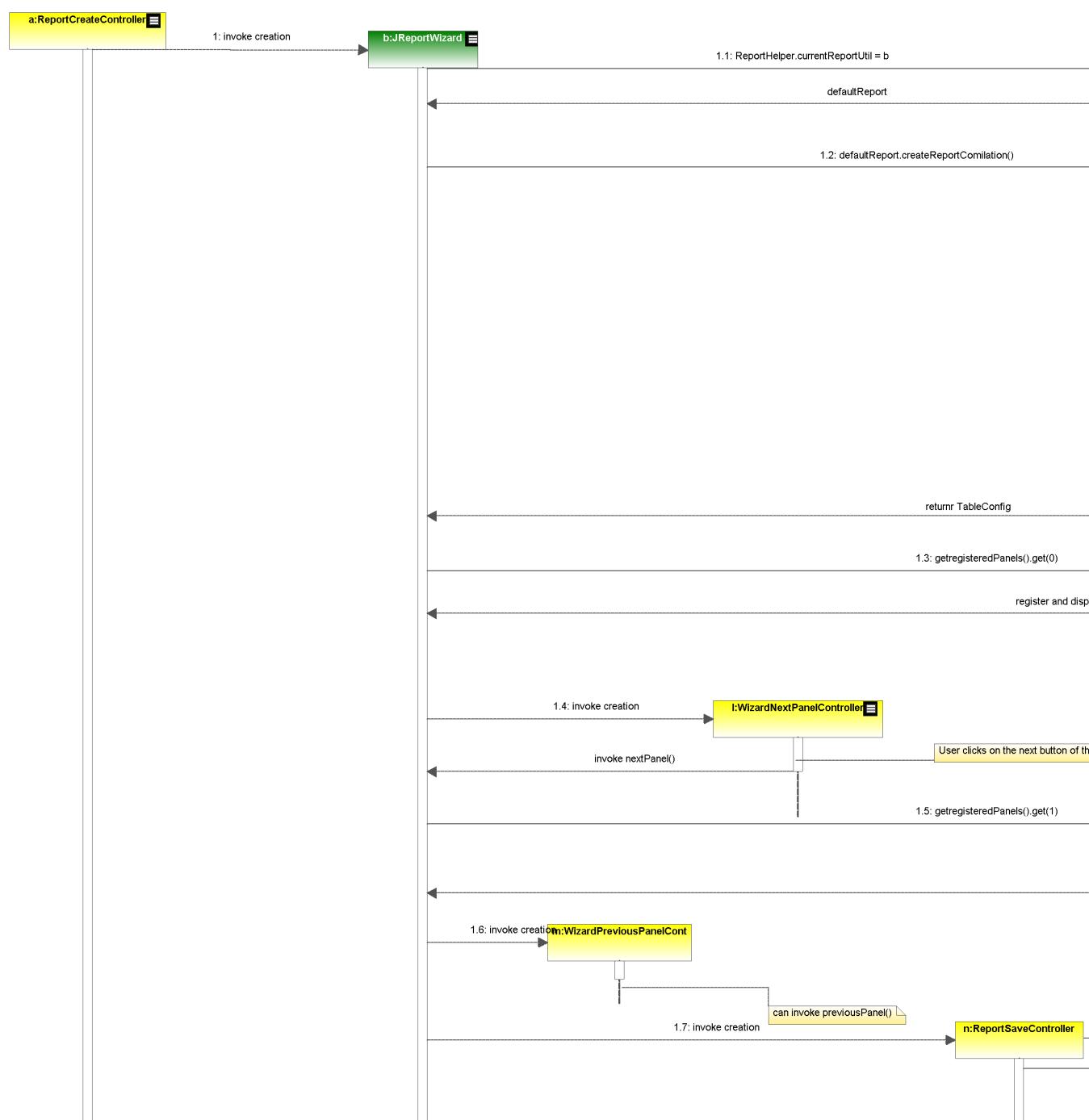
# Sequence diagram for the action "project switch"

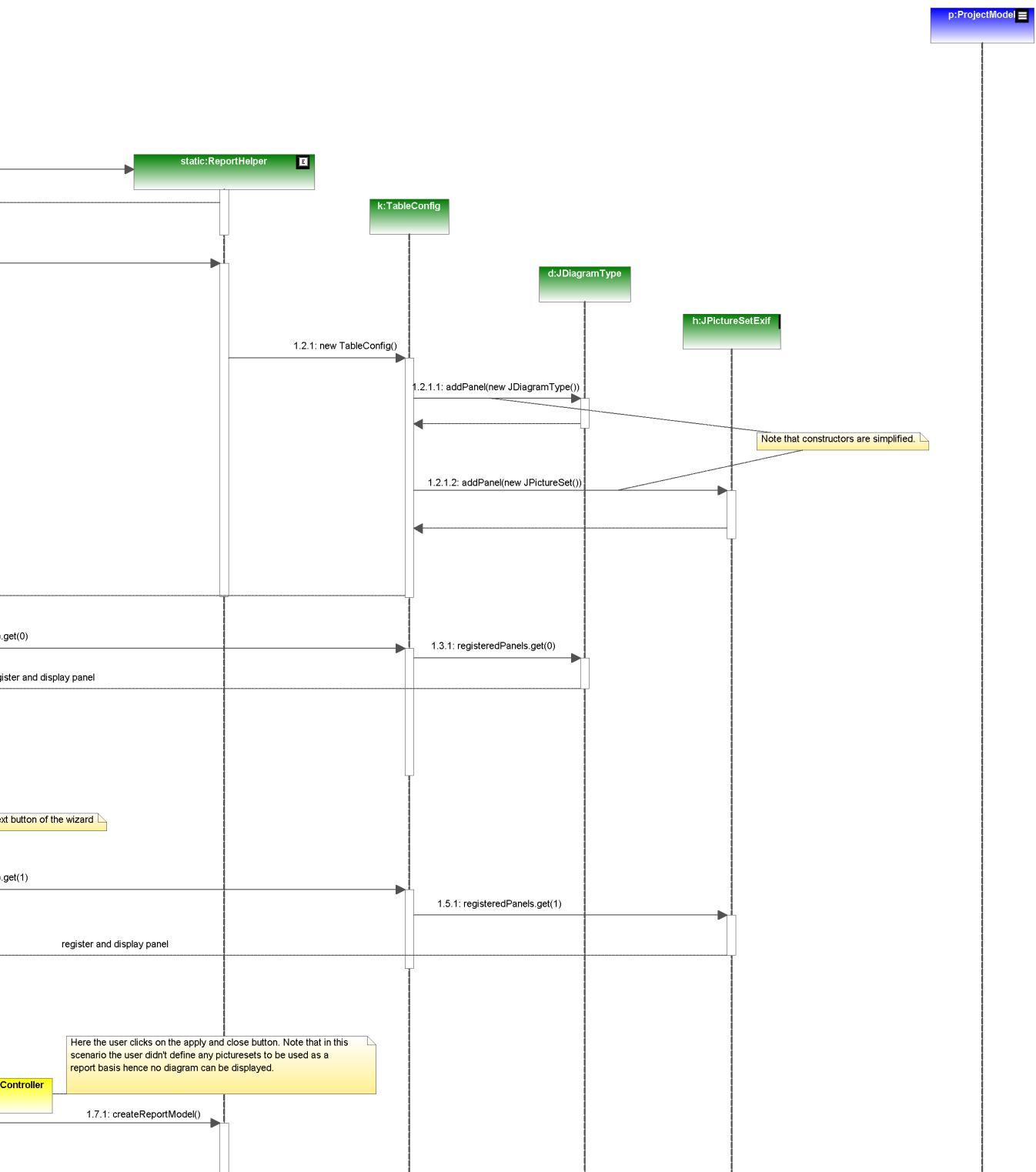


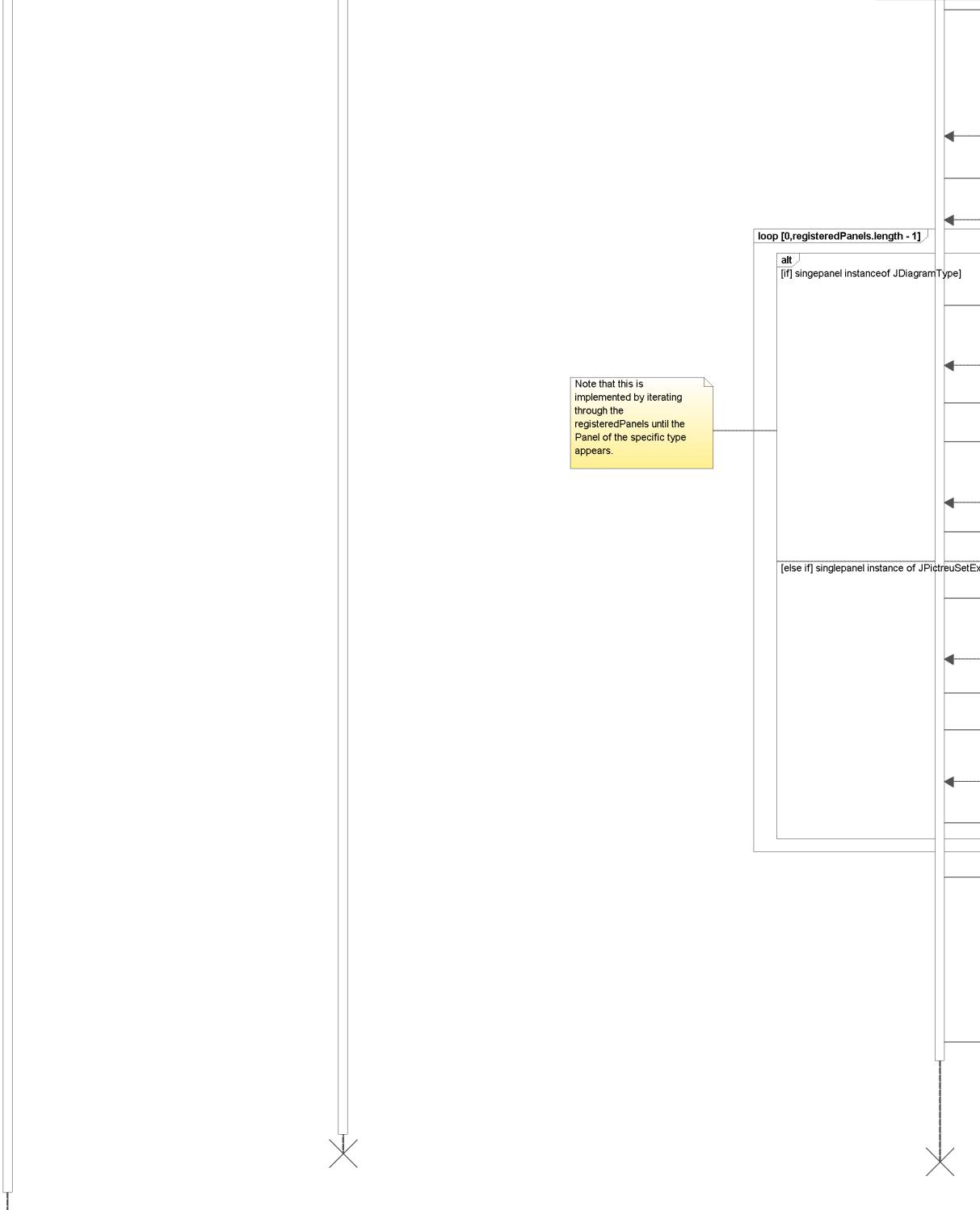


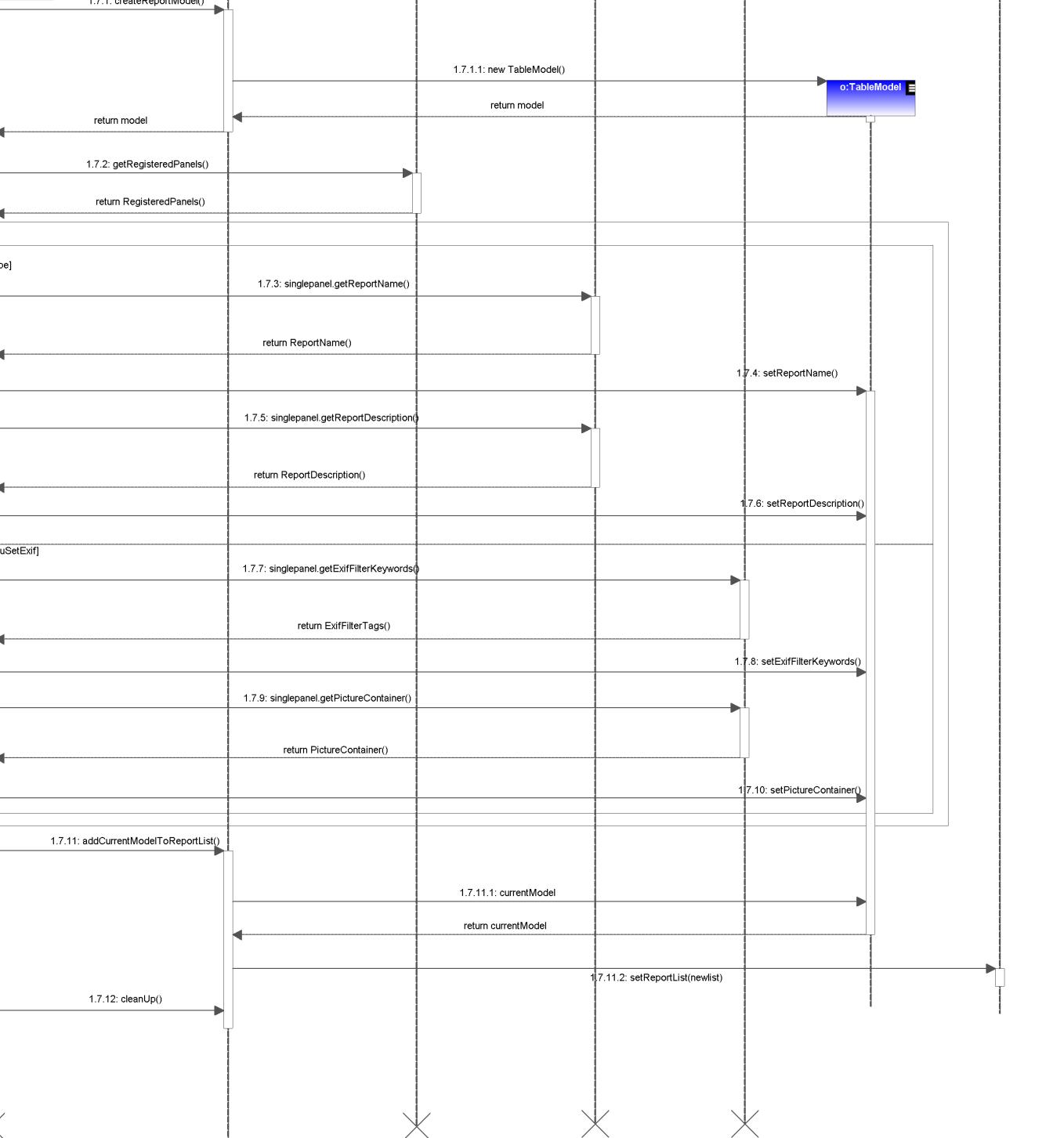


## Sequence diagram for the action "report creation"

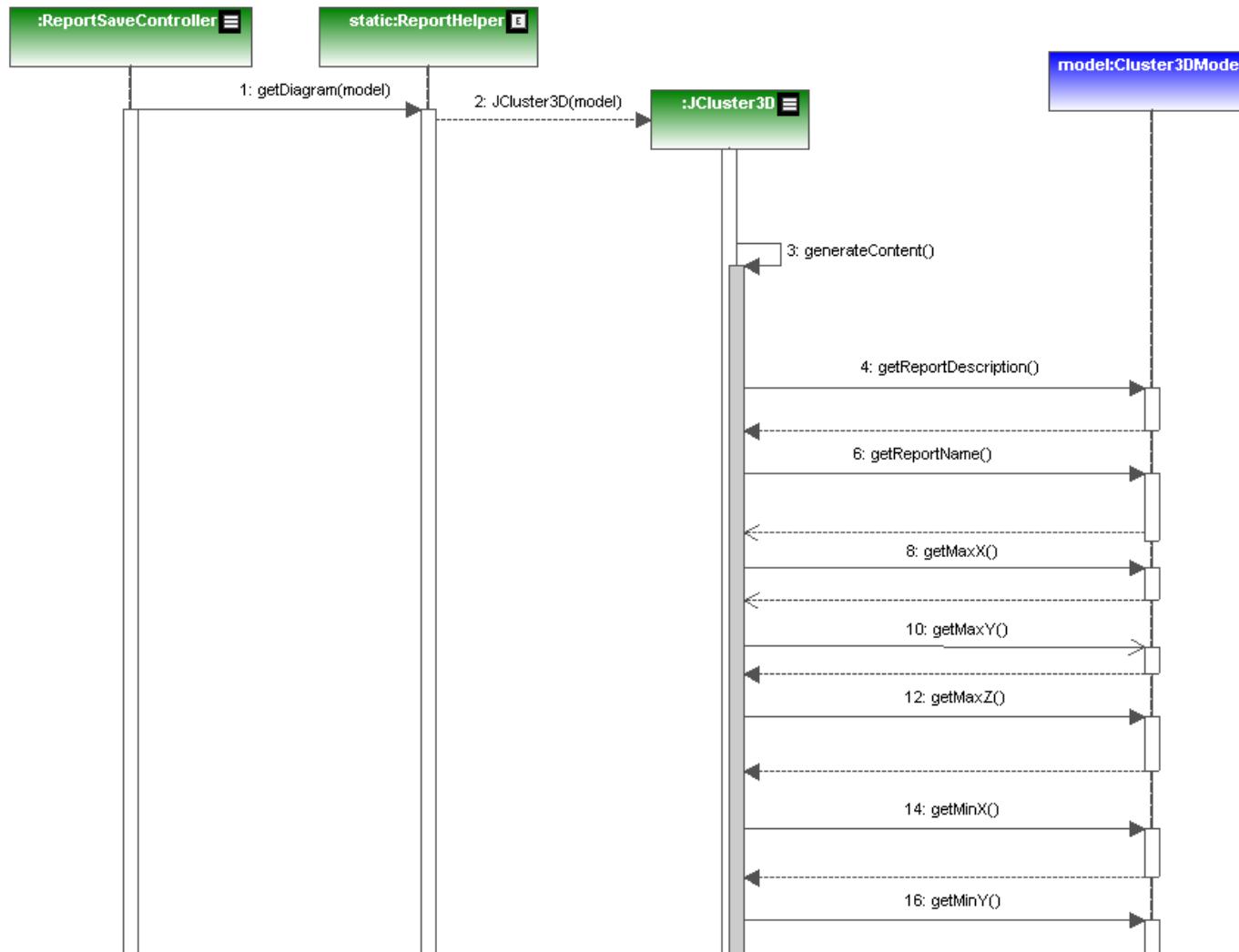


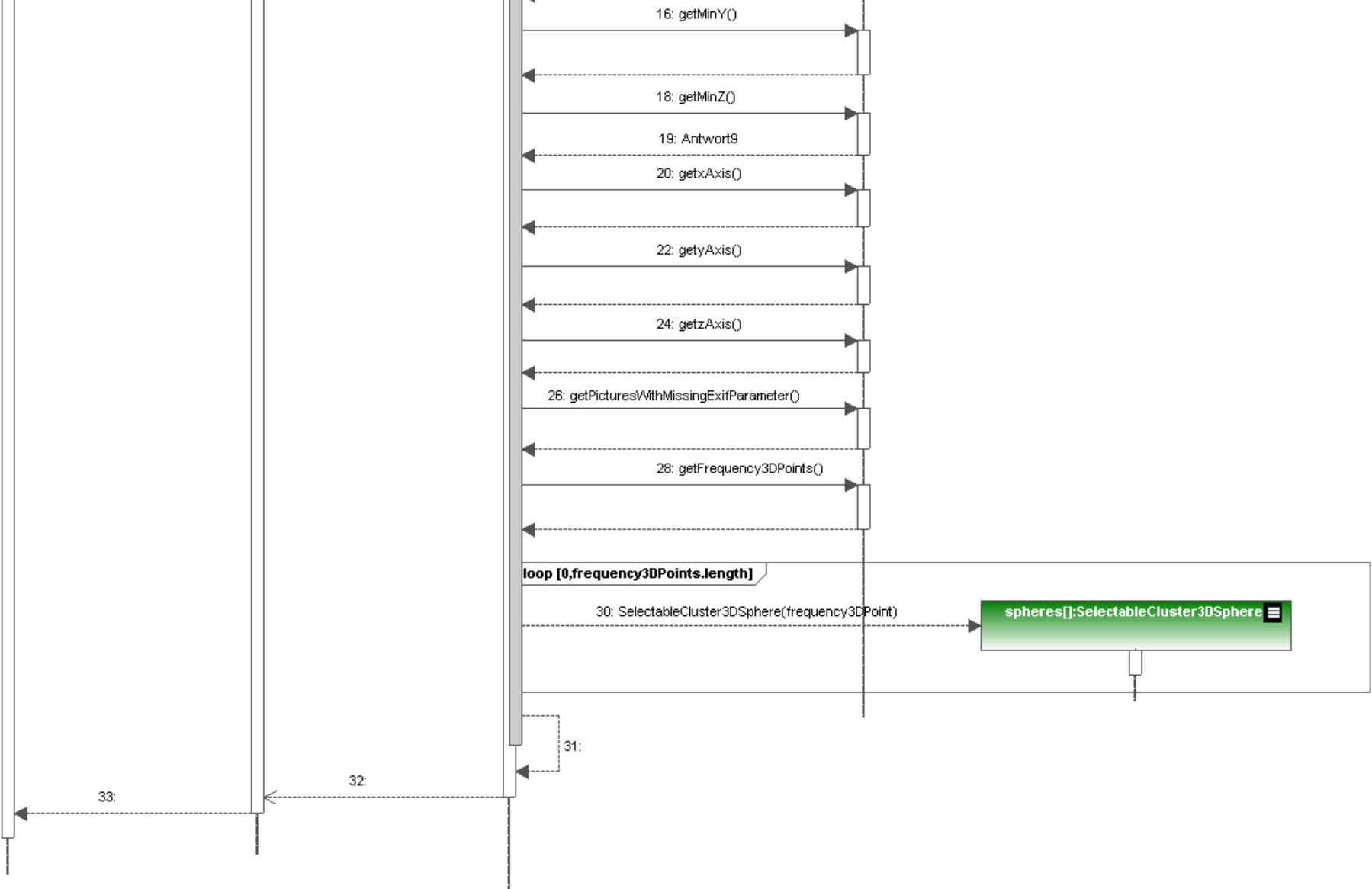






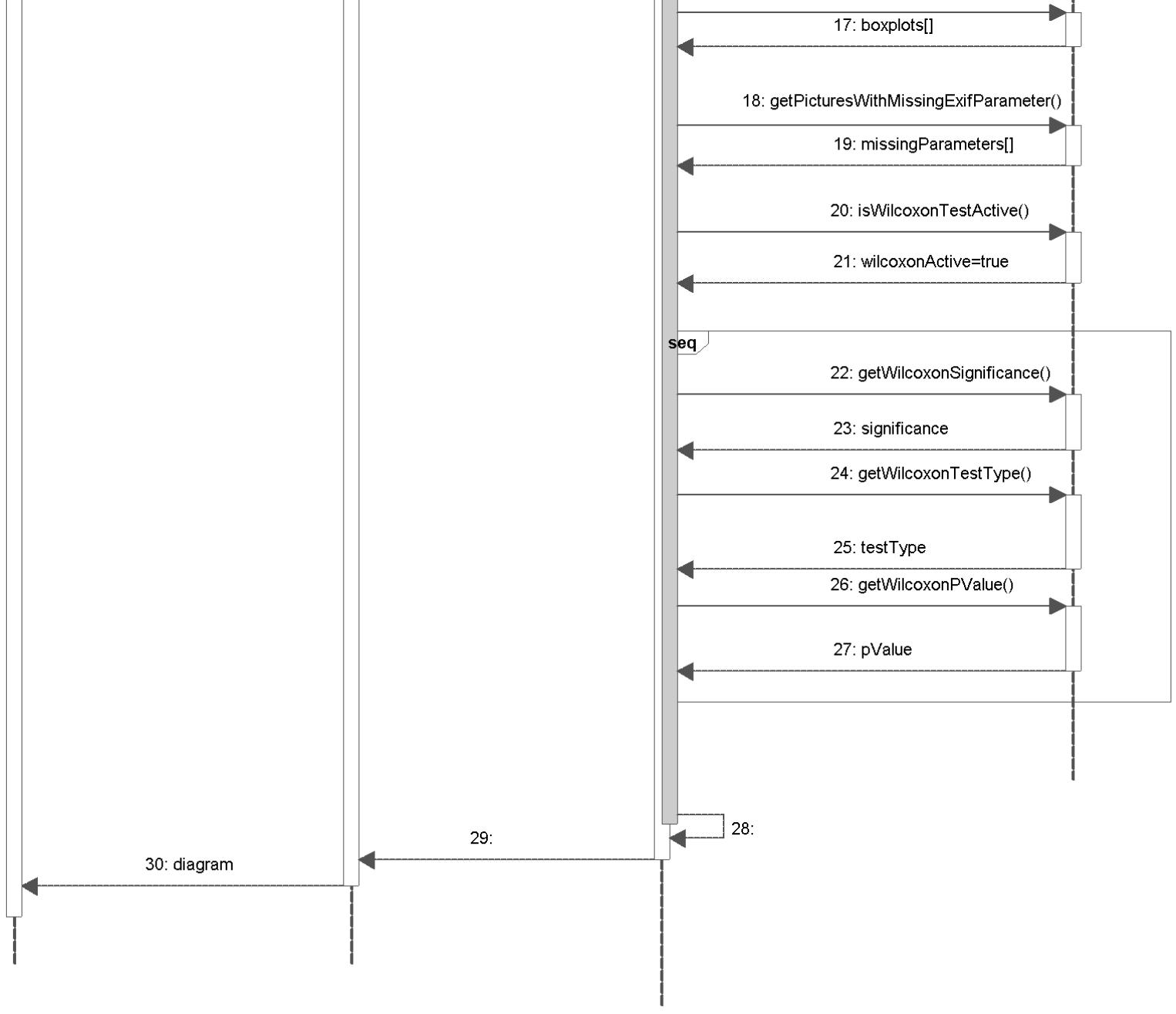
# Sequence diagram cluster 3D

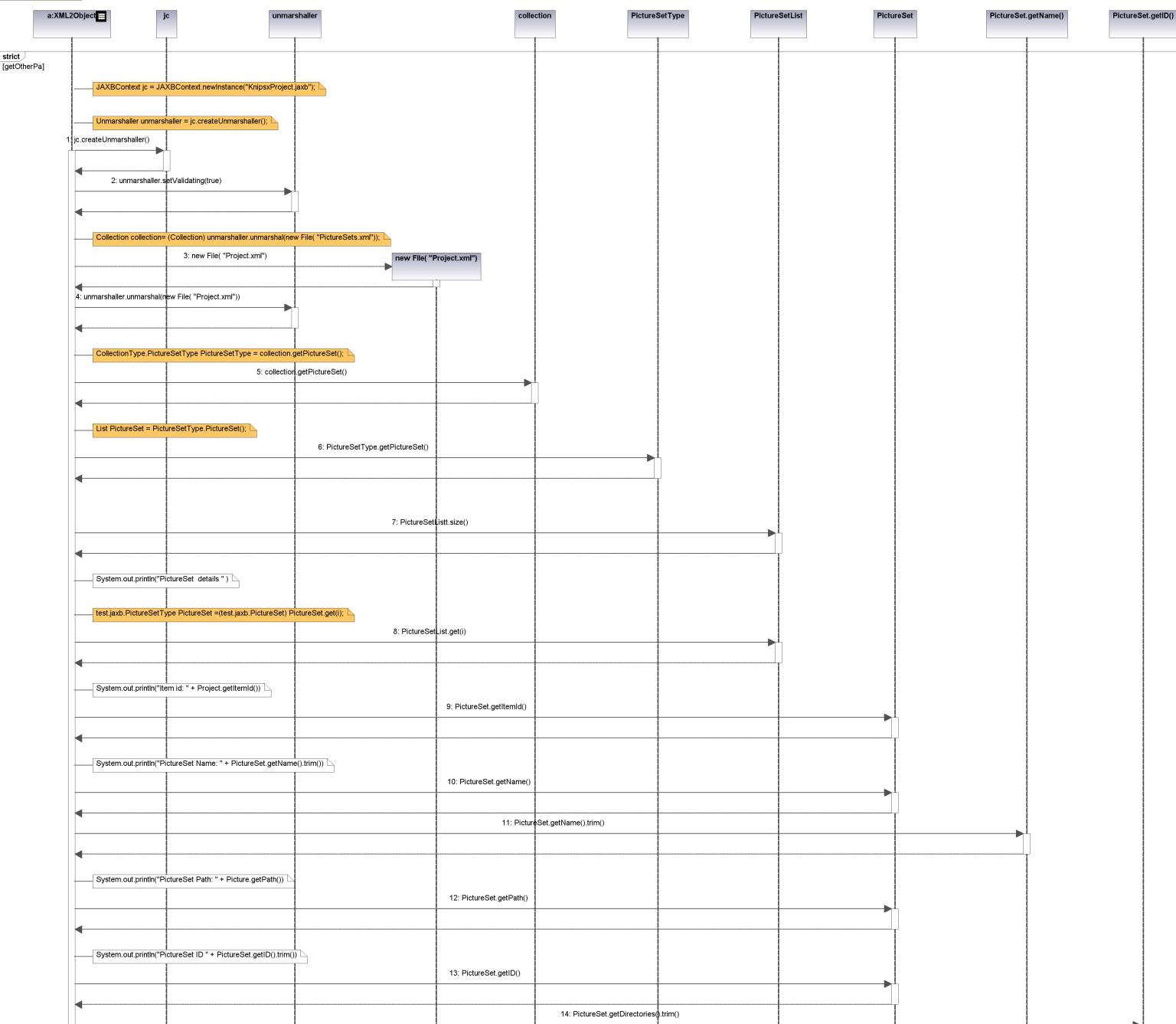


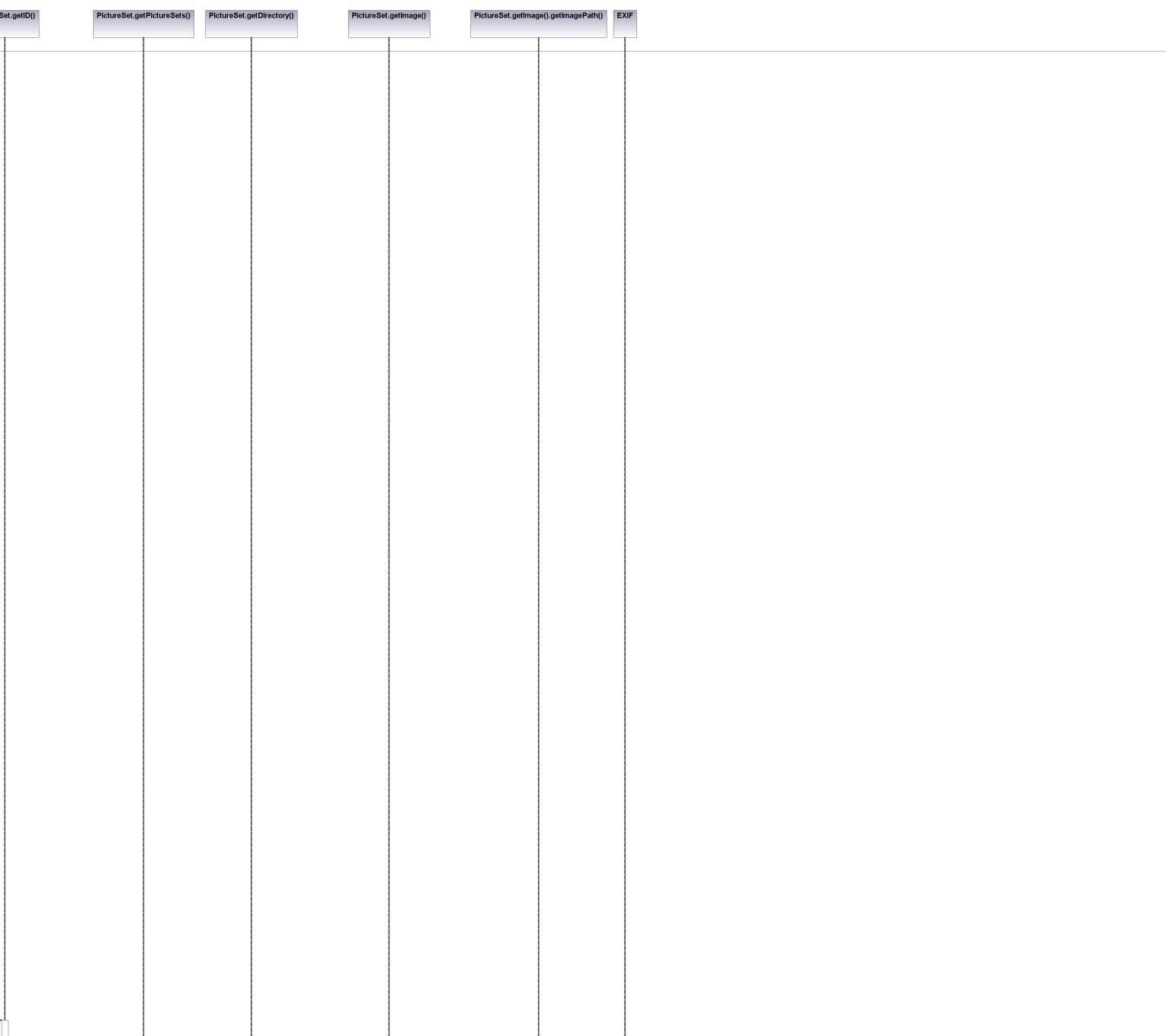


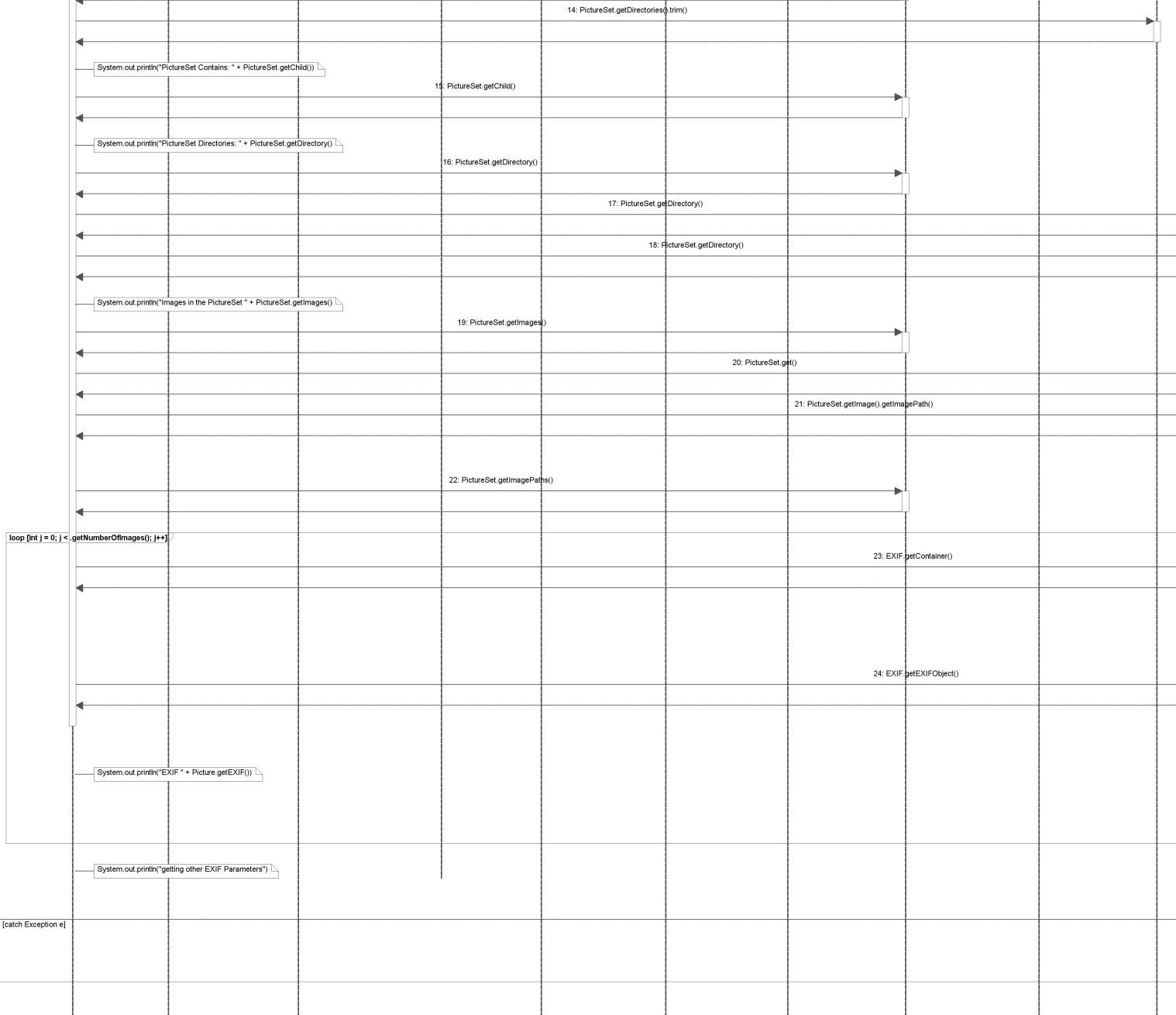
# Sequence diagram boxplot with wilcoxon test

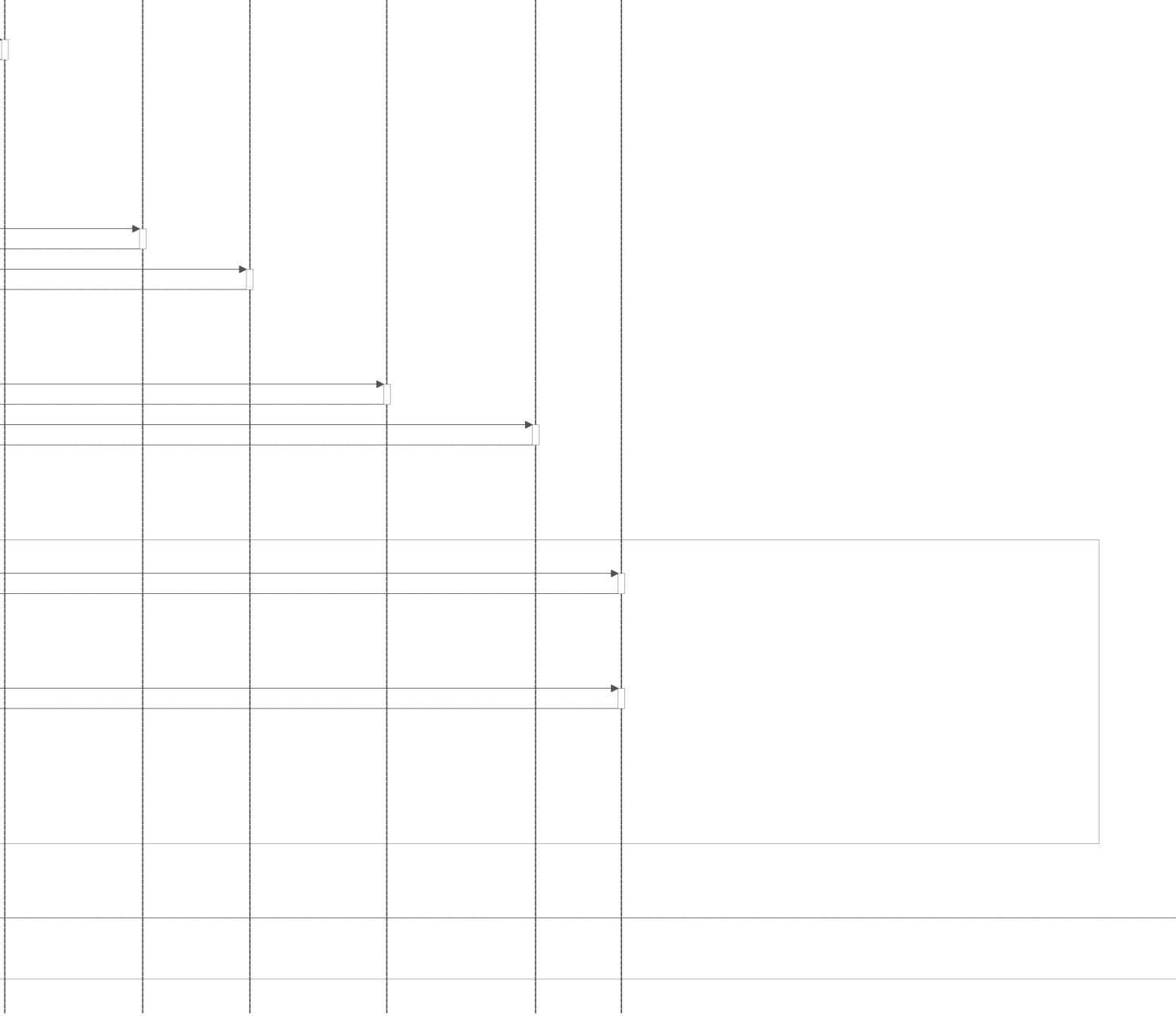












# knipsX.ump

project location <C:\users\david\Desktop\knipsX.ump>

## Index of elements:

Component	<a href="#"><u>Implementierung</u></a>
Enumeration	<a href="#"><u>ExifParameter</u></a> <a href="#"><u>Perspectives</u></a> <a href="#"><u>ReportHelper</u></a> <a href="#"><u>WilcoxonTestType</u></a>
Interface	<a href="#"><u>Iterator&lt;E&gt;Picture</u></a> <a href="#"><u>List&lt;E-&gt;?</u></a> <a href="#"><u>List&lt;E-&gt;AbstractReportModel</u></a> <a href="#"><u>List&lt;E-&gt;Picture</u></a> <a href="#"><u>List&lt;E-&gt;Picture</u></a> <a href="#"><u>List&lt;E-&gt;PictureContainer</u></a> <a href="#"><u>List&lt;E-&gt;PictureContainer</u></a> <a href="#"><u>List&lt;E-&gt;PictureSet</u></a> <a href="#"><u>List&lt;E-&gt;ProjectEntry</u></a> <a href="#"><u>List&lt;E-&gt;ProjectEntry</u></a> <a href="#"><u>PictureContainer</u></a> <a href="#"><u>PictureContainer</u></a>
Package	<a href="#"><u>Component</u></a> <a href="#"><u>View</u></a> <a href="#"><u>controller</u></a> <a href="#"><u>dataformat</u></a> <a href="#"><u>helper</u></a> <a href="#"><u>dataformat</u></a> <a href="#"><u>helper</u></a> <a href="#"><u>diagrams</u></a> <a href="#"><u>diagrams</u></a> <a href="#"><u>exifAdapter</u></a> <a href="#"><u>exifAdapter</u></a> <a href="#"><u>FileChooser</u></a> <a href="#"><u>fileChooser</u></a> <a href="#"><u>fileChoser</u></a> <a href="#"><u>files</u></a> <a href="#"><u>files</u></a> <a href="#"><u>jexifviewer</u></a> <a href="#"><u>jexifviewer</u></a> <a href="#"><u>knipsX</u></a> <a href="#"><u>Kompositum</u></a> <a href="#"><u>model</u></a> <a href="#"><u>org</u></a> <a href="#"><u>Package1</u></a> <a href="#"><u>picturemanagement</u></a> <a href="#"><u>plotter3D</u></a> <a href="#"><u>projectmanagement</u></a> <a href="#"><u>projectmanagement</u></a> <a href="#"><u>projectmanagement</u></a> <a href="#"><u>projectmanagement</u></a> <a href="#"><u>projectview</u></a> <a href="#"><u>projectview</u></a> <a href="#"><u>reportmanagement</u></a> <a href="#"><u>reportmanagement</u></a> <a href="#"><u>reportmanagement</u></a> <a href="#"><u>Root</u></a> <a href="#"><u>shared</u></a> <a href="#"><u>studien</u></a> <a href="#"><u>thumbnail</u></a> <a href="#"><u>thumbnail</u></a> <a href="#"><u>Adapeter</u></a> <a href="#"><u>tooltip</u></a> <a href="#"><u>utils</u></a> <a href="#"><u>viewXML</u></a>

## Class AbstractController

diagram	<pre> classDiagram     class AbstractController {         M         V         -model:M         -view:V         &lt;&lt;constructor&gt;&gt; AbstractController(in view:V)         &lt;&lt;constructor&gt;&gt; AbstractController(in model:M, in view:V)         actionPerformed(in e:ActionEvent):void     } </pre>
hierarchy	<pre> classDiagram     class EventListener     class ActionListener     class AbstractController     ActionListener &lt; -- AbstractController     EventListener &lt; -- ActionListener </pre>
owner	<a href="#">controller</a>
template parameters	name kind constrainingClassifier default <b>M Class V Class</b>
ownedMember	<a href="#">AbstractController</a> <a href="#">AbstractController</a> <a href="#">actionPerformed</a> <a href="#">model</a> <a href="#">view</a>
implemented interfaces	<b>ActionListener</b>
source of relation	InterfaceRealization <b>ActionListener</b>
target of relation	ComponentRealization <b>Implementierung</b>

## Operation AbstractController::AbstractController

owner	<a href="#">AbstractController</a>
parameter	name direction type multiplicity default <b>view in V</b>
ownedMember	<b>view</b>

## Operation AbstractController::AbstractController

owner	<a href="#">AbstractController</a>
parameter	name direction type multiplicity default <b>model in M view in V</b>
ownedMember	<b>model</b> <b>view</b>

## Operation AbstractController::actionPerformed

owner	<a href="#">AbstractController</a>
parameter	name direction type multiplicity default <b>e in ActionEvent return return void</b>
ownedMember	<b>e</b> <b>return</b>

**Property `AbstractController::model`**

owner	<a href="#">AbstractController</a>
-------	------------------------------------

**Property `AbstractController::view`**

owner	<a href="#">AbstractController</a>
-------	------------------------------------

## Class AbstractDoubleAxesModel

diagram	<p style="text-align: center;"><b>AbstractDoubleAxesModel</b></p> <table border="1"> <tr> <td> zAxis: Axis</td></tr> <tr> <td> minZ: Object</td></tr> <tr> <td> maxZ: Object</td></tr> <tr> <td> &lt;&lt;constructor&gt;&gt; AbstractDoubleAxesModel(in pictureContainer: ArrayList&lt;E-&gt;PictureContainer&gt;, in xAxis: Axis, in zAxis: Axis)</td></tr> <tr> <td> &lt;&lt;constructor&gt;&gt; AbstractDoubleAxesModel()</td></tr> <tr> <td> getzAxis(): Axis</td></tr> <tr> <td> setzAxis(in zAxis: Axis): void</td></tr> <tr> <td> getMinZ(): Object</td></tr> <tr> <td> getMaxZ(): Object</td></tr> </table>	zAxis: Axis	minZ: Object	maxZ: Object	<<constructor>> AbstractDoubleAxesModel(in pictureContainer: ArrayList<E->PictureContainer>, in xAxis: Axis, in zAxis: Axis)	<<constructor>> AbstractDoubleAxesModel()	getzAxis(): Axis	setzAxis(in zAxis: Axis): void	getMinZ(): Object	getMaxZ(): Object
zAxis: Axis										
minZ: Object										
maxZ: Object										
<<constructor>> AbstractDoubleAxesModel(in pictureContainer: ArrayList<E->PictureContainer>, in xAxis: Axis, in zAxis: Axis)										
<<constructor>> AbstractDoubleAxesModel()										
getzAxis(): Axis										
setzAxis(in zAxis: Axis): void										
getMinZ(): Object										
getMaxZ(): Object										
hierarchy	<pre> classDiagram     Observable &lt; -- AbstractModel     AbstractModel &lt; -- AbstractReportModel     AbstractReportModel &lt; -- AbstractSingleAxisModel     AbstractSingleAxisModel &lt; -- AbstractDoubleAxesModel     AbstractDoubleAxesModel --&gt;  AbstractTripleAxesModel     AbstractDoubleAxesModel --&gt;  Histogram3DModel     ...   </pre>									
owner	<b>reportmanagement</b>									
ownedMember	<a href="#">AbstractDoubleAxesModel</a> <a href="#">AbstractDoubleAxesModel</a> <a href="#">getMaxZ</a> <a href="#">getMinZ</a> <a href="#">getzAxis</a> <a href="#">maxZ</a> <a href="#">minZ</a> <a href="#">setzAxis</a> <a href="#">zAxis</a>									
general	<a href="#">AbstractSingleAxisModel</a>									
specific	<a href="#">AbstractTripleAxesModel</a> <a href="#">Histogram3DModel</a>									
target of relation	ComponentRealization <a href="#">Implementierung</a>									
documentation	<p>AbstractDoubleAxesModel is the superclass of all diagram models which need at least two axes with exif parameters and exif description, mostly 3D diagrams. Axis is singular and axes is plural :).</p> <p>@author Kevin Zuber</p>									

## Operation AbstractDoubleAxesModel::AbstractDoubleAxesModel

owner	<a href="#">AbstractDoubleAxesModel</a>
parameter	name direction type multiplicity default pictureContainer in <a href="#">ArrayList&lt;E-&gt;PictureContainer&gt;</a> xAxis in <a href="#">Axis</a> zAxis in <a href="#">Axis</a>
ownedMember	<b>pictureContainer</b> <b>xAxis</b> <b>zAxis</b>
documentation	Constructur with all parameters. @param pictureContainer @param xAxis

	@param zAxis
--	--------------

#### Operation **AbstractDoubleAxesModel::AbstractDoubleAxesModel**

owner	<a href="#">AbstractDoubleAxesModel</a>
-------	---

#### Operation **AbstractDoubleAxesModel::getMaxZ**

owner	<a href="#">AbstractDoubleAxesModel</a>
parameter	name direction type multiplicity default <b>return return Object</b>
ownedMember	<b>return</b>
documentation	Give the biggest value in the z-axis. @return the biggest value in the z-axis.

#### Operation **AbstractDoubleAxesModel::getMinZ**

owner	<a href="#">AbstractDoubleAxesModel</a>
parameter	name direction type multiplicity default <b>return return Object</b>
ownedMember	<b>return</b>
documentation	Give the smallest value in the z-axis. @return the smallest value in the z-axis.

#### Operation **AbstractDoubleAxesModel::getzAxis**

owner	<a href="#">AbstractDoubleAxesModel</a>
parameter	name direction type multiplicity default <b>return return Axis</b>
ownedMember	<b>return</b>

#### Property **AbstractDoubleAxesModel::maxZ**

owner	<a href="#">AbstractDoubleAxesModel</a>
-------	---

#### Property **AbstractDoubleAxesModel::minZ**

owner	<a href="#">AbstractDoubleAxesModel</a>
-------	---

#### Operation **AbstractDoubleAxesModel::setzAxis**

owner	<a href="#">AbstractDoubleAxesModel</a>
parameter	name direction type multiplicity default <b>zAxis in Axis return return void</b>
ownedMember	<b>return zAxis</b>
documentation	Sets the zAxis . @param a zAxis

--	--

Property **AbstractDoubleAxesModel::zAxis**

owner	<a href="#">AbstractDoubleAxesModel</a>
-------	---

## Class AbstractModel

diagram	<pre> classDiagram     class AbstractModel {         &lt;&lt;updateViews():void&gt;&gt;     }   </pre>
hierarchy	<pre> classDiagram     class Observable     class AbstractModel {         &lt;&lt;updateViews():void&gt;&gt;     }     Observable &lt; -- AbstractModel     class ProjectManagementModel     class ProjectEntry     class AbstractReportModel     ProjectManagementModel ..&gt; -- AbstractModel     ProjectEntry ..&gt; -- AbstractModel     AbstractReportModel ..&gt; -- AbstractModel   </pre>
owner	<a href="#">model</a>
ownedMember	<a href="#">updateViews</a>
general	<a href="#">Observable</a>
specific	<a href="#">AbstractReportModel</a> <a href="#">ProjectEntry</a> <a href="#">ProjectManagementModel</a>
target of relation	ComponentRealization <a href="#">Implementierung</a>
typedElements	Class <a href="#">DemoJFileChooser</a> Operation <a href="#">DemoJFileChooser</a>

## Operation AbstractModel::updateViews

owner	<a href="#">AbstractModel</a>
parameter	name direction type multiplicity default <b>return return void</b>
ownedMember	<b>return</b>
documentation	Aktualisiere alle View, die das Modell beobachten.

## Class AbstractReportCompilation

diagram	<p>The diagram shows the class <b>AbstractReportCompilation</b> with the following details:</p> <ul style="list-style-type: none"> <li><b>Attributes:</b> <ul style="list-style-type: none"> <li><code>model:M</code></li> <li><code>&lt;&lt;final&gt;&gt; serialVersionUID:long=1L</code></li> <li><code>registeredPanels:ArrayList&lt;E-&gt;JAbstractSinglePanel&gt;=new ArrayList&lt;JAbstractSinglePanel&gt;()</code></li> </ul> </li> <li><b>Operations:</b> <ul style="list-style-type: none"> <li><code>&lt;&lt;constructor&gt;&gt; AbstractReportCompilation(in model:M)</code></li> <li><code>addPanel(in component:JAbstractSinglePanel):void</code></li> <li><code>getRegisteredPanels():ArrayList&lt;E-&gt;JAbstractSinglePanel&gt;</code></li> </ul> </li> </ul>
owner	<b>reportmanagement</b>
template parameters	name kind constrainingClassifier default <b>M Class AbstractReportModel</b>
ownedMember	<b>AbstractReportCompilation addPanel getRegisteredPanels model registeredPanels serialVersionUID</b>
target of relation	ComponentRealization <b>Implementierung</b>
documentation	<p>This class represents a sequence of single panels which in turn generate a report configuration window.</p> <p>@author David Kaufman</p> <p>@param &lt;M&gt;</p>

## Operation AbstractReportCompilation::AbstractReportCompilation

owner	<b>AbstractReportCompilation</b>
parameter	name direction type multiplicity default <b>model in M</b>
ownedMember	<b>model</b>

## Operation AbstractReportCompilation::addPanel

owner	<b>AbstractReportCompilation</b>
parameter	name direction type multiplicity default <b>component in JAbstractSinglePanel return return void</b>
ownedMember	<b>component return</b>
documentation	<p>Adds the specified panel to the current report configuration.</p> <p>@param component</p>

## Operation AbstractReportCompilation::getRegisteredPanels

owner	<b>AbstractReportCompilation</b>
parameter	name direction type multiplicity default <b>return return ArrayList&lt;E-&gt;JAbstractSinglePanel&gt;</b>
ownedMember	<b>return</b>
documentation	Returns the registered panels inside the current report configuration.

	@return registered panels
--	---------------------------

Property **AbstractReportCompilation::model**

owner	<a href="#">AbstractReportCompilation</a>
-------	---

Property **AbstractReportCompilation::registeredPanels**

owner	<a href="#">AbstractReportCompilation</a>
-------	---

Property **AbstractReportCompilation::serialVersionUID**

owner	<a href="#">AbstractReportCompilation</a>
documentation	

## Class AbstractReportModel

diagram	<p style="text-align: center;"><b><i>AbstractReportModel</i></b></p> <pre> ■ pictureContainer:ArrayList&lt;E&gt;&gt;PictureContainer&gt; ■ reportName:String ■ reportDescription:String ■ exifFilterKeywords:String[*] ■ reportID:int ■ missingExifParameters:PictureParameter[*]  ■ getReportID():int ■ &lt;&lt;constructor&gt;&gt; AbstractReportModel() ■ &lt;&lt;constructor&gt;&gt; AbstractReportModel(in pictureContainer:ArrayList&lt;E&gt;&gt;PictureContainer&gt;) ■ &lt;&lt;constructor&gt;&gt; AbstractReportModel(in pictureContainer:ArrayList&lt;E&gt;&gt;PictureContainer&gt;, in reportName:String, in reportDescription:String) ■ &lt;&lt;constructor&gt;&gt; AbstractReportModel(in pictureContainer:ArrayList&lt;E&gt;&gt;PictureContainer&gt;, in reportName:String, in reportDescription:String, in exifFilterKeywords:String[*]) ■ getReportName():String ■ setReportName(in reportName:String):void ■ getReportDescription():String ■ setReportDescription(in reportDescription:String):void ■ getPictureContainer():ArrayList&lt;E&gt;&gt;PictureContainer&gt; ■ setPictureContainer(in pictureContainer:ArrayList&lt;E&gt;&gt;PictureContainer&gt;):void ■ setExifFilterKeywords(in exifFilterKeywords:String[*]):void ■ getExifFilterKeywords():String[*] ■ getPicturesWithMissingExifParameter():PictureParameter[*] </pre>
hierarchy	<pre> classDiagram     Observable &lt; -- AbstractModel     AbstractModel &lt; -- AbstractReportModel     AbstractReportModel &lt; -- AbstractSingleAxisModel     AbstractReportModel &lt; -- TableModel     AbstractReportModel &lt; -- TextModel </pre>
owner	<b>reportmanagement</b>
ownedMember	<a href="#">AbstractReportModel</a> <a href="#">AbstractReportModel</a> <a href="#">AbstractReportModel</a> <a href="#">AbstractReportModel</a> <a href="#">AbstractReportModel</a> <a href="#">exifFilterKeywords</a> <a href="#">getExifFilterKeywords</a> <a href="#">getPictureContainer</a> <a href="#">getPicturesWithMissingExifParameter</a> <a href="#">getReportDescription</a> <a href="#">getReportID</a> <a href="#">getReportName</a> <a href="#">missingExifParameters</a> <a href="#">pictureContainer</a> <a href="#">reportDescription</a> <a href="#">reportID</a> <a href="#">reportName</a> <a href="#">setExifFilterKeywords</a> <a href="#">setPictureContainer</a> <a href="#">setReportDescription</a> <a href="#">setReportName</a>
general	<a href="#">AbstractModel</a>
specific	<a href="#">AbstractSingleAxisModel</a> <a href="#">TableModel</a> <a href="#">TextModel</a>
target of relation	ComponentRealization <a href="#">Implementierung</a>
typedElements	Class <a href="#">ProjectModel</a> Operation <a href="#">addReport</a> <a href="#">removeReport</a> Enumeration <a href="#">ReportHelper</a> Property <a href="#">currentModel</a> Operation <a href="#">createReportModel</a> <a href="#">getDiagram</a>
documentation	<p>AbstractReportModel is the superclass of all ReportModels. It saves report name, description and the tags with will be used to filter the picture sets.</p> <p>@author Kevin Zuber</p>

## Operation **AbstractReportModel::AbstractReportModel**

**owner** **AbstractReportModel**

#### Operation **AbstractReportModel::AbstractReportModel**

owner	<b>AbstractReportModel</b>
parameter	name direction type multiplicity default <b>pictureContainer in ArrayList&lt;E&gt;PictureContainer</b>
ownedMember	<b>pictureContainer</b>

#### Operation **AbstractReportModel::AbstractReportModel**

owner	<b>AbstractReportModel</b>
parameter	name direction type multiplicity default <b>pictureContainer in ArrayList&lt;E&gt;PictureContainer</b> <b>reportName in String</b> <b>reportDescription in String</b>
ownedMember	<b>pictureContainer reportDescription reportName</b>

#### Operation **AbstractReportModel::AbstractReportModel**

owner	<b>AbstractReportModel</b>
parameter	name direction type multiplicity default <b>pictureContainer in ArrayList&lt;E&gt;PictureContainer</b> <b>reportName in String</b> <b>reportDescription in String</b> <b>exifFilterKeywords in String *</b>
ownedMember	<b>exifFilterKeywords pictureContainer reportDescription reportName</b>
documentation	Constructor with all parameters @param pictureContainer a picture container @param reportName the name of the report @param reportDescription a description of a report @param exifFilterKeywords keywords which must be in every picture for the report

#### Property **AbstractReportModel::exifFilterKeywords**

owner	<b>AbstractReportModel</b>
-------	----------------------------

#### Operation **AbstractReportModel::getExifFilterKeywords**

owner	<b>AbstractReportModel</b>
parameter	name direction type multiplicity default <b>return return String *</b>
ownedMember	<b>return</b>

#### Operation **AbstractReportModel::getPictureContainer**

owner	<b>AbstractReportModel</b>
parameter	name direction type multiplicity default <b>return return ArrayList&lt;E&gt;PictureContainer</b>
ownedMember	<b>return</b>

#### Operation **AbstractReportModel::getPicturesWithMissingExifParameter**

owner	<b>AbstractReportModel</b>
parameter	name direction type multiplicity default <b>return return PictureParameter *</b>
ownedMember	<b>return</b>

documentati on	Returns an array of PictureParameter objects with each a pair of a picture and the missing exif parameter inside the picture.
-------------------	---

#### Operation **AbstractReportModel::getReportDescription**

owner	<b>AbstractReportModel</b>
parameter	name direction type multiplicity default <b>return return String</b>
ownedMember	<b>return</b>

#### Operation **AbstractReportModel::getReportID**

owner	<b>AbstractReportModel</b>
parameter	name direction type multiplicity default <b>return return int</b>
ownedMember	<b>return</b>

#### Operation **AbstractReportModel::getReportName**

owner	<b>AbstractReportModel</b>
parameter	name direction type multiplicity default <b>return return String</b>
ownedMember	<b>return</b>

#### Property **AbstractReportModel::missingExifParameters**

owner	<b>AbstractReportModel</b>
-------	----------------------------

#### Property **AbstractReportModel::pictureContainer**

owner	<b>AbstractReportModel</b>
-------	----------------------------

#### Property **AbstractReportModel::reportDescription**

owner	<b>AbstractReportModel</b>
-------	----------------------------

#### Property **AbstractReportModel::reportID**

owner	<b>AbstractReportModel</b>
-------	----------------------------

#### Property **AbstractReportModel::reportName**

owner	<b>AbstractReportModel</b>
-------	----------------------------

#### Operation **AbstractReportModel::setExifFilterKeywords**

owner	<b>AbstractReportModel</b>
parameter	name direction type multiplicity default <b>exifFilterKeywords in String * return return void</b>
ownedMember	<b>exifFilterKeywords return</b>

	ber	
--	-----	--

#### Operation **AbstractReportModel::setPictureContainer**

owner	<b>AbstractReportModel</b>
parameter	name direction type multiplicity default <b>pictureContainer in ArrayList&lt;E&gt;&gt;PictureContainer</b> return return void
ownedMember	<b>pictureContainer return</b>

#### Operation **AbstractReportModel::setReportDescription**

owner	<b>AbstractReportModel</b>
parameter	name direction type multiplicity default <b>reportDescription in String</b> return return void
ownedMember	<b>reportDescription return</b>

#### Operation **AbstractReportModel::setReportName**

owner	<b>AbstractReportModel</b>
parameter	name direction type multiplicity default <b>reportName in String</b> return return void
ownedMember	<b>reportName return</b>

## Class AbstractSingleAxisModel

diagram	<pre> <b>AbstractSingleAxisModel</b>  xAxis: Axis minX: Object maxX: Object minY: Object maxY: Object  &lt;&lt;constructor&gt;&gt; AbstractSingleAxisModel(in pictureContainer: ArrayList&lt;E&gt;PictureContainer&gt;, in xAxis: Axis) &lt;&lt;constructor&gt;&gt; AbstractSingleAxisModel() getAxis(): Axis setAxis(in xAxis: Axis): void getMinX(): Object getMaxX(): Object getMinY(): Object getMaxY(): Object </pre>
hierarchy	<pre> classDiagram     Observable &lt; -- AbstractModel     AbstractModel &lt; -- AbstractReportModel     AbstractReportModel &lt; -- AbstractSingleAxisModel     AbstractSingleAxisModel &lt; -- AbstractDoubleAxesModel     AbstractSingleAxisModel &lt; -- BoxplotModel     AbstractSingleAxisModel &lt; -- Histogram2DModel     ...   </pre>
owner	<a href="#">reportmanagement</a>
ownedMember	<a href="#">AbstractSingleAxisModel</a> <a href="#">AbstractSingleAxisModel getMaxX</a> <a href="#">AbstractSingleAxisModel getMaxY</a> <a href="#">AbstractSingleAxisModel getMinX</a> <a href="#">AbstractSingleAxisModel getMinY</a> <a href="#">AbstractSingleAxisModel getxAxis</a> <a href="#">AbstractSingleAxisModel maxX</a> <a href="#">AbstractSingleAxisModel maxY</a> <a href="#">AbstractSingleAxisModel minX</a> <a href="#">AbstractSingleAxisModel minY</a> <a href="#">AbstractSingleAxisModel setxAxis</a> <a href="#">AbstractSingleAxisModel xAxis</a>
general	<a href="#">AbstractReportModel</a>
specific	<a href="#">AbstractDoubleAxesModel</a> <a href="#">BoxplotModel</a> <a href="#">Histogram2DModel</a>
target of relation	ComponentRealization <a href="#">Implementierung</a>
documentation	<p>AbstractDoubleSingleAxisModel is the superclass of all diagram models which need at least one axis with exif parameters and exif description, mostly 2D diagrams. Axis is singular and axes is plural :).</p> <p>@author Kevin Zuber</p>

## Operation AbstractSingleAxisModel::AbstractSingleAxisModel

owner	<a href="#">AbstractSingleAxisModel</a>
parameter	name direction type multiplicity default <a href="#">pictureContainer</a> in <a href="#">ArrayList&lt;E&gt;PictureContainer&gt;</a> <a href="#">xAxis</a> in <a href="#">Axis</a>
ownedMember	<a href="#">pictureContainer</a> <a href="#">xAxis</a>

#### Operation **AbstractSingleAxisModel::AbstractSingleAxisModel**

owner	<a href="#">AbstractSingleAxisModel</a>
-------	---

#### Operation **AbstractSingleAxisModel::getMaxX**

owner	<a href="#">AbstractSingleAxisModel</a>
parameter	name direction type multiplicity default <b>return return Object</b>
ownedMember	<b>return</b>
documentation	@return the biggest value in the x-axis.

#### Operation **AbstractSingleAxisModel::getMaxY**

owner	<a href="#">AbstractSingleAxisModel</a>
parameter	name direction type multiplicity default <b>return return Object</b>
ownedMember	<b>return</b>
documentation	@return the biggest value in the y-axis.

#### Operation **AbstractSingleAxisModel::getMinX**

owner	<a href="#">AbstractSingleAxisModel</a>
parameter	name direction type multiplicity default <b>return return Object</b>
ownedMember	<b>return</b>
documentation	@return the smalles value in the x-axis.

#### Operation **AbstractSingleAxisModel::getMinY**

owner	<a href="#">AbstractSingleAxisModel</a>
parameter	name direction type multiplicity default <b>return return Object</b>
ownedMember	<b>return</b>
documentation	@return the smalles value in the y-axis.

#### Operation **AbstractSingleAxisModel::getxAxis**

owner	<a href="#">AbstractSingleAxisModel</a>
parameter	name direction type multiplicity default <b>return return Axis</b>
ownedMember	<b>return</b>

ber	
-----	--

Property **AbstractSingleAxisModel::maxX**

owner	<a href="#">AbstractSingleAxisModel</a>
-------	---

Property **AbstractSingleAxisModel::maxY**

owner	<a href="#">AbstractSingleAxisModel</a>
-------	---

Property **AbstractSingleAxisModel::minX**

owner	<a href="#">AbstractSingleAxisModel</a>
-------	---

Property **AbstractSingleAxisModel::minY**

owner	<a href="#">AbstractSingleAxisModel</a>
-------	---

Operation **AbstractSingleAxisModel::setxAxis**

owner	<a href="#">AbstractSingleAxisModel</a>
parameter	name direction type multiplicity default <b>xAxis</b> in <a href="#">Axis</a> return return void
ownedMember	<b>return xAxis</b>

Property **AbstractSingleAxisModel::xAxis**

owner	<a href="#">AbstractSingleAxisModel</a>
-------	---

## Class AbstractTripleAxesModel

diagram	<pre> <b>AbstractTripleAxesModel</b>  yAxis: Axis &lt;&lt;constructor&gt;&gt; AbstractTripleAxesModel(<b>in</b> pictureContainer: <b>ArrayList&lt;E-&gt;PictureContainer&gt;</b>, <b>in</b> xAxi<b>s</b>: Axis, <b>in</b> zAxi<b>s</b>: Axis, <b>in</b> yAxi<b>s</b>: Axis) &lt;&lt;constructor&gt;&gt; AbstractTripleAxesModel() getyAxis(): Axis setyAxis(<b>in</b> yAxi<b>s</b>: Axis): void </pre>
hierarchy	<pre> classDiagram     Observable &lt; -- AbstractModel     AbstractModel &lt; -- AbstractReportModel     AbstractReportModel &lt; -- AbstractSingleAxisModel     AbstractSingleAxisModel &lt; -- AbstractDoubleAxesModel     AbstractDoubleAxesModel &lt; -- AbstractTripleAxesModel     AbstractTripleAxesModel &lt; -- Cluster3DModel </pre>
owner	<a href="#">reportmanagement</a>
ownedMember	<a href="#">AbstractTripleAxesModel</a> <a href="#">AbstractTripleAxesModel</a> <a href="#">getyAxis</a> <a href="#">setyAxis</a> <a href="#">yAxis</a>
general	<a href="#">AbstractDoubleAxesModel</a>
specific	<a href="#">Cluster3DModel</a>
target of relation	ComponentRealization <a href="#">Implementierung</a>
documentation	<p>AbstractTripleAxesModel is the superclass of all diagram models which need at least three axes with exif parameters and exif description.</p> <p>@author David Kaufman</p>

## Operation AbstractTripleAxesModel::AbstractTripleAxesModel

owner	<a href="#">AbstractTripleAxesModel</a>
parameter	name direction type multiplicity default <b>pictureContainer</b> in <b>ArrayList&lt;E-&gt;PictureContainer&gt;</b> <b>xAxis</b> in <b>Axis</b> <b>zAxis</b> in <b>Axis</b> <b>yAxis</b> in <b>Axis</b>
ownedMember	<b>pictureContainer</b> <b>xAxis</b> <b>yAxis</b> <b>zAxis</b>
documentation	<p>Constructor with all parameters needed for the model.</p> <p>@param pictureContainer a List of pictureContainer for the report</p> <p>@param xAxis the xAxi<b>s</b></p> <p>@param zAxis the zAxi<b>s</b></p> <p>@param yAxi<b>s</b> the yAxi<b>s</b></p>

**Operation AbstractTripleAxesModel::AbstractTripleAxesModel**

owner	<a href="#">AbstractTripleAxesModel</a>
-------	---

**Operation AbstractTripleAxesModel::getyAxis**

owner	<a href="#">AbstractTripleAxesModel</a>
parameter	name direction type multiplicity default <b>return return Axis</b>
ownedMember	<b>return</b>

**Operation AbstractTripleAxesModel::setyAxis**

owner	<a href="#">AbstractTripleAxesModel</a>
parameter	name direction type multiplicity default <b>yAxis in Axis return return void</b>
ownedMember	<b>return yAxis</b>

**Property AbstractTripleAxesModel::yAxis**

owner	<a href="#">AbstractTripleAxesModel</a>
-------	---

## Class Axis

diagram	<p style="text-align: center;"><b>Axis</b></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 5px;">description:<code>String</code></td></tr> <tr> <td style="padding: 5px;">parameter:<code>ExifParameter</code></td></tr> <tr> <td style="padding: 5px; vertical-align: top;"> <code>&lt;&lt;constructor&gt;&gt; Axis(in description:String, in parameter:ExifParameter)</code>  <code>&lt;&lt;constructor&gt;&gt; Axis(in parameter:ExifParameter)</code>  <code>getDescription():String</code>  <code>setDescription(in description:String):void</code>  <code>getParameter():ExifParameter</code>  <code>setParameter(in parameter:ExifParameter):void</code> </td></tr> </table>	description: <code>String</code>	parameter: <code>ExifParameter</code>	<code>&lt;&lt;constructor&gt;&gt; Axis(in description:String, in parameter:ExifParameter)</code> <code>&lt;&lt;constructor&gt;&gt; Axis(in parameter:ExifParameter)</code> <code>getDescription():String</code> <code>setDescription(in description:String):void</code> <code>getParameter():ExifParameter</code> <code>setParameter(in parameter:ExifParameter):void</code>
description: <code>String</code>				
parameter: <code>ExifParameter</code>				
<code>&lt;&lt;constructor&gt;&gt; Axis(in description:String, in parameter:ExifParameter)</code> <code>&lt;&lt;constructor&gt;&gt; Axis(in parameter:ExifParameter)</code> <code>getDescription():String</code> <code>setDescription(in description:String):void</code> <code>getParameter():ExifParameter</code> <code>setParameter(in parameter:ExifParameter):void</code>				
owner	<a href="#">reportmanagement</a>			
ownedMember	<a href="#">Axis</a> <a href="#">Axis description</a> <a href="#">getDescription</a> <a href="#">getParameter</a> <a href="#">parameter</a> <a href="#">setDescription</a> <a href="#">setParameter</a>			
target of relation	ComponentRealization <a href="#">Implementierung</a>			
typedElements	Class <a href="#">AbstractDoubleAxesModel</a> Property <a href="#">zAxis</a> Operation <a href="#">AbstractDoubleAxesModel getzAxis</a> <a href="#">setzAxis</a> Class <a href="#">AbstractSingleAxisModel</a> Property <a href="#">xAxis</a> Operation <a href="#">AbstractSingleAxisModel getxAxis</a> <a href="#">setxAxis</a> Class <a href="#">AbstractTripleAxesModel</a> Property <a href="#">yAxis</a> Operation <a href="#">AbstractTripleAxesModel getyAxis</a> <a href="#">setyAxis</a> Class <a href="#">BoxplotModel</a> Operation <a href="#">BoxplotModel</a> Class <a href="#">Cluster3DModel</a> Operation <a href="#">Cluster3DModel</a> Class <a href="#">Histogram2DModel</a> Operation <a href="#">Histogram2DModel</a> Class <a href="#">Histogram3DModel</a> Operation <a href="#">Histogram3DModel</a>			
documentation	Represents an axis in a diagram. It saves which exif parameter will be used for the axis and how it will be named in the diagram. @author Kevin Zuber			

## Operation Axis::Axis

owner	<a href="#">Axis</a>
parameter	name direction type multiplicity default <b>description</b> <code>in String</code> <b>parameter</b> <code>in ExifParameter</code>
ownedMember	<b>description</b> <b>parameter</b>
documentation	Constructor with all parameters for an axis. @param description text for the axis @param parameter the exif-parameter which will be evaluated for the axis

## Operation Axis::Axis

owner	<a href="#">Axis</a>
parameter	name direction type multiplicity default <b>parameter</b> <code>in ExifParameter</code>
ownedMember	<b>parameter</b>
documentation	Constructor with exif parameter @param parameter the exif-parameter which will be evaluated for the axis

### Property Axis::description

owner	<a href="#">Axis</a>
-------	----------------------

### Operation Axis::getDescription

owner	<a href="#">Axis</a>
parameter	name direction type multiplicity default <b>return return String</b>
ownedMember	<b>return</b>

### Operation Axis::getParameter

owner	<a href="#">Axis</a>
parameter	name direction type multiplicity default <b>return return ExifParameter</b>
ownedMember	<b>return</b>

### Property Axis::parameter

owner	<a href="#">Axis</a>
-------	----------------------

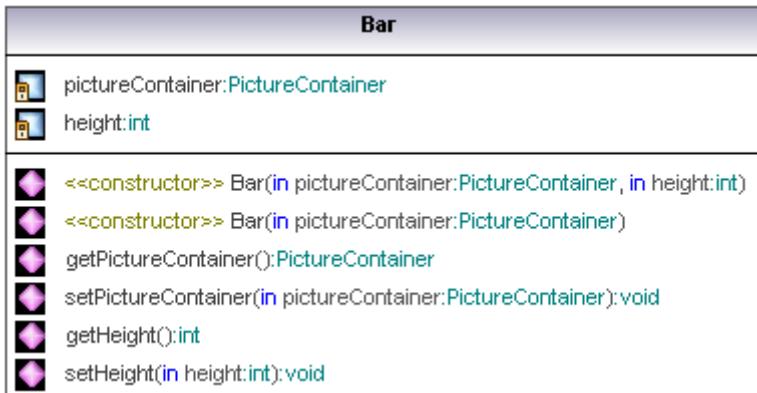
### Operation Axis::setDescription

owner	<a href="#">Axis</a>
parameter	name direction type multiplicity default <b>description in String return return void</b>
ownedMember	<b>description return</b>

### Operation Axis::setParameter

owner	<a href="#">Axis</a>
parameter	name direction type multiplicity default <b>parameter in ExifParameter return return void</b>
ownedMember	<b>parameter return</b>

## Class Bar

diagram	 <pre> classDiagram     class Bar {         PictureContainer pictureContainer         int height     }     Bar &lt;&gt; Bar :          &lt;&lt;constructor&gt;&gt; Bar(in pictureContainer:PictureContainer, in height:int)         &lt;&lt;constructor&gt;&gt; Bar(in pictureContainer:PictureContainer)         getPictureContainer():PictureContainer         setPictureContainer(in pictureContainer:PictureContainer):void         getHeight():int         setHeight(in height:int):void     end   </pre>
owner	<a href="#">reportmanagement</a>
ownedMember	<a href="#">Bar</a> <a href="#">Bar getHeight</a> <a href="#">getPictureContainer</a> <a href="#">height</a> <a href="#">pictureContainer</a> <a href="#">setHeight</a> <a href="#">setPictureContainer</a>
target of relation	ComponentRealization <a href="#">Implementierung</a>
typedElements	Class <a href="#">Category</a> Property <a href="#">bars</a> Operation <a href="#">Category getBars</a> <a href="#">setBars</a>
documentation	<p>Represents a bar which belongs to a picture category and PictureContainer.</p> <p>@author Kevin Zuber</p>

## Operation Bar::Bar

owner	<a href="#">Bar</a>
parameter	name direction type multiplicity default <a href="#">pictureContainer</a> in <a href="#">PictureContainer</a> <a href="#">height</a> in <a href="#">int</a>
ownedMember	<a href="#">height</a> <a href="#">pictureContainer</a>
documentation	<p>Constructor for the bar with pictureContainer and height parameter</p> <p>@param pictureContainer the picture container which is represented through the bar</p> <p>@param height the absolute height of the bar</p>

## Operation Bar::Bar

owner	<a href="#">Bar</a>
parameter	name direction type multiplicity default <a href="#">pictureContainer</a> in <a href="#">PictureContainer</a>
ownedMember	<a href="#">pictureContainer</a>
documentation	<p>Constructor for the bar with pictureContainer parameter</p> <p>@param pictureContainer the picture container which is represented through the bar</p>

## Operation Bar::getHeight

owner	<a href="#">Bar</a>
parameter	name direction type multiplicity default <b>return return int</b>

ownedMember	<b>return</b>
-------------	---------------

#### Operation **Bar::getPictureContainer**

owner	<b>Bar</b>
parameter	name direction type multiplicity default <b>return return PictureContainer</b>
ownedMember	<b>return</b>

#### Property **Bar::height**

owner	<b>Bar</b>
-------	------------

#### Property **Bar::pictureContainer**

owner	<b>Bar</b>
-------	------------

#### Operation **Bar::setHeight**

owner	<b>Bar</b>
parameter	name direction type multiplicity default <b>height in int return return void</b>
ownedMember	<b>height return</b>

#### Operation **Bar::setPictureContainer**

owner	<b>Bar</b>
parameter	name direction type multiplicity default <b>pictureContainer in PictureContainer return return void</b>
ownedMember	<b>pictureContainer return</b>

## Class Boxplot

diagram	<pre> <b>Boxplot</b>  mean:double median:double upperQuartile:double lowerQuartile:double upperWhisker:double lowerWhisker:double outlier:double[*] maxValue:double minValue:double PictureSetName:String  &lt;&lt;constructor&gt;&gt; Boxplot() &lt;&lt;constructor&gt;&gt; Boxplot(<b>in</b> pictures:<b>PictureContainer</b>) &lt;&lt;constructor&gt;&gt; Boxplot(<b>in</b> mean:<b>double</b>, <b>in</b> median:<b>double</b>, <b>in</b> upperQuartile:<b>double</b>, <b>in</b> lowerQuartile:<b>double</b>, <b>in</b> upperWhisker:<b>double</b>, <b>in</b> lowerWhisker:<b>double</b>, <b>in</b> outlier:<b>double[*]</b>, <b>in</b> maxValue:<b>double</b>, <b>in</b> minValue:<b>double</b>, <b>in</b> PictureSetName:<b>String</b>)  getMaxValue():double getMinValue():double getPictureSetName():String getMean():double getMedian():double getUpperQuartile():double getLowerQuartile():double getUpperWhisker():double getLowerWhisker():double getOutlier():double[*] </pre>
owner	<b>reportmanagement</b>
ownedMember	<b>Boxplot</b> <b>Boxplot</b> <b>Boxplot</b> <b>getLowerQuartile</b> <b>getLowerWhisker</b> <b>getMaxValue</b> <b>getMean</b> <b>getMedian</b> <b>getMinValue</b> <b>getOutlier</b> <b>getPictureSetName</b> <b>getUpperQuartile</b> <b>getUpperWhisker</b> <b>lowerQuartile</b> <b>lowerWhisker</b> <b>maxValue</b> <b>mean</b> <b>median</b> <b>minValue</b> <b>outlier</b> <b>PictureSetName</b> <b>upperQuartile</b> <b>upperWhisker</b>
target of relation	ComponentRealization <b>Implementierung</b>
typedElements	Class <b>BoxplotModel</b> Property <b>boxplots</b> Operation <b>getBoxplots</b>
documentation	<p>Represents the boxplot with all parts containing to it and calculate them.  This is:  mean, median, upperQuartile, lowerQuartile, upperWhisker, lowerWhisker, some outlier, maximum Value, minimum Value  All calculation which is needed to generate these parts will happen here.  @author Kevin Zuber</p>

## Operation Boxplot::Boxplot

owner	<b>Boxplot</b>
-------	----------------

## Operation Boxplot::Boxplot

owner	<b>Boxplot</b>
parameter	name direction type multiplicity default <b>pictures</b> in <b>PictureContainer</b>
ownedMember	<b>pictures</b>
documentation	Calculates the boxplot from the pictures @param pictures the pictures which will be represented with the boxplot

## Operation Boxplot::Boxplot

owner	<b>Boxplot</b>
parameter	name direction type multiplicity default <b>mean</b> <b>in double</b> <b>median</b> <b>in double</b> <b>upperQuartile</b> <b>in double</b> <b>lowerQuartile</b> <b>in double</b> <b>upperWhisker</b> <b>in double</b> <b>lowerWhisker</b> <b>in double</b> <b>outlier</b> <b>in double</b> <b>* maxValue</b> <b>in double</b> <b>minValue</b> <b>in double</b>

	<b>double pictureSetName in String</b>
ownedMember	<b>lowerQuartile lowerWhisker maxValue mean median minValue outlier pictureSetName upperQuartile upperWhisker</b>
documentation	Generates a Boxplot with all parameters. Should only be used for testing. @param mean @param median @param upperQuartile @param lowerQuartile @param upperWhisker @param lowerWhisker @param outlier @param maxValue @param minValue @param pictureSetName

#### Operation **Boxplot::getLowerQuartile**

owner	<b>Boxplot</b>
parameter	name direction type multiplicity default <b>return return double</b>
ownedMember	<b>return</b>

#### Operation **Boxplot::getLowerWhisker**

owner	<b>Boxplot</b>
parameter	name direction type multiplicity default <b>return return double</b>
ownedMember	<b>return</b>

#### Operation **Boxplot::getMaxValue**

owner	<b>Boxplot</b>
parameter	name direction type multiplicity default <b>return return double</b>
ownedMember	<b>return</b>

#### Operation **Boxplot::getMean**

owner	<b>Boxplot</b>
parameter	name direction type multiplicity default <b>return return double</b>
ownedMember	<b>return</b>

#### Operation **Boxplot::getMedian**

owner	<b>Boxplot</b>
parameter	name direction type multiplicity default <b>return return double</b>
ownedMember	<b>return</b>

#### Operation **Boxplot::getMinValue**

owner	<b>Boxplot</b>
-------	----------------

parameter	name direction type multiplicity default <b>return return double</b>
ownedMember	<b>return</b>

#### Operation **Boxplot::getOutlier**

owner	<b>Boxplot</b>
parameter	name direction type multiplicity default <b>return return double *</b>
ownedMember	<b>return</b>

#### Operation **Boxplot::getPictureSetName**

owner	<b>Boxplot</b>
parameter	name direction type multiplicity default <b>return return String</b>
ownedMember	<b>return</b>

#### Operation **Boxplot::getUpperQuartile**

owner	<b>Boxplot</b>
parameter	name direction type multiplicity default <b>return return double</b>
ownedMember	<b>return</b>

#### Operation **Boxplot::getUpperWhisker**

owner	<b>Boxplot</b>
parameter	name direction type multiplicity default <b>return return double</b>
ownedMember	<b>return</b>

#### Property **Boxplot::lowerQuartile**

owner	<b>Boxplot</b>
-------	----------------

#### Property **Boxplot::lowerWhisker**

owner	<b>Boxplot</b>
-------	----------------

#### Property **Boxplot::maxValue**

owner	<b>Boxplot</b>
-------	----------------

#### Property **Boxplot::mean**

owner	<b>Boxplot</b>
-------	----------------

#### Property **Boxplot::median**

owner	<b>Boxplot</b>
-------	----------------

Property **Boxplot::minValue**

owner	<a href="#">Boxplot</a>
-------	-------------------------

Property **Boxplot::outlier**

owner	<a href="#">Boxplot</a>
-------	-------------------------

Property **Boxplot::PictureSetName**

owner	<a href="#">Boxplot</a>
-------	-------------------------

Property **Boxplot::upperQuartile**

owner	<a href="#">Boxplot</a>
-------	-------------------------

Property **Boxplot::upperWhisker**

owner	<a href="#">Boxplot</a>
-------	-------------------------

## Class BoxplotConfig

diagram	<p>The diagram shows a class named <b>BoxplotConfig</b> with a dependency relationship (indicated by a dashed line with a diamond) to another class <b>BoxplotModel</b>. The <b>BoxplotConfig</b> class has three compartments: a top compartment containing the class name, a middle compartment containing a final attribute <code>serialVersionUID:long=1L</code>, and a bottom compartment containing a constructor <code>&lt;&lt;constructor&gt;&gt; BoxplotConfig(in model:M)</code>.</p>
hierarchy	<p>The hierarchy diagram shows <b>BoxplotConfig</b> inheriting from <b>AbstractReportCompilation&lt;M-&gt;M&gt;</b>.</p>
owner	<a href="#">reportmanagement</a>
template parameters	name kind constrainingClassifier default <b>M</b> Class <b>BoxplotModel</b>
ownedMember	<a href="#">BoxplotConfig serialVersionUID</a>
general	<a href="#">AbstractReportCompilation&lt;M-&gt;M&gt;</a>
target of relation	ComponentRealization <a href="#">Implementierung</a>
documentation	<p>This class represents the Boxplot configuration with all its necessary panels.</p> <p>@author David Kaufman</p> <p>@param &lt;M&gt;</p>

## Operation BoxplotConfig::BoxplotConfig

owner	<a href="#">BoxplotConfig</a>
parameter	name direction type multiplicity default <b>model in M</b>
ownedMember	<b>model</b>
documentation	<p>Constructor which initialized the report with all its panels</p> <p>@param boxplotmodel the model which is used by the views</p>

## Property BoxplotConfig::serialVersionUID

owner	<a href="#">BoxplotConfig</a>
-------	-------------------------------

## Class BoxplotModel

diagram	<pre> <b>BoxplotModel</b> +-- boxplots:Boxplot[*] +-- wilcoxonTestActive:boolean +-- wilcoxonTestType:WilcoxonTestType +-- wilcoxonSignificance:float  &lt;&lt;constructor&gt;&gt; BoxplotModel() &lt;&lt;constructor&gt;&gt; BoxplotModel(in pictureContainer:ArrayList&lt;E-&gt;PictureContainer&gt;, in xAxi +-- getBoxplots():Boxplot[*] +-- getWilcoxonPValue():float +-- isWilcoxonTestActive():boolean +-- setWilcoxonTestActive(in wilcoxonTestActive:boolean):void +-- getWilcoxonTestType():WilcoxonTestType +-- setWilcoxonTestType(in wilcoxonTestType:WilcoxonTestType):void +-- getWilcoxonSignificance():float +-- setWilcoxonSignificance(in wilcoxonSignificance:float):void </pre>
hierarchy	<pre> classDiagram     class Observable     class AbstractModel     class AbstractReportModel     class AbstractSingleAxisModel     class BoxplotModel      Observable &lt; -- AbstractModel     AbstractModel &lt; -- AbstractReportModel     AbstractReportModel &lt; -- AbstractSingleAxisModel     AbstractSingleAxisModel &lt; -- BoxplotModel </pre>
owner	<a href="#">reportmanagement</a>
ownedMember	<a href="#">BoxplotModel</a> <a href="#">BoxplotModel</a> <a href="#">boxplots</a> <a href="#">getBoxplots</a> <a href="#">getWilcoxonPValue</a> <a href="#">getWilcoxonSignificance</a> <a href="#">getWilcoxonTestType</a> <a href="#">isWilcoxonTestActive</a> <a href="#">setWilcoxonSignificance</a> <a href="#">setWilcoxonTestActive</a> <a href="#">setWilcoxonTestType</a> <a href="#">wilcoxonSignificance</a> <a href="#">wilcoxonTestActive</a> <a href="#">wilcoxonTestType</a>
general	<a href="#">AbstractSingleAxisModel</a>
target of relation	ComponentRealization <a href="#">Implementierung</a>
documentation	Model which contain the Boxplots and name them with an Axis. @author Kevin Zuber

## Operation BoxplotModel::BoxplotModel

owner	<a href="#">BoxplotModel</a>
-------	------------------------------

## Operation BoxplotModel::BoxplotModel

owner	<a href="#">BoxplotModel</a>
-------	------------------------------

parameter	name direction type multiplicity default <b>pictureContainer</b> in <i>ArrayList&lt;E&gt;PictureContainer&gt;</i> <b>xAxis</b> in <i>Axis</i>
ownedMember	<b>pictureContainer</b> <b>xAxis</b>
documentation	Constructor for the Boxplot Model @param pictureContainer @param xAxis

### Property **BoxplotModel::boxplots**

owner	<b>BoxplotModel</b>
-------	---------------------

### Operation **BoxplotModel::getBoxplots**

owner	<b>BoxplotModel</b>
parameter	name direction type multiplicity default <b>return return</b> <b>Boxplot</b> *
ownedMember	<b>return</b>

### Operation **BoxplotModel::getWilcoxonPValue**

owner	<b>BoxplotModel</b>
parameter	name direction type multiplicity default <b>return return</b> <b>float</b>
ownedMember	<b>return</b>
documentation	@return the result of the Wilcoxon test

### Operation **BoxplotModel::getWilcoxonSignificance**

owner	<b>BoxplotModel</b>
parameter	name direction type multiplicity default <b>return return</b> <b>float</b>
ownedMember	<b>return</b>

### Operation **BoxplotModel::getWilcoxonTestType**

owner	<b>BoxplotModel</b>
parameter	name direction type multiplicity default <b>return return</b> <b>WilcoxonTestType</b>
ownedMember	<b>return</b>

### Operation **BoxplotModel::isWilcoxonTestActive**

owner	<b>BoxplotModel</b>
parameter	name direction type multiplicity default <b>return return</b> <b>boolean</b>
ownedMember	<b>return</b>
documentation	@return if the wilcoxon test is used

#### Operation **BoxplotModel::setWilcoxonSignificance**

owner	<a href="#">BoxplotModel</a>
parameter	name direction type multiplicity default <b>wilcoxonSignificance</b> in float return return void
ownedMember	<b>return wilcoxonSignificance</b>

#### Operation **BoxplotModel::setWilcoxonTestActive**

owner	<a href="#">BoxplotModel</a>
parameter	name direction type multiplicity default <b>wilcoxonTestActive</b> in boolean return return void
ownedMember	<b>return wilcoxonTestActive</b>

#### Operation **BoxplotModel::setWilcoxonTestType**

owner	<a href="#">BoxplotModel</a>
parameter	name direction type multiplicity default <b>wilcoxonTestType</b> in <a href="#">WilcoxonTestType</a> return return void
ownedMember	<b>return wilcoxonTestType</b>

#### Property **BoxplotModel::wilcoxonSignificance**

owner	<a href="#">BoxplotModel</a>
-------	------------------------------

#### Property **BoxplotModel::wilcoxonTestActive**

owner	<a href="#">BoxplotModel</a>
-------	------------------------------

#### Property **BoxplotModel::wilcoxonTestType**

owner	<a href="#">BoxplotModel</a>
-------	------------------------------

## Class Category

diagram	<table border="1"> <thead> <tr> <th colspan="2">Category</th></tr> </thead> <tbody> <tr> <td> bars:Bar[*]</td><td></td></tr> <tr> <td> positionX:int</td><td></td></tr> <tr> <td> positionZ:int</td><td></td></tr> <tr> <td> &lt;&lt;constructor&gt;&gt; Category(<b>in</b> bars:Bar[*])</td><td></td></tr> <tr> <td> getBars():Bar[*]</td><td></td></tr> <tr> <td> setBars(<b>in</b> bars:Bar[*]):void</td><td></td></tr> <tr> <td> getPositionX():int</td><td></td></tr> <tr> <td> setPositionX(<b>in</b> positionX:int):void</td><td></td></tr> <tr> <td> getPositionZ():int</td><td></td></tr> <tr> <td> setPositionZ(<b>in</b> positionZ:int):void</td><td></td></tr> </tbody> </table>	Category		 bars:Bar[*]		 positionX:int		 positionZ:int		 <<constructor>> Category( <b>in</b> bars:Bar[*])		 getBars():Bar[*]		 setBars( <b>in</b> bars:Bar[*]):void		 getPositionX():int		 setPositionX( <b>in</b> positionX:int):void		 getPositionZ():int		 setPositionZ( <b>in</b> positionZ:int):void	
Category																							
 bars:Bar[*]																							
 positionX:int																							
 positionZ:int																							
 <<constructor>> Category( <b>in</b> bars:Bar[*])																							
 getBars():Bar[*]																							
 setBars( <b>in</b> bars:Bar[*]):void																							
 getPositionX():int																							
 setPositionX( <b>in</b> positionX:int):void																							
 getPositionZ():int																							
 setPositionZ( <b>in</b> positionZ:int):void																							
owner	<a href="#">reportmanagement</a>																						
ownedMember	<b>bars</b> Category <b>getBars</b> <b>getPositionX</b> <b>getPositionZ</b> <b>positionX</b> <b>positionZ</b> <b>setBars</b> <b>setPositionX</b> <b>setPositionZ</b>																						
target of relation	ComponentRealization <a href="#">Implementierung</a>																						
typedElements	Class <a href="#">Histogram2DModel</a> Property <b>categories</b> Operation <a href="#">getCategories</a> Class <a href="#">Histogram3DModel</a> Property <b>categories</b> Operation <a href="#">getCategories</a>																						
documentation	<p>Represents a class/category (statistical meaning) of bars in a histogram and allocate them to a PictureContainer.</p> <p>@author Kevin Zuber</p>																						

## Property Category::bars

owner	<a href="#">Category</a>
-------	--------------------------

## Operation Category::Category

owner	<a href="#">Category</a>
parameter	name direction type multiplicity default <b>bars</b> <b>in</b> <a href="#">Bar</a> *
ownedMember	<b>bars</b>
documentation	Constructor for the category @param bars

## Operation Category::getBars

owner	<a href="#">Category</a>
parameter	name direction type multiplicity default <b>return</b> <b>return</b> <a href="#">Bar</a> *
ownedMember	<b>return</b>

#### Operation **Category::getPositionX**

owner	<a href="#">Category</a>
parameter	name direction type multiplicity default <b>return return int</b>
ownedMember	<b>return</b>

#### Operation **Category::getPositionZ**

owner	<a href="#">Category</a>
parameter	name direction type multiplicity default <b>return return int</b>
ownedMember	<b>return</b>

#### Property **Category::positionX**

owner	<a href="#">Category</a>
-------	--------------------------

#### Property **Category::positionZ**

owner	<a href="#">Category</a>
-------	--------------------------

#### Operation **Category::setBars**

owner	<a href="#">Category</a>
parameter	name direction type multiplicity default <b>bars in Bar * return return void</b>
ownedMember	<b>bars return</b>

#### Operation **Category::setPositionX**

owner	<a href="#">Category</a>
parameter	name direction type multiplicity default <b>positionX in int return return void</b>
ownedMember	<b>positionX return</b>
documentation	Sets the x position of the category in the ground of the diagram. @param positionX

#### Operation **Category::setPositionZ**

owner	<a href="#">Category</a>
parameter	name direction type multiplicity default <b>positionZ in int return return void</b>
ownedMember	<b>positionZ return</b>
documentation	Sets the z position of the category in the ground of the diagram. Should be zero in 2D diagrams. @param positionZ



## Class Cluster3DConfig

diagram	<pre> classDiagram     class Cluster3DConfig {         &lt;&lt;final&gt;&gt; serialVersionUID:long=1L         &lt;&lt;constructor&gt;&gt; Cluster3DConfig(in model:M)     }     M "1..&gt;&gt;" Cluster3DConfig   </pre>
hierarchy	<pre> classDiagram     class Cluster3DConfig {         &lt;&lt;final&gt;&gt; serialVersionUID:long=1L         &lt;&lt;constructor&gt;&gt; Cluster3DConfig(in model:M)     }     Cluster3DConfig &lt; -- AbstractReportCompilation&lt;M-&gt;M&gt;   </pre>
owner	<a href="#">reportmanagement</a>
template parameters	name kind constrainingClassifier default <b>M</b> Class <a href="#">Cluster3DModel</a>
ownedMember	<a href="#">Cluster3DConfig serialVersionUID</a>
general	<a href="#">AbstractReportCompilation&lt;M-&gt;M&gt;</a>
target of relation	ComponentRealization <a href="#">Implementierung</a>
documentation	<p>This class represents the 3D Cluster configuration with all its necessary panels.</p> <p>@author David Kaufman</p> <p>@param &lt;M&gt;</p>

## Operation Cluster3DConfig::Cluster3DConfig

owner	<a href="#">Cluster3DConfig</a>
parameter	name direction type multiplicity default <b>model in M</b>
ownedMember	<b>model</b>
documentation	<p>Constructor which initialized the report with all its panels</p> <p>@param cluster3Dmodel the model which is used by the panels</p>

## Property Cluster3DConfig::serialVersionUID

owner	<a href="#">Cluster3DConfig</a>
-------	---------------------------------

## Class Cluster3DModel

diagram	<p style="text-align: center;"><b>Cluster3DModel</b></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;">frequency3DPoints: Frequency3DPoint[*]</td></tr> <tr> <td style="padding: 2px;">&lt;&lt;constructor&gt;&gt; Cluster3DModel(<b>in</b> pictureContainer: ArrayList&lt;E-&gt;PictureContainer&gt;, <b>in</b> xAxis: Axis, <b>in</b> zAxis: Axis, <b>in</b> yAxis: Axis)</td></tr> <tr> <td style="padding: 2px;">&lt;&lt;constructor&gt;&gt; Cluster3DModel()</td></tr> <tr> <td style="padding: 2px;">getFrequency3DPoints(): Frequency3DPoint[*]</td></tr> </table>	frequency3DPoints: Frequency3DPoint[*]	<<constructor>> Cluster3DModel( <b>in</b> pictureContainer: ArrayList<E->PictureContainer>, <b>in</b> xAxis: Axis, <b>in</b> zAxis: Axis, <b>in</b> yAxis: Axis)	<<constructor>> Cluster3DModel()	getFrequency3DPoints(): Frequency3DPoint[*]
frequency3DPoints: Frequency3DPoint[*]					
<<constructor>> Cluster3DModel( <b>in</b> pictureContainer: ArrayList<E->PictureContainer>, <b>in</b> xAxis: Axis, <b>in</b> zAxis: Axis, <b>in</b> yAxis: Axis)					
<<constructor>> Cluster3DModel()					
getFrequency3DPoints(): Frequency3DPoint[*]					
hierarchy	<pre> classDiagram     class AbstractModel {         &lt;&lt;Abstract Model&gt;&gt;     }     class AbstractReportModel {         &lt;&lt;Abstract Report Model&gt;&gt;     }     class AbstractSingleAxisModel {         &lt;&lt;Abstract Single Axis Model&gt;&gt;     }     class AbstractDoubleAxesModel {         &lt;&lt;Abstract Double Axes Model&gt;&gt;     }     class AbstractTripleAxesModel {         &lt;&lt;Abstract Triple Axes Model&gt;&gt;     }     class Cluster3DModel {         &lt;&lt;Cluster3D Model&gt;&gt;     }      AbstractModel &lt; -- AbstractReportModel     AbstractReportModel &lt; -- AbstractSingleAxisModel     AbstractSingleAxisModel &lt; -- AbstractDoubleAxesModel     AbstractDoubleAxesModel &lt; -- AbstractTripleAxesModel     AbstractTripleAxesModel &lt; -- Cluster3DModel   </pre>				
owner	<a href="#">reportmanagement</a>				
ownedMember	<a href="#">Cluster3DModel</a> <a href="#">Cluster3DModel frequency3DPoints getFrequency3DPoints</a>				
general	<a href="#">AbstractTripleAxesModel</a>				
target of relation	ComponentRealization <a href="#">Implementierung</a>				
documentation	<p>Represents the 3D-cluster and allocate the axes to it.</p> <p>@author Kevin Zuber</p>				

## Operation Cluster3DModel::Cluster3DModel

owner	<a href="#">Cluster3DModel</a>
parameter	name direction type multiplicity default pictureContainer in ArrayList<E->PictureContainer> xAxis in Axis zAxis in Axis yAxis in Axis
ownedMember	<a href="#">pictureContainer</a> <a href="#">xAxis</a> <a href="#">yAxis</a> <a href="#">zAxis</a>

## Operation Cluster3DModel::Cluster3DModel

owner	<a href="#">Cluster3DModel</a>
-------	--------------------------------

## Property Cluster3DModel::frequency3DPoints

owner	<a href="#">Cluster3DModel</a>
-------	--------------------------------

Operation **Cluster3DModel::getFrequency3DPoints**

owner	<a href="#">Cluster3DModel</a>
parameter	name direction type multiplicity default <b>return</b> <b>return</b> <a href="#">Frequency3DPoint</a> *
ownedMember	<b>return</b>

## Class ComplexCellRenderer

diagram	<pre> classDiagram     class ComplexCellRenderer {         DefaultListCellRenderer defaultRenderer         Component getListCellRendererComponent(list: JList, value: Object, index: int, isSelected: boolean, cellHasFocus: boolean)     } </pre>
hierarchy	<pre> classDiagram     class ListCellRenderer     class ComplexCellRenderer {         &lt; -- ListCellRenderer     } </pre>
owner	<a href="#">reportmanagement</a>
ownedMember	<a href="#">defaultRenderer getListCellRendererComponent</a>
implemented interfaces	<a href="#">ListCellRenderer</a>
source of relation	InterfaceRealization <a href="#">ListCellRenderer</a>
target of relation	ComponentRealization <a href="#">Implementierung</a>

### Property ComplexCellRenderer::defaultRenderer

owner	<a href="#">ComplexCellRenderer</a>
-------	-------------------------------------

### Operation ComplexCellRenderer::getListCellRendererComponent

owner	<a href="#">ComplexCellRenderer</a>
parameter	name direction type multiplicity default list in JList value in Object index in int isSelected in boolean cellHasFocus in boolean return return Component
ownedMember	<b>cellHasFocus index isSelected list return value</b>

## Package Component View

owner	<a href="#">Root</a>
ownedMember	<a href="#">Implementierung</a>

## Class DemoJFileChooser

diagram	<p><b>DemoJFileChooser</b></p> <table border="1"> <tr> <td> <code>&lt;&lt;final&gt;&gt; serialVersionUID:long=1L</code></td></tr> <tr> <td> <code>chooser:JFileChooser</code></td></tr> <tr> <td> <code>choosertitle:String</code></td></tr> <tr> <td> </td></tr> <tr> <td> <code>&lt;&lt;constructor&gt;&gt; DemoJFileChooser(in model:AbstractModel)</code></td></tr> <tr> <td> <code>getPreferredSize():Dimension</code></td></tr> <tr> <td> <code>&lt;&lt;annotations&gt;&gt; update(in model:Observable, in argument:Object):void</code></td></tr> </table>	<code>&lt;&lt;final&gt;&gt; serialVersionUID:long=1L</code>	<code>chooser:JFileChooser</code>	<code>choosertitle:String</code>	 	<code>&lt;&lt;constructor&gt;&gt; DemoJFileChooser(in model:AbstractModel)</code>	<code>getPreferredSize():Dimension</code>	<code>&lt;&lt;annotations&gt;&gt; update(in model:Observable, in argument:Object):void</code>	<p><b>update() (Operation)</b></p> <p><code>&lt;&lt;annotations&gt;&gt; modifiers = Override</code></p>
<code>&lt;&lt;final&gt;&gt; serialVersionUID:long=1L</code>									
<code>chooser:JFileChooser</code>									
<code>choosertitle:String</code>									
<code>&lt;&lt;constructor&gt;&gt; DemoJFileChooser(in model:AbstractModel)</code>									
<code>getPreferredSize():Dimension</code>									
<code>&lt;&lt;annotations&gt;&gt; update(in model:Observable, in argument:Object):void</code>									
hierarchy	<pre> classDiagram     class Container {         &lt;&lt;Accessible&gt;&gt;     }     class Window {         &lt;&lt;Accessible&gt;&gt;     }     class Frame {         &lt;&lt;Window, Accessible&gt;&gt;     }     class JFrame {         &lt;&lt;Frame, VWindowConstants, Accessible, RootPaneContainer&gt;&gt;     }     class JAbstractView {         &lt;&lt;JFrame, Observer&gt;&gt;     }     class DemoJFileChooser {         &lt;&lt;JAbstractView&gt;&gt;     }     class MenuContainer {         &lt;&lt;Window&gt;&gt;     }     class VWindowConstants     class Accessible     class RootPaneContainer     class Observer   </pre>								
owner	<a href="#">FileChooser</a>								
ownedMember	<a href="#">chooser</a> <a href="#">choosertitle</a> <a href="#">DemoJFileChooser</a> <a href="#">getPreferredSize</a> <a href="#">serialVersionUID</a> <a href="#">update</a>								
general	<a href="#">JAbstractView</a>								
target of relation		ComponentRealization <a href="#">Implementierung</a>							

### Property DemoJFileChooser::chooser

owner	<a href="#">DemoJFileChooser</a>
-------	----------------------------------

### Property DemoJFileChooser::choosertitle

owner	<a href="#">DemoJFileChooser</a>
-------	----------------------------------

### Operation DemoJFileChooser::DemoJFileChooser

owner	<a href="#">DemoJFileChooser</a>
parameter	name direction type multiplicity default <code>model in AbstractModel</code>
ownedMember	<code>model</code>

### Operation DemoJFileChooser::getPreferredSize

owner	<a href="#">DemoJFileChooser</a>
parameter	name direction type multiplicity default <code>return return Dimension</code>
ownedMember	<code>return</code>

Property **DemoJFileChooser::serialVersionUID**

owner	<a href="#">DemoJFileChooser</a>
-------	----------------------------------

Operation **DemoJFileChooser::update**

owner	<a href="#">DemoJFileChooser</a>
parameter	name direction type multiplicity default <b>model in Observable argument in Object return void</b>
ownedMember	<b>argument model return</b>

## Class DemoJFileChooser

diagram	<pre> class DemoJFileChooser {     JButton go;     JFileChooser chooser;     String choosertitle;      &lt;&lt;constructor&gt;&gt; DemoJFileChooser()     actionPerformed(in e:ActionEvent):void     getPreferredSize():Dimension     main(in s:String[*]):void } </pre>
hierarchy	<pre> class DemoJFileChooser {     JButton go;     JFileChooser chooser;     String choosertitle;      &lt;&lt;constructor&gt;&gt; DemoJFileChooser()     actionPerformed(in e:ActionEvent):void     getPreferredSize():Dimension     main(in s:String[*]):void }  class JPanel {     JComponent component }  class JComponent {     Container container }  class Container {     Component component }  class Component {     Serializable serializable }  class Serializable {     Accessible accessible     EventListener eventListener     ActionListener actionPerformed } </pre>
owner	<a href="#">fileChooser</a>
ownedMember	<a href="#">actionPerformed</a> <a href="#">chooser</a> <a href="#">choosertitle</a> <a href="#">DemoJFileChooser</a> <a href="#">getPreferredSize</a> <a href="#">go</a> <a href="#">main</a>
general	<a href="#">JPanel</a>
implemented interfaces	<a href="#">ActionListener</a>
source of relation	InterfaceRealization <a href="#">ActionListener</a>
target of relation	ComponentRealization <a href="#">Implementierung</a>

## Operation DemoJFileChooser::actionPerformed

owner	<a href="#">DemoJFileChooser</a>
parameter	name direction type multiplicity default <b>e</b> in ActionEvent return return void
ownedMember	<b>e</b> return

## Property DemoJFileChooser::chooser

owner	<a href="#">DemoJFileChooser</a>
-------	----------------------------------

## Property DemoJFileChooser::choosertitle

owner	<a href="#">DemoJFileChooser</a>
-------	----------------------------------

Operation **DemoJFileChooser::DemoJFileChooser**

owner	<a href="#">DemoJFileChooser</a>
-------	----------------------------------

Operation **DemoJFileChooser::getPreferredSize**

owner	<a href="#">DemoJFileChooser</a>
parameter	name direction type multiplicity default <b>return return Dimension</b>
ownedMember	<b>return</b>

Property **DemoJFileChooser::go**

owner	<a href="#">DemoJFileChooser</a>
-------	----------------------------------

Operation **DemoJFileChooser::main**

owner	<a href="#">DemoJFileChooser</a>
parameter	name direction type multiplicity default <b>s in String * return return void</b>
ownedMember	<b>return s</b>

## Class DemoJFileChooser

diagram	<pre> class DemoJFileChooser {     JButton go;     JFileChooser chooser;     String choosertitle;      &lt;&lt;constructor&gt;&gt; DemoJFileChooser()     actionPerformed(in e:ActionEvent):void     getPreferredSize():Dimension     main(in s:String[*]):void } </pre>
hierarchy	<pre> class DemoJFileChooser {     JButton go;     JFileChooser chooser;     String choosertitle;      &lt;&lt;constructor&gt;&gt; DemoJFileChooser()     actionPerformed(in e:ActionEvent):void     getPreferredSize():Dimension     main(in s:String[*]):void }  class JPanel {     JComponent component; }  class JComponent {     Container container; }  class Container {     Component component; }  class Component {     Serializable serializable; }  class Serializable {     Accessible accessible;     EventListener eventListener;     ActionListener actionPerformed; }  DemoJFileChooser --&gt; JPanel : component  JPanel --&gt; JComponent : component  JComponent --&gt; Container : container  Container --&gt; Component : component  Component --&gt; Serializable : serializable  Serializable --&gt; Accessible : accessible  Serializable --&gt; EventListener : eventListener  Serializable --&gt; ActionListener : actionPerformed </pre>
owner	<a href="#">fileChoser</a>
ownedMember	<a href="#">actionPerformed</a> <a href="#">chooser</a> <a href="#">choosertitle</a> <a href="#">DemoJFileChooser</a> <a href="#">getPreferredSize</a> <a href="#">go</a> <a href="#">main</a>
general	<a href="#">JPanel</a>
implemented interfaces	<a href="#">ActionListener</a>
source of relation	InterfaceRealization <a href="#">ActionListener</a>
target of relation	ComponentRealization <a href="#">Implementierung</a>

## Operation DemoJFileChooser::actionPerformed

owner	<a href="#">DemoJFileChooser</a>
parameter	name direction type multiplicity default <b>e</b> in ActionEvent return return void
ownedMember	<b>e</b> return

## Property DemoJFileChooser::chooser

owner	<a href="#">DemoJFileChooser</a>
-------	----------------------------------

## Property DemoJFileChooser::choosertitle

owner	<a href="#">DemoJFileChooser</a>
-------	----------------------------------

Operation **DemoJFileChooser::DemoJFileChooser**

owner	<a href="#">DemoJFileChooser</a>
-------	----------------------------------

Operation **DemoJFileChooser::getPreferredSize**

owner	<a href="#">DemoJFileChooser</a>
parameter	name direction type multiplicity default <b>return return Dimension</b>
ownedMember	<b>return</b>

Property **DemoJFileChooser::go**

owner	<a href="#">DemoJFileChooser</a>
-------	----------------------------------

Operation **DemoJFileChooser::main**

owner	<a href="#">DemoJFileChooser</a>
parameter	name direction type multiplicity default <b>s in String * return return void</b>
ownedMember	<b>return s</b>

## Class DemoThumbnailQuality

diagram	<p><b>DemoThumbnailQuality</b></p> <ul style="list-style-type: none"> <li><b>tPath:</b> String = "Implementierung/src/org/studien/thumbnail/"</li> <li><b>main(in args: String[*]): void</b></li> <li><b>getThumbOf(in bImage: BufferedImage, in maxWidthOrHeight: int, in hints: int): BufferedImage</b></li> </ul>	
owner	<b>thumbnail</b>	
ownedMember	<b>getThumbOf main tPath</b>	
target of relation		ComponentRealization <b>Implementierung</b>

## Operation DemoThumbnailQuality::getThumbOf

owner	<b>DemoThumbnailQuality</b>
parameter	name direction type multiplicity default <b>bImage</b> in <b>BufferedImage</b> <b>maxWidthOrHeight</b> in <b>int</b> <b>hints</b> in <b>int</b> return <b>BufferedImage</b>
ownedMember	<b>bImage hints maxWidthOrHeight return</b>

## Operation DemoThumbnailQuality::main

owner	<b>DemoThumbnailQuality</b>
parameter	name direction type multiplicity default <b>args</b> in <b>String *</b> return <b>return void</b>
ownedMember	<b>args return</b>

## Property DemoThumbnailQuality::tPath

owner	<b>DemoThumbnailQuality</b>
documentation	@param args @throws IOException

## Class DemoThumbnailSpeed

diagram	<pre> class DemoThumbnailSpeed {     long zstVorher;     long zstNachher;     long zstGesamt = 0;      void main(String[] args);     Image getThumbOf(BufferedImage bImage, int maxWidthOrHeight, int hints);     void measurepoint(String comment); } </pre>
owner	<a href="#">thumbnail</a>
ownedMember	<a href="#">getThumbOf</a> <a href="#">main</a> <a href="#">measurepoint</a> <a href="#">zstGesamt</a> <a href="#">zstNachher</a> <a href="#">zstVorher</a>
target of relation	ComponentRealization <a href="#">Implementierung</a>

## Operation DemoThumbnailSpeed::getThumbOf

owner	<a href="#">DemoThumbnailSpeed</a>
parameter	name direction type multiplicity default <b>bImage</b> in <b>BufferedImage</b> <b>maxWidthOrHeight</b> in <b>int</b> <b>hints</b> in <b>int</b> return <b>Image</b>
ownedMember	<b>bImage</b> <b>hints</b> <b>maxWidthOrHeight</b> return

## Operation DemoThumbnailSpeed::main

owner	<a href="#">DemoThumbnailSpeed</a>
parameter	name direction type multiplicity default <b>args</b> in <b>String</b> * return return <b>void</b>
ownedMember	<b>args</b> return

## Operation DemoThumbnailSpeed::measurepoint

owner	<a href="#">DemoThumbnailSpeed</a>
parameter	name direction type multiplicity default <b>comment</b> in <b>String</b> return return <b>void</b>
ownedMember	<b>comment</b> return

## Property DemoThumbnailSpeed::zstGesamt

owner	<a href="#">DemoThumbnailSpeed</a>
-------	------------------------------------

## Property DemoThumbnailSpeed::zstNachher

owner	<a href="#">DemoThumbnailSpeed</a>
-------	------------------------------------

## Property DemoThumbnailSpeed::zstVorher

owner	<a href="#">DemoThumbnailSpeed</a>
documentation	@param args

	<code>@throws IOException</code>	
--	----------------------------------	--

## Class DemoTooltip

diagram	
owner	<a href="#">tooltip</a>
ownedMember	<a href="#">main</a>
target of relation	ComponentRealization <a href="#">Implementierung</a>

## Operation DemoTooltip::main

owner	<a href="#">DemoTooltip</a>
parameter	name direction type multiplicity default <b>args in String * return void</b>
ownedMember	<b>args return</b>

## Class DiagramCloseController

diagram	<p>The diagram shows the class <b>DiagramCloseController</b> with the following details:</p> <ul style="list-style-type: none"> <li><b>Generalization:</b> It is a generalization of <b>AbstractController&lt;M-&gt;M,V-&gt;V&gt;</b>.</li> <li><b>Operations:</b> <ul style="list-style-type: none"> <li><b>actionPerformed() (Operation)</b>: An operation with annotations indicating it is an override.</li> <li><b>&lt;&lt;constructor&gt;&gt; DiagramCloseController(in view:V)</b>: A constructor.</li> <li><b>&lt;&lt;annotations&gt;&gt; actionPerformed(in e:ActionEvent):void</b>: Another annotation for the actionPerformed operation.</li> </ul> </li> </ul>
hierarchy	<pre> classDiagram     class AbstractController {         &lt;&lt;AbstractController&lt;M-&gt;M,V-&gt;V&gt;&gt;     }     class DiagramCloseController {         &lt;&lt;DiagramCloseController&gt;&gt;     }     AbstractController &lt; -- DiagramCloseController   </pre>
owner	<b>diagrams</b>
template parameters	name kind constrainingClassifier default <b>M Class V Class JAbstractDiagram&lt;M-&gt;?&gt;</b>
ownedMember	<b>actionPerformed DiagramCloseController</b>
general	<b>AbstractController&lt;M-&gt;M,V-&gt;V&gt;</b>
target of relation	ComponentRealization <b>Implementierung</b>
documentation	<p>This controller manages the closing of a diagram window</p> <p>@author David Kaufman</p>

## Operation DiagramCloseController::actionPerformed

owner	<b>DiagramCloseController</b>
parameter	name direction type multiplicity default <b>e in ActionEvent return return void</b>
ownedMember	<b>e return</b>

## Operation DiagramCloseController::DiagramCloseController

owner	<b>DiagramCloseController</b>
parameter	name direction type multiplicity default <b>view in V</b>
ownedMember	<b>view</b>

## Class Diagram ExportAsJPGController

diagram	<pre> classDiagram     class DiagramExportAsJPGController {         &lt;&lt;constructor&gt;&gt; DiagramExportAsJPGController(in diagram:V)         &lt;&lt;annotations&gt;&gt; actionPerformed(in e:ActionEvent):void     }     class AbstractController {         &lt;&lt;constructor&gt;&gt; AbstractController&lt;M-&gt;M,V-&gt;V&gt;     }     DiagramExportAsJPGController &lt; -- AbstractController     DiagramExportAsJPGController "1" --&gt; "1" JAbstractDiagram&lt;M-&gt;?     DiagramExportAsJPGController "1" --&gt; "1" ActionEvent "e"   </pre>
hierarchy	<pre> classDiagram     class AbstractController {         &lt;&lt;constructor&gt;&gt; AbstractController&lt;M-&gt;M,V-&gt;V&gt;     }     class DiagramExportAsJPGController {         &lt;&lt;constructor&gt;&gt; DiagramExportAsJPGController()     }     DiagramExportAsJPGController &lt; -- AbstractController   </pre>
owner	<a href="#">diagrams</a>
template parameters	name kind constrainingClassifier default <b>M Class V Class JAbstractDiagram&lt;M-&gt;?</b>
ownedMember	<a href="#">actionPerformed DiagramExportAsJPGController</a>
general	<a href="#">AbstractController&lt;M-&gt;M,V-&gt;V&gt;</a>
target of relation	ComponentRealization <a href="#">Implementierung</a>
documentation	<p>This controller manages the exportation of a screenshot of the current diagram.</p> <p>@author David Kaufman</p>

### Operation DiagramExportAsJPGController::actionPerformed

owner	<a href="#">DiagramExportAsJPGController</a>
parameter	name direction type multiplicity default <b>e in ActionEvent return return void</b>
ownedMember	<b>e return</b>

### Operation DiagramExportAsJPGController::DiagramExportAsJPGController

owner	<a href="#">DiagramExportAsJPGController</a>
parameter	name direction type multiplicity default <b>diagram in V</b>
ownedMember	<b>diagram</b>

## Class DiagramResetViewController

diagram	<p>The diagram shows the class <b>DiagramResetViewController</b> with its constructor and two operations: <code>actionPerformed()</code> and <code>actionPerformed(in e:ActionEvent):void</code>. It also shows its inheritance relationship from <b>AbstractController&lt;M-&gt;M,V-&gt;V&gt;</b>.</p>
hierarchy	<pre> classDiagram     class AbstractController {         &lt;&lt;constructor&gt;&gt; AbstractController&lt;M-&gt;M,V-&gt;V&gt;     }     class DiagramResetViewController {         &lt;&lt;constructor&gt;&gt; DiagramResetViewController(in view:V)         &lt;&lt;operations&gt;&gt; actionPerformed(in e:ActionEvent):void     }     AbstractController &lt; -- DiagramResetViewController   </pre>
owner	<a href="#">diagrams</a>
template parameters	name kind constrainingClassifier default <b>M Class V Class JAbstract3DView&lt;M-&gt;?&gt;</b>
ownedMember	<a href="#">actionPerformed DiagramResetViewController</a>
general	<a href="#">AbstractController&lt;M-&gt;M,V-&gt;V&gt;</a>
target of relation	ComponentRealization <a href="#">Implementierung</a>
documentation	<p>This controller resets the view to the default perspective setting of the specified JAbstract3DView.</p> <p>@author David Kaufman</p>

## Operation DiagramResetViewController::actionPerformed

owner	<a href="#">DiagramResetViewController</a>
parameter	name direction type multiplicity default <b>e in ActionEvent return return void</b>
ownedMember	<b>e return</b>

## Operation DiagramResetViewController::DiagramResetViewController

owner	<a href="#">DiagramResetViewController</a>
parameter	name direction type multiplicity default <b>view in V</b>
ownedMember	<b>view</b>

## Class DiagramTypeSelectController

diagram	<p><b>DiagramTypeSelectController</b></p> <p>view: JDiaDiagramType</p> <p>&lt;&lt;constructor&gt;&gt; DiagramTypeSelectController(in view: JDiaDiagramType)</p> <p>valueChanged(in e: ListSelectionEvent): void</p>
hierarchy	<pre> classDiagram     class DiagramTypeSelectController     class ListSelectionListener     class EventListener     DiagramTypeSelectController &lt; -- ListSelectionListener     ListSelectionListener &lt; -- EventListener   </pre>
owner	<a href="#">reportmanagement</a>
ownedMember	<a href="#">DiagramTypeSelectController valueChanged view</a>
implemented interfaces	<a href="#">ListSelectionListener</a>
source of relation	InterfaceRealization <a href="#">ListSelectionListener</a>
target of relation	ComponentRealization <a href="#">Implementierung</a>
documentation	<p>This controller is responsible for managing the update of the report configuration utility when the user clicks a entry in the list</p> <p>@author David Kaufman</p>

## Operation DiagramTypeSelectController::DiagramTypeSelectController

owner	<a href="#">DiagramTypeSelectController</a>
parameter	name direction type multiplicity default <b>view</b> in <a href="#">JDiaDiagramType</a>
ownedMember	<a href="#">view</a>

## Operation DiagramTypeSelectController::valueChanged

owner	<a href="#">DiagramTypeSelectController</a>
parameter	name direction type multiplicity default <b>e</b> in <a href="#">ListSelectionEvent</a> return return void
ownedMember	<a href="#">e return</a>

## Property DiagramTypeSelectController::view

owner	<a href="#">DiagramTypeSelectController</a>
-------	---



## Class Directory

diagram	<pre> <b>Directory</b> + pictures:List&lt;E-&gt;Picture&gt;=new LinkedList&lt;Picture&gt;() + directoryName:String + path:String  # &lt;&lt;constructor&gt;&gt; Directory() # getItems():List&lt;E-&gt;? # hasNext():boolean # next():Picture # remove():void # getName():String # setName(in name:String):void </pre>
hierarchy	<pre> classDiagram     class Directory     class PictureContainer     class Iterator&lt;E&gt;Picture      Directory &lt; -- PictureContainer     PictureContainer &lt; -- Iterator&lt;E&gt;Picture </pre>
owner	<a href="#">picturemanagement</a>
ownedMember	<a href="#">Directory</a> <a href="#">directoryName</a> <a href="#">getItems</a> <a href="#">getName</a> <a href="#">hasNext</a> <a href="#">next</a> <a href="#">path</a> <a href="#">pictures</a> <a href="#">remove</a> <a href="#">setName</a>
implemented Interfaces	<a href="#">PictureContainer</a>
source of relation	InterfaceRealization <a href="#">PictureContainer</a>
target of relation	ComponentRealization <a href="#">Implementierung</a>
typedElements	Class <a href="#">ProjectModel</a> Operation <a href="#">getDirectoriesOfAPictureSet</a> <a href="#">getPicturesOfADirectory</a>

## Operation **Directory::Directory**

owner	<a href="#">Directory</a>
-------	---------------------------

## Property **Directory::directoryName**

owner	<a href="#">Directory</a>
-------	---------------------------

## Operation **Directory::getItems**

owner	<a href="#">Directory</a>
parameter	name direction type multiplicity default <b>return return List&lt;E-&gt;?</b>
ownedMember	<b>return</b>

**Operation `Directory::getName`**

owner	<a href="#">Directory</a>
parameter	name direction type multiplicity default <b>return return String</b>
ownedMember	<b>return</b>

**Operation `Directory::hasNext`**

owner	<a href="#">Directory</a>
parameter	name direction type multiplicity default <b>return return boolean</b>
ownedMember	<b>return</b>

**Operation `Directory::next`**

owner	<a href="#">Directory</a>
parameter	name direction type multiplicity default <b>return return Picture</b>
ownedMember	<b>return</b>

**Property `Directory::path`**

owner	<a href="#">Directory</a>
-------	---------------------------

**Property `Directory::pictures`**

owner	<a href="#">Directory</a>
-------	---------------------------

**Operation `Directory::remove`**

owner	<a href="#">Directory</a>
parameter	name direction type multiplicity default <b>return return void</b>
ownedMember	<b>return</b>

**Operation `Directory::setName`**

owner	<a href="#">Directory</a>
parameter	name direction type multiplicity default <b>name in String return return void</b>
ownedMember	<b>name return</b>

## Class Directory

diagram	<pre> <b>Directory</b>     pictures:List&lt;E-&gt;Picture&gt;=new ArrayList&lt;Picture&gt;()      &lt;&lt;constructor&gt;&gt; Directory()     getItems():ArrayList&lt;E-&gt;?     hasNext():boolean     next():Object     remove():void   </pre>
hierarchy	<pre> classDiagram     class Directory {         pictures:List&lt;E-&gt;Picture&gt;=new ArrayList&lt;Picture&gt;()     }     class PictureContainer {         &lt;&lt;constructor&gt;&gt;         getItems():ArrayList&lt;E-&gt;?         hasNext():boolean         next():Object         remove():void     }     class Iterator     Directory &lt; -- PictureContainer     PictureContainer &lt; -- Iterator   </pre>
owner	<a href="#">Kompositum</a>
ownedMember	<a href="#">Directory</a> <a href="#">getItems</a> <a href="#">hasNext</a> <a href="#">next</a> <a href="#">pictures</a> <a href="#">remove</a>
implemented interfaces	<a href="#">PictureContainer</a>
source of relation	InterfaceRealization <a href="#">PictureContainer</a>
target of relation	ComponentRealization <a href="#">Implementierung</a>

## Operation **Directory::Directory**

owner	<a href="#">Directory</a>
-------	---------------------------

## Operation **Directory::getItems**

owner	<a href="#">Directory</a>
parameter	name direction type multiplicity default <b>return return ArrayList&lt;E-&gt;?</b>
ownedMember	<b>return</b>

## Operation **Directory::hasNext**

owner	<a href="#">Directory</a>
parameter	name direction type multiplicity default <b>return return boolean</b>
ownedMember	<b>return</b>

## Operation **Directory::next**

owner	<a href="#">Directory</a>
parameter	name direction type multiplicity default <b>return return Object</b>
ownedMember	<b>return</b>

	ber	
--	-----	--

### Property **Directory::pictures**

owner	<a href="#">Directory</a>
-------	---------------------------

### Operation **Directory::remove**

owner	<a href="#">Directory</a>
parameter	name direction type multiplicity default <b>return return void</b>
ownedMember	<b>return</b>

## Class ExifAdapter

diagram	<pre> <b>ExifAdapter</b> + exifData:JIfdData + strFilePath:String  # getExifParameter(in e:ExifParameter):Object # &lt;&lt;constructor&gt;&gt; ExifAdapter() # &lt;&lt;constructor&gt;&gt; ExifAdapter(in imageFile:File) # &lt;&lt;constructor&gt;&gt; ExifAdapter(in imageFile:String) # setFilePath(in imageFile:String):void # getCameraModel():String # getFlash():int # getFNumber():float # getFNumberAsFormattedString():String # getExposureTime():int # getExposureTimeAsFormattedString():String # getISOSpeedRatings():int # getFocalLength():float # getFocalLength35mm():float # getOriginalDate():String # getOriginalDayOfWeek():String # getOriginalTime():String # getObjective():String </pre>
owner	<a href="#">jexifviewer</a>
ownedMember	<a href="#">ExifAdapter</a> <a href="#">ExifAdapter</a> <a href="#">ExifAdapter</a> <a href="#">exifData</a> <a href="#">getCameraModel</a> <a href="#">getExifParameter</a> <a href="#">getExposureTime</a> <a href="#">getExposureTimeAsFormattedString</a> <a href="#">setFlash</a> <a href="#">getFNumber</a> <a href="#">getFNumberAsFormattedString</a> <a href="#">getFocalLength</a> <a href="#">getFocalLength35mm</a> <a href="#">getISOSpeedRatings</a> <a href="#">getObjective</a> <a href="#">getOriginalDate</a> <a href="#">getOriginalDayOfWeek</a> <a href="#">getOriginalTime</a> <a href="#">setFilePath</a> <a href="#">strFilePath</a>
target of relation	ComponentRealization <a href="#">Implementierung</a>

## Operation ExifAdapter::ExifAdapter

owner	<a href="#">ExifAdapter</a>
-------	-----------------------------

## Operation ExifAdapter::ExifAdapter

owner	<a href="#">ExifAdapter</a>
parameter	name direction type multiplicity default <b>imageFile</b> <b>in File</b>
ownedMember	<b>imageFile</b>

## Operation ExifAdapter::ExifAdapter

owner	<a href="#">ExifAdapter</a>
parameter	name direction type multiplicity default <b>imageFile</b> <b>in String</b>
ownedMember	<b>imageFile</b>

**Property ExifAdapter::exifData**

owner	<a href="#">ExifAdapter</a>
-------	-----------------------------

**Operation ExifAdapter::getCameraModel**

owner	<a href="#">ExifAdapter</a>
parameter	name direction type multiplicity default <b>return return String</b>
ownedMember	<b>return</b>

**Operation ExifAdapter::getExifParameter**

owner	<a href="#">ExifAdapter</a>
parameter	name direction type multiplicity default <b>e in ExifParameter return return Object</b>
ownedMember	<b>e return</b>

**Operation ExifAdapter::getExposureTime**

owner	<a href="#">ExifAdapter</a>
parameter	name direction type multiplicity default <b>return return int</b>
ownedMember	<b>return</b>

**Operation ExifAdapter::getExposureTimeAsFormattedString**

owner	<a href="#">ExifAdapter</a>
parameter	name direction type multiplicity default <b>return return String</b>
ownedMember	<b>return</b>

**Operation ExifAdapter::getFlash**

owner	<a href="#">ExifAdapter</a>
parameter	name direction type multiplicity default <b>return return int</b>
ownedMember	<b>return</b>

**Operation ExifAdapter::getFNumber**

owner	<a href="#">ExifAdapter</a>
parameter	name direction type multiplicity default <b>return return float</b>
ownedMember	<b>return</b>

**Operation ExifAdapter::getFNumberAsFormattedString**

owner	<a href="#">ExifAdapter</a>
parameter	name direction type multiplicity default <b>return return String</b>

ownedMember	<b>return</b>
-------------	---------------

#### Operation **ExifAdapter::getFocalLength**

owner	<b>ExifAdapter</b>
parameter	name direction type multiplicity default <b>return return float</b>
ownedMember	<b>return</b>

#### Operation **ExifAdapter::getFocalLength35mm**

owner	<b>ExifAdapter</b>
parameter	name direction type multiplicity default <b>return return float</b>
ownedMember	<b>return</b>

#### Operation **ExifAdapter::getISOSpeedRatings**

owner	<b>ExifAdapter</b>
parameter	name direction type multiplicity default <b>return return int</b>
ownedMember	<b>return</b>

#### Operation **ExifAdapter::getObjective**

owner	<b>ExifAdapter</b>
parameter	name direction type multiplicity default <b>return return String</b>
ownedMember	<b>return</b>

#### Operation **ExifAdapter::getOriginalDate**

owner	<b>ExifAdapter</b>
parameter	name direction type multiplicity default <b>return return String</b>
ownedMember	<b>return</b>

#### Operation **ExifAdapter::getOriginalDayOfWeek**

owner	<b>ExifAdapter</b>
parameter	name direction type multiplicity default <b>return return String</b>
ownedMember	<b>return</b>

#### Operation **ExifAdapter::getOriginalTime**

owner	<b>ExifAdapter</b>
parameter	name direction type multiplicity default <b>return return String</b>
ownedMember	<b>return</b>

### Operation **ExifAdapter::setFilePath**

owner	<a href="#">ExifAdapter</a>
parameter	name direction type multiplicity default <b>imageFile</b> in <b>String</b> return return void
ownedMember	<b>imageFile</b> return

### Property **ExifAdapter::strFilePath**

owner	<a href="#">ExifAdapter</a>
-------	-----------------------------

## Class ExifAdapter

diagram	<pre> classDiagram     class ExifAdapter {         exifData: JIfdData         strFilePath: String          &lt;&lt;constructor&gt;&gt; ExifAdapter()         &lt;&lt;constructor&gt;&gt; ExifAdapter(in imageFile: File)         &lt;&lt;constructor&gt;&gt; ExifAdapter(in imageFile: String)         setFilePath(in imageFile: String): void         getCameraModel(): String         getFlash(): int         getFNumber(): float         getFNumberAsFormattedString(): String         getExposureTime(): int         getExposureTimeAsFormattedString(): String         getISOSpeedRatings(): int         getFocalLength(): float         getFocalLength35mm(): float         getOriginalDate(): String         getOriginalDayOfWeek(): String         getOriginalTime(): String         getObjective(): String          &lt;&lt;static&gt;&gt; ExifParameter     }   </pre>
owner	jexifviewer
ownedMember	<b>ExifAdapter</b> <b>ExifAdapter</b> <b>ExifAdapter</b> <b>exifData</b> <b>ExifParameter</b> <b>getCameraModel</b> <b>getExifParameter</b> <b>getExposureTime</b> <b>getExposureTimeAsFormattedString</b> <b>setFlash</b> <b>getFNumber</b> <b>getFNumberAsFormattedString</b> <b>getFocalLength</b> <b>getFocalLength35mm</b> <b>getISOSpeedRatings</b> <b>getObjective</b> <b>getOriginalDate</b> <b>getOriginalDayOfWeek</b> <b>getOriginalTime</b> <b>setFilePath</b> <b>strFilePath</b>
target of relation	ComponentRealization <b>Implementierung</b>

## Operation ExifAdapter::ExifAdapter

owner	<b>ExifAdapter</b>
-------	--------------------

## Operation ExifAdapter::ExifAdapter

owner	<b>ExifAdapter</b>
parameter	name direction type multiplicity default <b>imageFile</b> <b>in File</b>
ownedMember	<b>imageFile</b>

## Operation ExifAdapter::ExifAdapter

owner	<b>ExifAdapter</b>
parameter	name direction type multiplicity default <b>imageFile</b> <b>in String</b>

ownedMember	<b>imageFile</b>
-------------	------------------

#### Property **ExifAdapter::exifData**

owner	<a href="#">ExifAdapter</a>
-------	-----------------------------

#### Enumeration **ExifAdapter::ExifParameter**

diagram	<pre> &lt;&lt;static&gt;&gt; &lt;&lt;enumeration&gt;&gt; <b>ExifParameter</b>  <b>ordinal:boolean</b>  &lt;&lt;constructor&gt;&gt; ExifParameter(<b>in ordinal:boolean</b>) <b>setOrdinal(<b>in ordinal:boolean</b>):void</b> <b>isOrdinal():boolean</b>  Kameramodell = false Blitz = false Blende = true Verschlusszeit = true ISO_Wert = true Brennweite = true Datum = true Wochentag = true Uhrzeit = true Objektivname = false </pre>
owner	<a href="#">ExifAdapter</a>
ownedMember	<a href="#">Blende</a> <a href="#">Blitz</a> <a href="#">Brennweite</a> <a href="#">Datum</a> <a href="#">ExifParameter</a> <a href="#">ISO_Wert</a> <a href="#">isOrdinal</a> <a href="#">Kameramodell</a> <a href="#">Objektivname</a> <a href="#">ordinal</a> <a href="#">setOrdinal</a> <a href="#">Uhrzeit</a> <a href="#">Verschlusszeit</a> <a href="#">Wochentag</a>
typedElements	Class <a href="#">ExifAdapter</a> Operation <a href="#">getExifParameter</a>

#### EnumerationLiteral **ExifAdapter::ExifParameter::Blende**

owner	<a href="#">ExifParameter</a>
-------	-------------------------------

#### EnumerationLiteral **ExifAdapter::ExifParameter::Blitz**

owner	<a href="#">ExifParameter</a>
-------	-------------------------------

#### EnumerationLiteral **ExifAdapter::ExifParameter::Brennweite**

owner	<a href="#">ExifParameter</a>
-------	-------------------------------

EnumerationLiteral **ExifAdapter::ExifParameter::Datum**

owner	<a href="#">ExifParameter</a>
-------	-------------------------------

Operation **ExifAdapter::ExifParameter::ExifParameter**

owner	<a href="#">ExifParameter</a>
parameter	name direction type multiplicity default <b>ordinal in boolean</b>
ownedMember	<b>ordinal</b>

EnumerationLiteral **ExifAdapter::ExifParameter::ISO\_Wert**

owner	<a href="#">ExifParameter</a>
-------	-------------------------------

Operation **ExifAdapter::ExifParameter::isOrdinal**

owner	<a href="#">ExifParameter</a>
parameter	name direction type multiplicity default <b>return return boolean</b>
ownedMember	<b>return</b>

EnumerationLiteral **ExifAdapter::ExifParameter::Kameramodell**

owner	<a href="#">ExifParameter</a>
-------	-------------------------------

EnumerationLiteral **ExifAdapter::ExifParameter::Objektivname**

owner	<a href="#">ExifParameter</a>
-------	-------------------------------

Property **ExifAdapter::ExifParameter::ordinal**

owner	<a href="#">ExifParameter</a>
-------	-------------------------------

Operation **ExifAdapter::ExifParameter::setOrdinal**

owner	<a href="#">ExifParameter</a>
parameter	name direction type multiplicity default <b>ordinal in boolean return return void</b>
ownedMember	<b>ordinal return</b>

EnumerationLiteral **ExifAdapter::ExifParameter::Uhrzeit**

owner	<a href="#">ExifParameter</a>
-------	-------------------------------

EnumerationLiteral **ExifAdapter::ExifParameter::Verschlusszeit**

owner	<a href="#">ExifParameter</a>
-------	-------------------------------

EnumerationLiteral **ExifAdapter::ExifParameter::Wochentag**

owner	<a href="#">ExifParameter</a>
-------	-------------------------------

Operation **ExifAdapter::getCameraModel**

owner	<a href="#">ExifAdapter</a>
parameter	name direction type multiplicity default <b>return return String</b>
ownedMember	<b>return</b>

Operation **ExifAdapter::getExifParameter**

owner	<a href="#">ExifAdapter</a>
parameter	name direction type multiplicity default <b>e in ExifParameter return return Object</b>
ownedMember	<b>e return</b>

Operation **ExifAdapter::getExposureTime**

owner	<a href="#">ExifAdapter</a>
parameter	name direction type multiplicity default <b>return return int</b>
ownedMember	<b>return</b>

Operation **ExifAdapter::getExposureTimeAsFormattedString**

owner	<a href="#">ExifAdapter</a>
parameter	name direction type multiplicity default <b>return return String</b>
ownedMember	<b>return</b>

Operation **ExifAdapter::getFlash**

owner	<a href="#">ExifAdapter</a>
parameter	name direction type multiplicity default <b>return return int</b>
ownedMember	<b>return</b>

Operation **ExifAdapter::getFNumber**

owner	<a href="#">ExifAdapter</a>
parameter	name direction type multiplicity default <b>return return float</b>
ownedMember	<b>return</b>

Operation **ExifAdapter::getFNumberAsFormattedString**

owner	<a href="#">ExifAdapter</a>
parameter	name direction type multiplicity default <b>return return String</b>
ownedMember	<b>return</b>

#### Operation **ExifAdapter::getFocalLength**

owner	<a href="#">ExifAdapter</a>
parameter	name direction type multiplicity default <b>return return float</b>
ownedMember	<b>return</b>

#### Operation **ExifAdapter::getFocalLength35mm**

owner	<a href="#">ExifAdapter</a>
parameter	name direction type multiplicity default <b>return return float</b>
ownedMember	<b>return</b>

#### Operation **ExifAdapter::getISOSpeedRatings**

owner	<a href="#">ExifAdapter</a>
parameter	name direction type multiplicity default <b>return return int</b>
ownedMember	<b>return</b>

#### Operation **ExifAdapter::getObjective**

owner	<a href="#">ExifAdapter</a>
parameter	name direction type multiplicity default <b>return return String</b>
ownedMember	<b>return</b>

#### Operation **ExifAdapter::getOriginalDate**

owner	<a href="#">ExifAdapter</a>
parameter	name direction type multiplicity default <b>return return String</b>
ownedMember	<b>return</b>

#### Operation **ExifAdapter::getOriginalDayOfWeek**

owner	<a href="#">ExifAdapter</a>
parameter	name direction type multiplicity default <b>return return String</b>
ownedMember	<b>return</b>

#### Operation **ExifAdapter::getOriginalTime**

owner	<a href="#">ExifAdapter</a>
parameter	name direction type multiplicity default <b>return return String</b>
ownedMember	<b>return</b>

Operation **ExifAdapter::setFilePath**

owner	<a href="#">ExifAdapter</a>
parameter	name direction type multiplicity default <b>imageFile</b> in <b>String</b> return return void
ownedMember	<b>imageFile</b> return

Property **ExifAdapter::strFilePath**

owner	<a href="#">ExifAdapter</a>
-------	-----------------------------

## Enumeration **ExifParameter**

diagram	<pre>&lt;&lt;enumeration&gt;&gt; <b>ExifParameter</b>  CAMERAMODEL FLASH FNUMBER EXPOSURETIME ISO FOCALLENGTH DATE DAYOFWEEK TIME OBJECTIVENAME</pre>
owner	<a href="#">utils</a>
ownedMember	<a href="#">CAMERAMODEL</a> <a href="#">DATE</a> <a href="#">DAYOFWEEK</a> <a href="#">EXPOSURETIME</a> <a href="#">FLASH</a> <a href="#">FNUMBER</a> <a href="#">FOCALLENGTH</a> <a href="#">ISO</a> <a href="#">OBJECTIVENAME</a> <a href="#">TIME</a>
target of relation	ComponentRealization <a href="#">Implementierung</a>
typedElements	Class <a href="#">Axis</a> Property <a href="#">parameter</a> Operation <a href="#">Axis getParameter</a> <a href="#">setParameter</a> Class <a href="#">ExifAdapter</a> Operation <a href="#">getExifParameter</a> Class <a href="#">Picture</a> Operation <a href="#">getExifParameter</a> Class <a href="#">PictureParameter</a> Property <a href="#">exifParameter</a> Operation <a href="#">getExifParameter</a> <a href="#">PictureParameter setExifParameter</a>

## EnumerationLiteral **ExifParameter::CAMERAMODEL**

owner	<a href="#">ExifParameter</a>
-------	-------------------------------

## EnumerationLiteral **ExifParameter::DATE**

owner	<a href="#">ExifParameter</a>
-------	-------------------------------

## EnumerationLiteral **ExifParameter::DAYOFWEEK**

owner	<a href="#">ExifParameter</a>
-------	-------------------------------

## EnumerationLiteral **ExifParameter::EXPOSURETIME**

owner	<a href="#">ExifParameter</a>
-------	-------------------------------

## EnumerationLiteral **ExifParameter::FLASH**

owner	<a href="#">ExifParameter</a>
-------	-------------------------------

## EnumerationLiteral **ExifParameter::FNUMBER**

owner	<a href="#">ExifParameter</a>
-------	-------------------------------

## EnumerationLiteral **ExifParameter::FOCALLENGTH**

owner	<a href="#">ExifParameter</a>
-------	-------------------------------

EnumerationLiteral **ExifParameter::ISO**

owner	<a href="#">ExifParameter</a>
-------	-------------------------------

EnumerationLiteral **ExifParameter::OBJECTIVENAME**

owner	<a href="#">ExifParameter</a>
-------	-------------------------------

EnumerationLiteral **ExifParameter::TIME**

owner	<a href="#">ExifParameter</a>
-------	-------------------------------

## Package **FileChooser**

owner	<a href="#"><u>utils</u></a>
ownedMember	<a href="#"><b>DemoJFileChooser</b></a>

UML documentation generated by [\*\*UModel\*\*](#) UML Editor <http://www.altova.com/umodel>

12/17/09 13:54:45

## Class FileHandler

diagram	<pre> classDiagram     class FileHandler {         &lt;&lt;final&gt;&gt; PROJECTS_PATH: String = System.getProperty("user.home") + File.separator + ".knipsX"         &lt;&lt;constructor&gt;&gt; FileHandler()         createNewProjectFile(in newProject: ProjectEntry): void         deleteProjectFile(in newProject: ProjectEntry): void         scanProjectDirectory(): List&lt;E-&gt;ProjectEntry&gt;         scanProjectFile(in projectId: int): ProjectModel         writeProjectToFile(in project: ProjectModel): void         manipulateFileNameAndID(in toManipulate: ProjectEntry): void         copyProject(in toCopy: ProjectEntry, in projectName: String): ProjectEntry     }   </pre>
owner	<a href="#">utils</a>
ownedMember	<a href="#">copyProject</a> <a href="#">createNewProjectFile</a> <a href="#">deleteProjectFile</a> <a href="#">FileHandler</a> <a href="#">manipulateFileNameAndID</a> <a href="#">PROJECTS_PATH</a> <a href="#">scanProjectDirectory</a> <a href="#">scanProjectFile</a> <a href="#">writeProjectToFile</a>
target of relation	ComponentRealization <a href="#">Implementierung</a>
documentation	Has functions that can handle with the filesystem.

## Operation FileHandler::copyProject

owner	<a href="#">FileHandler</a>
parameter	name direction type multiplicity default <b>toCopy</b> in <a href="#">ProjectEntry</a> projectName in String return return <a href="#">ProjectEntry</a>
ownedMember	<b>projectName</b> <b>return</b> <b>toCopy</b>
documentation	<p>Copies a given project</p> <p>@param toCopy the project to copy.</p> <p>@param the project name for the new project.</p>

## Operation FileHandler::createNewProjectFile

owner	<a href="#">FileHandler</a>
parameter	name direction type multiplicity default <b>newProject</b> in <a href="#">ProjectEntry</a> return return void
ownedMember	<b>newProject</b> <b>return</b>
documentation	<p>Creates a new project configuration file.</p> <p>@param newProject the new project to be saved.</p>

## Operation FileHandler::deleteProjectFile

owner	<a href="#">FileHandler</a>
-------	-----------------------------

parameter	name direction type multiplicity default <b>newProject</b> in <a href="#">ProjectEntry</a> return return void
ownedMember	<b>newProject</b> return
documentation	<p>Deletes a project configuration file and thus the project.</p> <p>@param newProject the project to be deleted.</p>

#### Operation **FileHandler::FileHandler**

owner	<a href="#">FileHandler</a>
-------	-----------------------------

#### Operation **FileHandler::manipulateFileNameAndID**

owner	<a href="#">FileHandler</a>
parameter	name direction type multiplicity default <b>toManipulate</b> in <a href="#">ProjectEntry</a> return return void
ownedMember	<b>return toManipulate</b>

#### Property **FileHandler::PROJECTS\_PATH**

owner	<a href="#">FileHandler</a>
documentation	Determines the path to the directory where all projects of the program are stored.

#### Operation **FileHandler::scanProjectDirectory**

owner	<a href="#">FileHandler</a>
parameter	name direction type multiplicity default <b>return return List&lt;E&gt;ProjectEntry&gt;</b>
ownedMember	<b>return</b>
documentation	<p>Scans a the directory where all projects of the program are stored an creates for each project configuration file a ProjectEntry.</p> <p>@return a list of ProjectEntry objects.</p>

#### Operation **FileHandler::scanProjectFile**

owner	<a href="#">FileHandler</a>
parameter	name direction type multiplicity default <b>projectId in int return return ProjectModel</b>
ownedMember	<b>projectId return</b>
documentation	<p>Scans a project configuration file and sets the data (e.g. create picture containers).</p> <p>@param projectId the Id of the project configuration file.</p> <p>@return a model which contains all data for a project.</p>

### Operation **FileHandler::writeProjectToFile**

owner	<b>FileHandler</b>
parameter	name direction type multiplicity default <b>project</b> in <a href="#">ProjectModel</a> return return void
ownedMember	<b>project return</b>
documentation	<p>Writes the project to a project configuration file.</p> <p><b>@param project</b> the project to save.</p>

## Class Frequency3DPoint

diagram	<pre> classDiagram     class Frequency3DPoint {         +x:int         +y:int         +z:int         +frequency:int         +pictures:Picture[*]                  #getPictures():Picture[*]         #setPictures(in pictures:Picture[*]):void         #addPicture(in picture:Picture):void         &lt;&lt;constructor&gt;&gt; Frequency3DPoint(in x:int, in y:int, in z:int, in frequency:int)         #getX():int         #setX(in x:int):void         #getY():int         #setY(in y:int):void         #getZ():int         #setZ(in z:int):void         #getFrequency():int         #setFrequency(in frequency:int):void     }   </pre>
owner	<a href="#">reportmanagement</a>
ownedMember	<a href="#">addPicture</a> <a href="#">frequency</a> <a href="#">Frequency3DPoint</a> <a href="#">getFrequency</a> <a href="#">getPictures</a> <a href="#">getX</a> <a href="#">getY</a> <a href="#">getZ</a> <a href="#">pictures</a> <a href="#">setFrequency</a> <a href="#">setPictures</a> <a href="#">setX</a> <a href="#">setY</a> <a href="#">setZ</a> <a href="#">x</a> <a href="#">y</a> <a href="#">z</a>
target of relation	ComponentRealization <a href="#">Implementierung</a>
typedElements	Class <a href="#">Cluster3DModel</a> Property <a href="#">frequency3DPoints</a> Operation <a href="#">getFrequency3DPoints</a> Class <a href="#">SelectableCluster3DSphere</a> Property <a href="#">frequencypoint</a> Operation <a href="#">getFrequenc3DPoint</a> <a href="#">SelectableCluster3DSphere</a>
documentation	<p>Data capsule for Cluster3DModel.  It saves the coordinates of the frequency point and the frequency.  It also contains a list of the pictures that are represented with this Frequency3DPoint.  @author Kevin Zuber</p>

### Operation Frequency3DPoint::addPicture

owner	<a href="#">Frequency3DPoint</a>
parameter	name direction type multiplicity default <b>picture</b> in <a href="#">Picture</a> return void
ownedMember	<b>picture</b> return

### Property Frequency3DPoint::frequency

owner	<a href="#">Frequency3DPoint</a>
-------	----------------------------------

### Operation Frequency3DPoint::Frequency3DPoint

owner	<a href="#">Frequency3DPoint</a>
-------	----------------------------------

parameter	name direction type multiplicity default <b>x in int y in int z in int frequency in int</b>
ownedMember	<b>frequency x y z</b>

#### Operation Frequency3DPoint::getFrequency

owner	<b>Frequency3DPoint</b>
parameter	name direction type multiplicity default <b>return return int</b>
ownedMember	<b>return</b>

#### Operation Frequency3DPoint::getPictures

owner	<b>Frequency3DPoint</b>
parameter	name direction type multiplicity default <b>return return Picture *</b>
ownedMember	<b>return</b>

#### Operation Frequency3DPoint::getX

owner	<b>Frequency3DPoint</b>
parameter	name direction type multiplicity default <b>return return int</b>
ownedMember	<b>return</b>

#### Operation Frequency3DPoint::getY

owner	<b>Frequency3DPoint</b>
parameter	name direction type multiplicity default <b>return return int</b>
ownedMember	<b>return</b>

#### Operation Frequency3DPoint::getZ

owner	<b>Frequency3DPoint</b>
parameter	name direction type multiplicity default <b>return return int</b>
ownedMember	<b>return</b>

#### Property Frequency3DPoint::pictures

owner	<b>Frequency3DPoint</b>
-------	-------------------------

#### Operation Frequency3DPoint::setFrequency

owner	<b>Frequency3DPoint</b>
parameter	name direction type multiplicity default <b>frequency in int return return void</b>
ownedMember	<b>frequency return</b>

#### Operation **Frequency3DPoint::setPictures**

owner	<a href="#">Frequency3DPoint</a>
parameter	name direction type multiplicity default <b>pictures in Picture * return return void</b>
ownedMember	<b>pictures return</b>

#### Operation **Frequency3DPoint::setX**

owner	<a href="#">Frequency3DPoint</a>
parameter	name direction type multiplicity default <b>x in int return return void</b>
ownedMember	<b>return x</b>

#### Operation **Frequency3DPoint::setY**

owner	<a href="#">Frequency3DPoint</a>
parameter	name direction type multiplicity default <b>y in int return return void</b>
ownedMember	<b>return y</b>

#### Operation **Frequency3DPoint::setZ**

owner	<a href="#">Frequency3DPoint</a>
parameter	name direction type multiplicity default <b>z in int return return void</b>
ownedMember	<b>return z</b>

#### Property **Frequency3DPoint::x**

owner	<a href="#">Frequency3DPoint</a>
-------	----------------------------------

#### Property **Frequency3DPoint::y**

owner	<a href="#">Frequency3DPoint</a>
-------	----------------------------------

#### Property **Frequency3DPoint::z**

owner	<a href="#">Frequency3DPoint</a>
-------	----------------------------------

## Class Histogram2DConfig

diagram	<p>The diagram shows the class <b>Histogram2DConfig</b> with a dependency on <b>Histogram2DModel</b>. It has a final attribute <code>serialVersionUID:long=1L</code> and a constructor <code>Histogram2DConfig(in model:M)</code>.</p>
hierarchy	<p>The hierarchy diagram shows <b>Histogram2DConfig</b> inheriting from <b>AbstractReportCompilation&lt;M-&gt;M&gt;</b>.</p>
owner	<b>reportmanagement</b>
template parameters	name kind constrainingClassifier default <b>M</b> Class <b>Histogram2DModel</b>
ownedMember	<b>Histogram2DConfig serialVersionUID</b>
general	<b>AbstractReportCompilation&lt;M-&gt;M&gt;</b>
target of relation	ComponentRealization <b>Implementierung</b>
documentation	<p>This class represents the 2D Histogram configuration with all its necessary panels.</p> <p>@author David Kaufman</p> <p>@param &lt;M&gt;</p>

## Operation Histogram2DConfig::Histogram2DConfig

owner	<b>Histogram2DConfig</b>
parameter	name direction type multiplicity default <b>model in M</b>
ownedMember	<b>model</b>
documentation	<p>Constructor which initialized the report with all its panels</p> <p>@param histogram2dmodel the model which is used by the panels</p>

## Property Histogram2DConfig::serialVersionUID

owner	<b>Histogram2DConfig</b>
-------	--------------------------

## Class Histogram2DModel

diagram	<pre> <b>Histogram2DModel</b>   categories:Category[*]   &lt;&lt;constructor&gt;&gt; Histogram2DModel()   &lt;&lt;constructor&gt;&gt; Histogram2DModel(in pictureContainer:ArrayList&lt;E-&gt;PictureContainer&gt;, in xAxi   &lt;&lt;constructor&gt;&gt; getCategories():Category[*] </pre>
hierarchy	<pre> classDiagram     Observable &lt; -- AbstractModel     AbstractModel &lt; -- AbstractReportModel     AbstractReportModel &lt; -- AbstractSingleAxisModel     AbstractSingleAxisModel &lt; -- Histogram2DModel </pre>
owner	<a href="#">reportmanagement</a>
ownedMember	<a href="#">categories</a> <a href="#">getCategories</a> <a href="#">Histogram2DModel</a> <a href="#">Histogram2DModel</a>
general	<a href="#">AbstractSingleAxisModel</a>
target of relation	ComponentRealization <a href="#">Implementierung</a>
documentation	The model auf the Histogram2D which calculate the statistic categories and allocate the axes to the Data. @author Kevin Zuber

## Property Histogram2DModel::categories

owner	<a href="#">Histogram2DModel</a>
-------	----------------------------------

## Operation Histogram2DModel::getCategories

owner	<a href="#">Histogram2DModel</a>
parameter	name direction type multiplicity default <b>return</b> <b>return</b> <a href="#">Category</a> *
ownedMember	<b>return</b>

## Operation Histogram2DModel::Histogram2DModel

owner	<a href="#">Histogram2DModel</a>
-------	----------------------------------

## Operation Histogram2DModel::Histogram2DModel

owner	<a href="#">Histogram2DModel</a>
parameter	name direction type multiplicity default <b>pictureContainer</b> in <a href="#">ArrayList&lt;E-&gt;PictureContainer&gt;</a> <b>xAxis</b> in <a href="#">Axis</a>

ownedMember	<b>pictureContainer xAxis</b>
-------------	-------------------------------

## Class Histogram3DConfig

diagram	<p>The diagram shows the class <b>Histogram3DConfig</b> with a dependency on <b>Histogram3DModel</b>. It has a final attribute <code>serialVersionUID:long=1L</code> and a constructor <code>Histogram3DConfig(in model:M)</code>.</p>
hierarchy	<p>The hierarchy diagram shows <b>Histogram3DConfig</b> inheriting from <b>AbstractReportCompilation&lt;M-&gt;M&gt;</b>.</p>
owner	<a href="#">reportmanagement</a>
template parameters	name kind constrainingClassifier default <b>M</b> Class <b>Histogram3DModel</b>
ownedMember	<b>Histogram3DConfig serialVersionUID</b>
general	<a href="#">AbstractReportCompilation&lt;M-&gt;M&gt;</a>
target of relation	ComponentRealization <a href="#">Implementierung</a>
documentation	<p>This class represents the 3D Histogram configuration with all its necessary panels.</p> <p>@author David Kaufman</p> <p>@param &lt;M&gt;</p>

## Operation Histogram3DConfig::Histogram3DConfig

owner	<a href="#">Histogram3DConfig</a>
parameter	name direction type multiplicity default <b>model in M</b>
ownedMember	<b>model</b>
documentation	<p>Constructor which initialized the report with all its panels</p> <p>@param histogram3dmodel the model which is used by the panels</p>

## Property Histogram3DConfig::serialVersionUID

owner	<a href="#">Histogram3DConfig</a>
-------	-----------------------------------

## Class Histogram3DModel

diagram	<pre> <b>Histogram3DModel</b>  <b>categories:Category[*]</b>  &lt;&lt;constructor&gt;&gt; Histogram3DModel(<b>in</b> pictureContainer:<b>ArrayList&lt;E-&gt;PictureContainer&gt;</b>, <b>in</b> xAxis:<b>Axis</b>, <b>in</b> zAxis:<b>Axis</b>) &lt;&lt;constructor&gt;&gt; Histogram3DModel() <b>getCategories():Category[*]</b> </pre>
hierarchy	<pre> classDiagram     Observable &lt; -- AbstractModel     AbstractModel &lt; -- AbstractReportModel     AbstractReportModel &lt; -- AbstractSingleAxisModel     AbstractSingleAxisModel &lt; -- AbstractDoubleAxesModel     AbstractDoubleAxesModel &lt; -- Histogram3DModel </pre>
owner	<a href="#">reportmanagement</a>
ownedMember	<a href="#">categories getCategories Histogram3DModel Histogram3DModel</a>
general	<a href="#">AbstractDoubleAxesModel</a>
target of relation	ComponentRealization <a href="#">Implementierung</a>
documentation	The model auf the Histogram3D which calculate the statistic categories and allocate the axes to the Data. @author Kevin Zuber

## Property Histogram3DModel::categories

owner	<a href="#">Histogram3DModel</a>
-------	----------------------------------

## Operation Histogram3DModel::getCategories

owner	<a href="#">Histogram3DModel</a>
parameter	name direction type multiplicity default <b>return return Category *</b>
ownedMember	<b>return</b>

## Operation Histogram3DModel::Histogram3DModel

owner	<a href="#">Histogram3DModel</a>
parameter	name direction type multiplicity default <b>pictureContainer in ArrayList&lt;E-&gt;PictureContainer&gt; xAxis in Axis zAxis in Axis</b>
ownedMember	<b>pictureContainer xAxis zAxis</b>

Operation **Histogram3DModel::Histogram3DModel**

owner	<a href="#">Histogram3DModel</a>
-------	----------------------------------

UML documentation generated by [UModel](#) UML Editor <http://www.altova.com/umodel>

12/17/09 13:54:45

## Component Implementierung

owner	Component View
source of relation	ComponentRealization <a href="#">PictureSetContentListClickOnController</a> <a href="#">JJPEGHelper</a> <a href="#">ProjectSaveController</a> <a href="#">JAbstract3DDiagram</a> <a href="#">JAbstract2DDiagram</a> <a href="#">PictureSetListCreateController</a> <a href="#">JExifTag</a> <a href="#">JProjectManagement</a> <a href="#">DemoJFileChooser</a> <a href="#">JDiagramButtons3D</a> <a href="#">ProjectEditDescriptionController</a> <a href="#">JDiagramButtonsPlain</a> <a href="#">ProjectDeleteController</a> <a href="#">ReportDeleteController</a> <a href="#">Klient</a> <a href="#">FileHandler</a> <a href="#">JProjectView</a> <a href="#">JRelPathHelper</a> <a href="#">JMyFileFilter</a> <a href="#">JPathHelper</a> <a href="#">JFileHelper</a> <a href="#">JAbstractView</a> <a href="#">ProjectCopyController</a> <a href="#">JCluster3D</a> <a href="#">PictureSetListDragController</a> <a href="#">PictureSetContentListAddController</a> <a href="#">SelectableCluster3DSphere</a> <a href="#">JHistogram2D</a> <a href="#">DiagramTypeSelectController</a> <a href="#">ReportOpenController</a> <a href="#">ReportAddPictureSetController</a> <a href="#">ReportPictureSetRemoveController</a> <a href="#">JTiffHeader</a> <a href="#">PictureSetListDeleteController</a> <a href="#">JTableDiagram</a> <a href="#">JHistogram3D</a> <a href="#">JffdData</a> <a href="#">JDataFormatHelper</a> <a href="#">PictureListClickOnController</a> <a href="#">TextModel</a> <a href="#">ProjectSwitchController</a> <a href="#">PictureThumbnailAdapter</a> <a href="#">View3DCycleController</a> <a href="#">Programm</a> <a href="#">ReportCloseController</a> <a href="#">Perspectives</a> <a href="#">PictureSetListCopyController</a> <a href="#">ExifAdapter</a> <a href="#">Histogram2DModel</a> <a href="#">TableModel</a> <a href="#">View3DClickController</a> <a href="#">PictureParameter</a> <a href="#">Histogram3DModel</a> <a href="#">BoxplotModel</a> <a href="#">ReportEditController</a> <a href="#">Cluster3DModel</a> <a href="#">Frequency3DPoint</a> <a href="#">Category</a> <a href="#">DiagramResetViewController</a> <a href="#">Boxplot</a> <a href="#">Axis</a> <a href="#">DiagramExportAsJPEGController</a> <a href="#">Bar</a> <a href="#">Pointplotter</a> <a href="#">AbstractTripleAxesModel</a> <a href="#">AbstractSingleAxisModel</a> <a href="#">ProjectModel</a> <a href="#">AbstractReportModel</a> <a href="#">AbstractDoubleAxesModel</a> <a href="#">PointPlotterWindow</a> <a href="#">Object2XML</a> <a href="#">XML2Object</a> <a href="#">WizardCloseController</a> <a href="#">ProjectEntry</a> <a href="#">ProjectManagementModel</a> <a href="#">DemoThumbnailQuality</a> <a href="#">MyProjectListCellRenderer</a> <a href="#">JDiagramButtons2D</a> <a href="#">ProjectOpenController</a> <a href="#">StringChecker</a> <a href="#">ExifParameter</a> <a href="#">ProjectCreateController</a> <a href="#">PictureListCheckBoxController</a> <a href="#">JAbstract3DView</a> <a href="#">JExif</a> <a href="#">PictureSetListClickOnController</a> <a href="#">JTextDiagram</a> <a href="#">ProjectEditNameController</a> <a href="#">ReportAddExifKeywordController</a> <a href="#">PictureSetContentListRefreshController</a> <a href="#">ReportRemoveExifKeywordController</a> <a href="#">PictureSetContentListDeleteController</a> <a href="#">JBoxplot</a> <a href="#">JAbstractDiagram</a> <a href="#">PictureSet</a> <a href="#">PictureContainer</a> <a href="#">WizardPreviousPanelController</a> <a href="#">AbstractModel</a> <a href="#">Picture</a> <a href="#">Directory</a> <a href="#">WizardNextPanelController</a> <a href="#">DemoThumbnailSpeed</a> <a href="#">DemoTooltip</a> <a href="#">ReportSaveController</a> <a href="#">PictureSetContentListDropController</a> <a href="#">Jffd</a> <a href="#">JFileSaver</a> <a href="#">ReportCreateController</a> <a href="#">WilcoxonTestType</a> <a href="#">DemoJFileChooser</a> <a href="#">DemoJFileChooser</a> <a href="#">Directory</a> <a href="#">JFileHelper</a> <a href="#">JMyFileFilter</a> <a href="#">JPathHelper</a> <a href="#">JRelPathHelper</a> <a href="#">Main</a> <a href="#">DiagramCloseController</a> <a href="#">AbstractController</a> <a href="#">Picture</a> <a href="#">PictureContainer</a> <a href="#">PictureSet</a> <a href="#">JffdData</a> <a href="#">JJPEGHelper</a> <a href="#">JTiffHeader</a> <a href="#">JDataFormatHelper</a> <a href="#">MyPictureListCellRenderer</a> <a href="#">MyPictureSetListCellRenderer</a> <a href="#">MyPictureSetContentListCellRenderer</a> <a href="#">MyPictureListCellRenderer</a> <a href="#">MyReportListCellRenderer</a> <a href="#">MyTableCellRenderer</a> <a href="#">AbstractReportCompilation</a> <a href="#">BoxplotConfig</a> <a href="#">Cluster3DConfig</a> <a href="#">Histogram2DConfig</a> <a href="#">Histogram3DConfig</a> <a href="#">JAbstractReportUtil</a> <a href="#">JAbstractSinglePanel</a> <a href="#">JDiagramType</a> <a href="#">ComplexCellRenderer</a> <a href="#">JParameters</a> <a href="#">JPictureSetExif</a> <a href="#">JReportConfig</a> <a href="#">JReportWizard</a> <a href="#">JWilcoxon</a> <a href="#">ReportHelper</a> <a href="#">TableConfig</a> <a href="#">ExifAdapter</a> <a href="#">JExif</a> <a href="#">JExifTag</a> <a href="#">Jffd</a>

## Class JAbstract2DDiagram

diagram	<pre> classDiagram     JAbstract2DDiagram "1" -- "1" M&gt;AbstractModel     JAbstract2DDiagram {         &lt;&lt;final&gt;&gt; serialVersionUID:long=1L         &lt;&lt;constructor&gt;&gt; JAbstract2DDiagram(in model:M)         &lt;&lt;annotation&gt;&gt; preinitialize():void     }     note over preinitialize()         &lt;&lt;annotations&gt;&gt; modifiers = Override     end   </pre>
hierarchy	<pre> classDiagram     JAbstract3DView&lt;M-&gt;M&gt; --&gt; JAbstract2DDiagram   </pre>
owner	<a href="#">diagrams</a>
template parameters	name kind constrainingClassifier default <b>M Class AbstractModel</b>
ownedMember	<a href="#">JAbstract2DDiagram preinitialize serialVersionUID</a>
general	<a href="#">JAbstract3DView&lt;M-&gt;M&gt;</a>
target of relation	ComponentRealization <a href="#">Implementierung</a>
documentation	<p>This class represents a 2D Diagram. It implements the 2D specific preinitialization routine so that all child classes use the specified configuration.</p> <p>@author David Kaufman</p> <p>@param &lt;M&gt;</p>

## Operation JAbstract2DDiagram::JAbstract2DDiagram

owner	<a href="#">JAbstract2DDiagram</a>
parameter	name direction type multiplicity default <b>model in M</b>
ownedMember	<b>model</b>
documentation	<p>Constructor</p> <p>@param abstractModel the model from which the drawing information is taken</p>

## Operation JAbstract2DDiagram::preinitialize

owner	<a href="#">JAbstract2DDiagram</a>
parameter	name direction type multiplicity default <b>return return void</b>
ownedMember	<b>return</b>

## Property JAbstract2DDiagram::serialVersionUID

owner	<a href="#">JAbstract2DDiagram</a>
-------	------------------------------------



## Class JAbstract3DDiagram

diagram	<pre> classDiagram     class JAbstract3DDiagram {         &lt;&lt;final&gt;&gt; serialVersionUID:long=1L         &lt;&lt;constructor&gt;&gt; JAbstract3DDiagram(in model:M)         &lt;&lt;annotations&gt;&gt; preinitialize():void     }     JAbstract3DDiagram "1" --&gt; "1" &lt;M&gt;AbstractModel     note over JAbstract3DDiagram: preinitialize() (Operation)     note over JAbstract3DDiagram: &lt;&lt;annotations&gt;&gt; modifiers = Override   </pre>
hierarchy	<pre> classDiagram     class JAbstract3DView&lt;M-M&gt;     class JAbstract3DDiagram     JAbstract3DView&lt;M-M&gt; &lt; -- JAbstract3DDiagram   </pre>
owner	<a href="#">diagrams</a>
template parameters	name kind constrainingClassifier default <b>M</b> Class <a href="#">AbstractModel</a>
ownedMember	<a href="#">JAbstract3DDiagram preinitialize serialVersionUID</a>
general	<a href="#">JAbstract3DView&lt;M-M&gt;</a>
target of relation	ComponentRealization <a href="#">Implementierung</a>
documentation	<p>This class represents a 3D Diagram. It implements the 3D specific preinitialization routine so that all child classes use the specified configuration.</p> <p>@author David Kaufman</p> <p>@param &lt;M&gt;</p>

## Operation JAbstract3DDiagram::JAbstract3DDiagram

owner	<a href="#">JAbstract3DDiagram</a>
parameter	name direction type multiplicity default <b>model</b> in <b>M</b>
ownedMember	<b>model</b>
documentation	<p>Constructor</p> <p>@param abstractModel the model from which the drawing information is taken</p>

## Operation JAbstract3DDiagram::preinitialize

owner	<a href="#">JAbstract3DDiagram</a>
parameter	name direction type multiplicity default <b>return return void</b>
ownedMember	<b>return</b>

## Property JAbstract3DDiagram::serialVersionUID

owner	<a href="#">JAbstract3DDiagram</a>
-------	------------------------------------



## Class JAbstract3DView

diagram	<p>The diagram shows the JAbstract3DView class with its attributes and operations. It includes a note indicating it is a M&gt;AbstractModel relationship.</p> <pre> classDiagram     class JAbstract3DView {         &lt;&lt;final&gt;&gt; serialVersionUID:long=1L         canvas3D:Canvas3D         simpleU:SimpleUniverse         objRoot:BranchGroup         &lt;&lt;final&gt;&gt; AXISSIZE:double=10.0         &lt;&lt;final&gt;&gt; GRIDDENSITYFACTOR:double=1         &lt;&lt;final&gt;&gt; GEODETAIL:int=10         numberofAxes:int=3         &lt;&lt;final&gt;&gt; NUMBEROFSSEGMENTS:int=10         textautorotate:boolean=true          &lt;&lt;constructor&gt;&gt; JAbstract3DView(in model:M)         preinitialize():void         generateContent():void         createCube(in position:Vector3d, in scale:Vector3d, in material:Appearance):void         createSphere(in position:Vector3d, in scale:Vector3d, in material:Appearance):void         createText(in position:Vector3d, in scale:Vector3d, in material:Appearance, in text:String):void         createAxis():void         createGrid():void         postinitialize():void         createLabels(in xAxis:String, in zAxis:String, in yAxis:String):void         generateSegmentDescription(in minValue:Object, in maxValue:Object):String[]         setSegmentDescription(in xAxis:String[], in zAxis:String[], in yAxis:String[]):void         setCurrentDescription(in pic:Picture):void         &lt;&lt;annotation&gt;&gt; getDiagramScreenshot():BufferedImage         createBackground(in color:Color3f):void         addLights():void         basicMaterial(in r:float, in g:float, in b:float):Appearance         changeCameraPosition(in x:float, in y:float, in z:float):void         createTransformGroup(in position:Vector3d, in scale:Vector3d):TransformGroup         setCameraPerspective(in perspEnum:Perspectives):void         changeCamtoFaceYXPlane():void         setCurrentPicture(in pic:Picture):void     } </pre>
hierarchy	<p>The hierarchy diagram shows JAbstract3DView inheriting from JAbstractDiagram&lt;M-M&gt;.</p> <pre> graph TD     JAbstractDiagram["JAbstractDiagram&lt;M-&gt;M&gt;"] --&gt; JAbstract3DView["JAbstract3DView"] </pre>
owner	<a href="#">diagrams</a>
template parameters	name kind constrainingClassifier default <b>M Class AbstractModel</b>
ownedMember	<a href="#">addLights</a> <a href="#">AXISSIZE</a> <a href="#">basicMaterial</a> <a href="#">canvas3D</a> <a href="#">changeCameraPosition</a> <a href="#">changeCamtoFaceYXPlane</a> <a href="#">createAxis</a> <a href="#">createBackground</a> <a href="#">createCube</a> <a href="#">createGrid</a> <a href="#">createLabels</a> <a href="#">createSphere</a> <a href="#">createText</a> <a href="#">createTransformGroup</a> <a href="#">generateContent</a> <a href="#">generateSegmentDescription</a> <a href="#">GEODETAIL</a> <a href="#">getDiagramScreenshot</a> <a href="#">GRIDDENSITYFACTOR</a> <a href="#">JAbstract3DView</a> <a href="#">numberofAxes</a> <a href="#">NUMBEROFSSEGMENTS</a> <a href="#">objRoot</a> <a href="#">postinitialize</a> <a href="#">preinitialize</a> <a href="#">serialVersionUID</a> <a href="#">setCameraPerspective</a> <a href="#">setCurrentDescription</a> <a href="#">setCurrentPicture</a> <a href="#">setSegmentDescription</a> <a href="#">simpleU</a> <a href="#">textautorotate</a>
general	<a href="#">JAbstractDiagram&lt;M-&gt;M&gt;</a>
target of relation	ComponentRealization <a href="#">Implementierung</a>
documentation	<p>This class manages all java 3d interactions and offers methods than simplify the utilization of the java 3d functions significantly</p> <p>@author David Kaufman</p>

	@param <M> the model which is to be drawn
--	---

#### Operation **JAbstract3DView::addLights**

owner	<a href="#">JAbstract3DView</a>
parameter	name direction type multiplicity default <b>return return void</b>
ownedMember	<b>return</b>
documentation	Creates and assigns a default light setup to the root BranchGroup

#### Property **JAbstract3DView::AXISSIZE**

owner	<a href="#">JAbstract3DView</a>
documentation	The size of the various axes

#### Operation **JAbstract3DView::basicMaterial**

owner	<a href="#">JAbstract3DView</a>
parameter	name direction type multiplicity default <b>r in float g in float b in float return return Appearance</b>
ownedMember	<b>b g r return</b>
documentation	Creates a basicMaterial with the specified color  @param r the amount of red of the material @param g the amount of green of the material @param b the amount of blue of the material @return

#### Property **JAbstract3DView::canvas3D**

owner	<a href="#">JAbstract3DView</a>
documentation	The canvas in which the 3D elements are painted to

#### Operation **JAbstract3DView::changeCameraPosition**

owner	<a href="#">JAbstract3DView</a>
parameter	name direction type multiplicity default <b>x in float y in float z in float return return void</b>
ownedMember	<b>return x y z</b>
documentation	Changes the camera position to the specified location  @param x the x coordinate @param y the y coordinate @param z the z coordinate

#### Operation **JAbstract3DView::changeCamtoFaceYXPlane**

owner	<b>JAbstract3DView</b>
parameter	name direction type multiplicity default <b>return return void</b>
ownedMember	<b>return</b>

#### Operation **JAbstract3DView::createAxis**

owner	<b>JAbstract3DView</b>
parameter	name direction type multiplicity default <b>return return void</b>
ownedMember	<b>return</b>
documentation	Creates the axes which are automatically placed into the root BranchGroup. To change the number of axis you must change the numberOfAxes variable before calling this function.

#### Operation **JAbstract3DView::createBackground**

owner	<b>JAbstract3DView</b>
parameter	name direction type multiplicity default <b>color in Color3f return return void</b>
ownedMember	<b>color return</b>
documentation	Creates and assigns the specified background color to the root BranchGroup  @param color the background color

#### Operation **JAbstract3DView::createCube**

owner	<b>JAbstract3DView</b>
parameter	name direction type multiplicity default <b>position in Vector3d scale in Vector3d material in Appearance return void</b>
ownedMember	<b>material position return scale</b>
documentation	Creates a Cube  @param position the position of the cube @param scale the scale value of the cube @param material the material of the cube

#### Operation **JAbstract3DView::createGrid**

owner	<b>JAbstract3DView</b>
parameter	name direction type multiplicity default <b>return return void</b>
ownedMember	<b>return</b>
documentation	Creates a grid which is automatically placed into the root BranchGroup

--	--

#### Operation **JAbstract3DView::createLabels**

owner	<b>JAbstract3DView</b>
parameter	name direction type multiplicity default <b>xAxis in String zAxis in String yAxis in String return return void</b>
ownedMember	<b>return xAxis yAxis zAxis</b>
documentation	Specifies the labels that are displayed next to each axis @param xAxis The label of the xAis @param zAxis The label of the zAis @param yAis The label of the yAis

#### Operation **JAbstract3DView::createSphere**

owner	<b>JAbstract3DView</b>
parameter	name direction type multiplicity default <b>position in Vector3d scale in Vector3d material in Appearance return return void</b>
ownedMember	<b>material position return scale</b>
documentation	Creates a Sphere  @param position the position of the sphere @param scale the scale value of the sphere @param material the material of the sphere

#### Operation **JAbstract3DView::createText**

owner	<b>JAbstract3DView</b>
parameter	name direction type multiplicity default <b>position in Vector3d scale in Vector3d material in Appearance text in String return return void</b>
ownedMember	<b>material position return scale text</b>
documentation	Creates Text  @param position the position of the text @param scale the scale of the text @param material the material of the text @param text the text itself

#### Operation **JAbstract3DView::createTransformGroup**

owner	<b>JAbstract3DView</b>
parameter	name direction type multiplicity default <b>position in Vector3d scale in Vector3d return return TransformGroup</b>
ownedMember	<b>position return scale</b>
documentation	Creates a TransformGroup with the specified positon and scale  @param position the position of the TransformGroup @param scale the scale of the TransformGroup @return

--	--

#### Operation **JAbstract3DView::generateContent**

owner	<b>JAbstract3DView</b>
parameter	name direction type multiplicity default <b>return return void</b>
ownedMember	<b>return</b>
documentation	Specifies the contents which are to be drawn in the canvas3D

#### Operation **JAbstract3DView::generateSegmentDescription**

owner	<b>JAbstract3DView</b>
parameter	name direction type multiplicity default <b>minValue in Object maxValue in Object return return String *</b>
ownedMember	<b>maxValue minValue return</b>
documentation	Generates a string array which contains the segment description of one axis. The array will have the length of NUMBEROFS SEGMENTS. @param minValue the minimum value which will be placed at the origin of the axis @param maxValue the maximum value of the axis

#### Property **JAbstract3DView::GEODETAIL**

owner	<b>JAbstract3DView</b>
documentation	Specifies at which detail level the geometry is to be drawn

#### Operation **JAbstract3DView::getDiagramScreenshot**

owner	<b>JAbstract3DView</b>
parameter	name direction type multiplicity default <b>return return BufferedImage</b>
ownedMember	<b>return</b>

#### Property **JAbstract3DView::GRIDDENSITYFACTOR**

owner	<b>JAbstract3DView</b>
documentation	Specifies how dense the grid is to be drawn, a high number indicates a high grid density

#### Operation **JAbstract3DView::JAbstract3DView**

owner	<b>JAbstract3DView</b>
parameter	name direction type multiplicity default <b>model in M</b>
ownedMember	<b>model</b>
documentation	

on	Constructor initialized the canvas3D  @param abstractModel the model from which the drawing information is taken
----	---

#### Property **JAbstract3DView::numberOfAxes**

owner	<a href="#">JAbstract3DView</a>
documentati on	Specifies the number of axes used

#### Property **JAbstract3DView::NUMBEROFSEGMENTS**

owner	<a href="#">JAbstract3DView</a>
documentati on	Specifies the number of segments for one axis

#### Property **JAbstract3DView::objRoot**

owner	<a href="#">JAbstract3DView</a>
documentati on	The BranchGroup serves as a pointer to the root of a scene graph branch. This is where all of the 3D Components are stored

#### Operation **JAbstract3DView::postinitialize**

owner	<a href="#">JAbstract3DView</a>
parameter	name direction type multiplicity default <b>return return void</b>
ownedMem ber	<b>return</b>

#### Operation **JAbstract3DView::preinitialize**

owner	<a href="#">JAbstract3DView</a>
parameter	name direction type multiplicity default <b>return return void</b>
ownedMem ber	<b>return</b>
documentati on	Specifies the preinitialization routine which is executed before every scene draw

#### Property **JAbstract3DView::serialVersionUID**

owner	<a href="#">JAbstract3DView</a>
-------	---------------------------------

#### Operation **JAbstract3DView::setCameraPerspective**

owner	<a href="#">JAbstract3DView</a>
parameter	name direction type multiplicity default <b>perspEnum in Perspectives return return void</b>
ownedMem ber	<b>perspEnum return</b>

ber	
documentati on	<p>Sets the camera perspective to one of the predefined perspectives in the perspective enumeration</p> <p>@param perspEnum the perspective to change to</p>

#### Operation **JAbstract3DView::setCurrentDescription**

owner	<b>JAbstract3DView</b>
parameter	name direction type multiplicity default <b>pic</b> in <a href="#">Picture</a> return return void
ownedMem ber	<b>pic return</b>
documentati on	<p>Sets the current picture which is displayed outside of the 3D view with the specified exif parameters</p> <p>@param pic the picture which will be displayed outside of the 3D view</p>

#### Operation **JAbstract3DView::setCurrentPicture**

owner	<b>JAbstract3DView</b>
parameter	name direction type multiplicity default <b>pic</b> in <a href="#">Picture</a> return return void
ownedMem ber	<b>pic return</b>
documentati on	<p>Sets the current picture which is displayed outside of the 3D View</p> <p>@param pic</p>

#### Operation **JAbstract3DView::setSegmentDescription**

owner	<b>JAbstract3DView</b>
parameter	name direction type multiplicity default <b>xAxis</b> in <a href="#">String</a> * <b>zAxis</b> in <a href="#">String</a> * <b>yAxis</b> in <a href="#">String</a> * return return void
ownedMem ber	<b>return xAxis yAxis zAxis</b>
documentati on	<p>Defines the description of each segment of an axis. Note that the string arrays you pass in here have length of NUMBEROFS SEGMENTS</p> <p>@param xAxis the string array which represent the xAxis</p> <p>@param zAxis the string array which represent the zAxis</p> <p>@param yAxis the string array which represent the yAxis</p>

#### Property **JAbstract3DView::simpleU**

owner	<b>JAbstract3DView</b>
documentati on	This SimpleUniverse is a minimal user environment to quickly and easily get a Java 3D program up and running.

## Property **JAbstract3DView::textautorotate**

owner	<a href="#">JAbstract3DView</a>
documentation	Specifies if the text in the current view should point to the camera

## Class JAbstractDiagram

diagram	<p><b>JAbstractDiagram</b></p> <ul style="list-style-type: none"> <li>&lt;&lt;final&gt;&gt; serialVersionUID:long=1L</li> <li>registeredButtons:JComponent</li> </ul> <p><b>Operations:</b></p> <ul style="list-style-type: none"> <li>&lt;&lt;constructor&gt;&gt; JAbstractDiagram(in model:M)</li> <li>getDiagramScreenshot():BufferedImage</li> <li>&lt;&lt;annotations&gt;&gt; update(in model:Observable, in argument:Object):void</li> </ul>
hierarchy	<pre>     └── JAbstractView&lt;M-&gt;M&gt;                   +-- JAbstractDiagram   </pre>
owner	<a href="#">diagrams</a>
template parameters	name kind constrainingClassifier default <b>M Class AbstractModel</b>
ownedMember	<a href="#">getDiagramScreenshot</a> <a href="#">JAbstractDiagram</a> <a href="#">registeredButtons</a> <a href="#">serialVersionUID</a> <a href="#">update</a>
general	<a href="#">JAbstractView&lt;M-&gt;M&gt;</a>
target of relation	ComponentRealization <a href="#">Implementierung</a>
documentation	<p>This class specifies the main functionality of every diagram in knipsX</p> <p>@author David Kaufman</p> <p>@param &lt;M&gt;</p>

## Operation JAbstractDiagram::getDiagramScreenshot

owner	<a href="#">JAbstractDiagram</a>
parameter	name direction type multiplicity default <b>return return BufferedImage</b>
ownedMember	<b>return</b>
documentation	<p>This method has to be implemented by every diagram which returns a BufferedImage of the current diagram which can later be exported as an image.</p> <p>@return BufferedImage containing the current view</p>

## Operation JAbstractDiagram::JAbstractDiagram

owner	<a href="#">JAbstractDiagram</a>
parameter	name direction type multiplicity default <b>model in M</b>
ownedMember	<b>model</b>

### Property **JAbstractDiagram::registeredButtons**

owner	<a href="#">JAbstractDiagram</a>
documentation	This JComponent contains the button set of the specific diagram view. Defining this in the abstract class allows greater flexibility, because for example all 3DDiagram use the same button sets. If any diagram view wants to have their individual button set this variable can be overwritten.

### Property **JAbstractDiagram::serialVersionUID**

owner	<a href="#">JAbstractDiagram</a>
-------	----------------------------------

### Operation **JAbstractDiagram::update**

owner	<a href="#">JAbstractDiagram</a>
parameter	name direction type multiplicity default <b>model in Observable argument in Object return return void</b>
ownedMember	<b>argument model return</b>

## Class JAbstractReportUtil

diagram	<p>The diagram shows the class <b>JAbstractReportUtil</b> with the following details:</p> <ul style="list-style-type: none"> <li><b>Attributes:</b> <code>&lt;&lt;final&gt;&gt; serialVersionUID:long=1L</code></li> <li><b>Operations:</b> <code>&lt;&lt;constructor&gt;&gt; JAbstractReportUtil(in model:M)</code>, <code>setReportType(in jAbstractReport:AbstractReportCompilation&lt;M-&gt;?&gt;):void</code></li> </ul>
hierarchy	<pre> classDiagram     JAbstractView &lt; -- JAbstractReportUtil   </pre> <p>The hierarchy diagram shows <b>JAbstractReportUtil</b> inheriting from <b>JAbstractView&lt;M-&gt;M&gt;</b>.</p>
owner	<b>reportmanagement</b>
template parameters	name kind constrainingClassifier default <b>M</b> Class <b>AbstractReportModel</b> V Class <b>AbstractReportCompilation&lt;M-&gt;M&gt;</b>
ownedMember	<b>JAbstractReportUtil serialVersionUID setReportType</b>
general	<b>JAbstractView&lt;M-&gt;M&gt;</b>
target of relation	ComponentRealization <b>Implementierung</b>
documentation	<p>This class is the parent class of every configuration utility.</p> <p>@author David Kaufman</p> <p>@param &lt;M&gt; @param &lt;V&gt;</p>

## Operation JAbstractReportUtil::JAbstractReportUtil

owner	<b>JAbstractReportUtil</b>
parameter	name direction type multiplicity default <b>model in M</b>
ownedMember	<b>model</b>

## Property JAbstractReportUtil::serialVersionUID

owner	<b>JAbstractReportUtil</b>
-------	----------------------------

## Operation JAbstractReportUtil::setReportType

owner	<b>JAbstractReportUtil</b>
parameter	name direction type multiplicity default <b>jAbstractReport in AbstractReportCompilation&lt;M-&gt;?&gt;</b> return return void
ownedMember	<b>jAbstractReport return</b>
documentation	<p>Each configuration utility has to implement this method which registeres a new report type to the current configuration utility</p> <p>@param jAbstractReport the new report type</p>

--	--

## Class JAbstractSinglePanel

diagram	<pre> <b>JAbstractSinglePanel</b> +--&lt;&gt; serialVersionUID:long=1L +--&lt;&gt; title:String +--&lt;&gt; icon:Icon +--&lt;&gt; tip:String  +--&lt;&gt; getTitle():String +--&lt;&gt; setTitle(in title:String):void +--&lt;&gt; getIcon():Icon +--&lt;&gt; setIcon(in icon:Icon):void +--&lt;&gt; getTip():String +--&lt;&gt; setTip(in tip:String):void </pre>
hierarchy	<pre> classDiagram     class ImageObserver     class MenuContainer     class Serializable     class Component     class Container     class JComponent     class JAbstractSinglePanel     class JDigramType     class JParameters     class JPicuterSetExif     class JWilcoxon      ImageObserver &lt; -- JAbstractSinglePanel     MenuContainer &lt; -- JAbstractSinglePanel     Serializable &lt; -- JAbstractSinglePanel     JAbstractSinglePanel -- &gt; Component     JAbstractSinglePanel -- &gt; Container     JAbstractSinglePanel -- &gt; JComponent     JAbstractSinglePanel -- &gt; JDigramType     JAbstractSinglePanel -- &gt; JParameters     JAbstractSinglePanel -- &gt; JPicuterSetExif     JAbstractSinglePanel -- &gt; JWilcoxon </pre>
owner	<a href="#">reportmanagement</a>
ownedMember	<a href="#">getIcon</a> <a href="#">getTip</a> <a href="#">getTitle</a> <a href="#">icon</a> <a href="#">serialVersionUID</a> <a href="#">setIcon</a> <a href="#">setTip</a> <a href="#">setTitle</a> <a href="#">tip</a> <a href="#">title</a>
general	<a href="#">JComponent</a>
specific	<a href="#">JDigramType</a> <a href="#">JParameters</a> <a href="#">JPicuterSetExif</a> <a href="#">JWilcoxon</a>
target of relation	ComponentRealization <a href="#">Implementierung</a>
typedElements	Class <a href="#">AbstractReportCompilation</a> Operation <a href="#">addPanel</a>
documentation	<p>This class represents a single panel in a report configuration.</p> <p>@author David Kaufman</p>

## Operation JAbstractSinglePanel::getIcon

owner	<a href="#">JAbstractSinglePanel</a>
parameter	name direction type multiplicity default <b>return</b> <b>return</b> <b>Icon</b>
ownedMember	<b>return</b>

**Operation JAbstractSinglePanel::getTip**

owner	<a href="#">JAbstractSinglePanel</a>
parameter	name direction type multiplicity default <b>return return String</b>
ownedMember	<b>return</b>

**Operation JAbstractSinglePanel::getTitle**

owner	<a href="#">JAbstractSinglePanel</a>
parameter	name direction type multiplicity default <b>return return String</b>
ownedMember	<b>return</b>

**Property JAbstractSinglePanel::icon**

owner	<a href="#">JAbstractSinglePanel</a>
documentation	The icon which is registered with this panel.

**Property JAbstractSinglePanel::serialVersionUID**

owner	<a href="#">JAbstractSinglePanel</a>
-------	--------------------------------------

**Operation JAbstractSinglePanel::setIcon**

owner	<a href="#">JAbstractSinglePanel</a>
parameter	name direction type multiplicity default <b>icon in Icon return return void</b>
ownedMember	<b>icon return</b>

**Operation JAbstractSinglePanel::setTip**

owner	<a href="#">JAbstractSinglePanel</a>
parameter	name direction type multiplicity default <b>tip in String return return void</b>
ownedMember	<b>return tip</b>

**Operation JAbstractSinglePanel::setTitle**

owner	<a href="#">JAbstractSinglePanel</a>
parameter	name direction type multiplicity default <b>title in String return return void</b>
ownedMember	<b>return title</b>

**Property JAbstractSinglePanel::tip**

owner	<a href="#">JAbstractSinglePanel</a>
documentation	The tooltip which is registered with this panel.

**Property `JAbstractSinglePanel::title`**

owner	<a href="#">JAbstractSinglePanel</a>
documentation	The title which is registered with this panel.

## Class JAbstractView

diagram	<p>The diagram shows the class <b>JAbstractView</b> with the following details:</p> <ul style="list-style-type: none"> <li><b>Attributes:</b> <code>&lt;&lt;final&gt;&gt; serialVersionUID:long=-5981384605515636896L</code>, <code>model:M</code></li> <li><b>Operations:</b> <code>&lt;&lt;constructor&gt;&gt; JAbstractView(in model:M)</code>, <code>update(in model:Observable, in argument:Object):void</code>, <code>&lt;&lt;annotations&gt;&gt; dispose():void</code></li> </ul>
hierarchy	<pre> classDiagram     class JAbstractView {         &lt;&lt;final&gt;&gt; serialVersionUID:long=-5981384605515636896L         model:M     }     class AbstractModel {         &lt;&lt;operations&gt;&gt;     }     JAbstractView "1" -- "1" &lt;&lt;AbstractModel&gt;&gt; : M&gt;&gt;AbstractModel     JAbstractView "1" -- "1" &lt;&lt;operations&gt;&gt; dispose() (Operation)     &lt;&lt;annotations&gt;&gt; modifiers = Override     JAbstractView "1" -- "1" &lt;&lt;operations&gt;&gt; &lt;&lt;final&gt;&gt; serialVersionUID:long=-5981384605515636896L     JAbstractView "1" -- "1" &lt;&lt;operations&gt;&gt; model:M     JAbstractView "1" -- "1" &lt;&lt;operations&gt;&gt; &lt;&lt;constructor&gt;&gt; JAbstractView(in model:M)     JAbstractView "1" -- "1" &lt;&lt;operations&gt;&gt; update(in model:Observable, in argument:Object):void     JAbstractView "1" -- "1" &lt;&lt;operations&gt;&gt; &lt;&lt;final&gt;&gt; dispose():void     class Component     class Container     class Accessible     class Window     class MenuContainer     class Frame     class WindowConstants     class Accessible     class RootPaneContainer     class JFrame     class DemoJFileChooser     class Observer      Component &lt; -- Container     Container &lt; -- Accessible     Window &lt; -- MenuContainer     Window &lt; -- Frame     Frame &lt; -- WindowConstants     Frame &lt; -- Accessible     Frame &lt; -- RootPaneContainer     JFrame &lt; -- DemoJFileChooser     JFrame &lt; -- Observer     JAbstractView &lt; -- DemoJFileChooser     JAbstractView &lt; -- Observer   </pre>
owner	<a href="#">view</a>
template parameters	name kind constrainingClassifier default <b>M</b> Class <a href="#">AbstractModel</a>
ownedMember	<a href="#">dispose</a> <a href="#">JAbstractView</a> <a href="#">model</a> <a href="#">serialVersionUID</a> <a href="#">update</a>
general	<a href="#">JFrame</a>
specific	<a href="#">DemoJFileChooser</a>
implemented interfaces	<a href="#">Observer</a>
source of relation	InterfaceRealization <a href="#">Observer</a>
target of relation	ComponentRealization <a href="#">Implementierung</a>

## Operation JAbstractView::dispose

owner	<a href="#">JAbstractView</a>
parameter	name direction type multiplicity default <b>return</b> <b>return void</b>
ownedMember	<b>return</b>
documentation	Has the same behavior as a normal dispose(), but also disconnects the view from a model.

## Operation JAbstractView::JAbstractView

owner	<a href="#">JAbstractView</a>
parameter	name direction type multiplicity default <b>model</b> in <b>M</b>

ownedMember	<b>model</b>
documentation	<p>Creates a new view which is connected to a model.</p> <p><b>@param model</b> the model which the view should connect to.</p>

#### Property **JAbstractView::model**

owner	<a href="#">JAbstractView</a>
-------	-------------------------------

#### Property **JAbstractView::serialVersionUID**

owner	<a href="#">JAbstractView</a>
documentation	Only for serialization

#### Operation **JAbstractView::update**

owner	<a href="#">JAbstractView</a>
parameter	name direction type multiplicity default <b>model in Observable argument in Object return return void</b>
ownedMember	<b>argument model return</b>
documentation	<p>Have to be implemented by an subclass.</p> <p>It is called every time a model is updated. It decides (based on the program state) what the view shows.</p> <p><b>@param model</b> this is a reference to the model which a view observe.</p> <p><b>@param argument</b> this is a reference of an argument passed to the model.</p> <p><b>@see #java.util.Observer.update(Observable, Object)</b></p>

## Class JBoxplot

diagram	<p>The diagram shows the class <b>JBoxplot</b> with a dependency to <b>M&gt;BoxplotModel</b>. The class has a final attribute <code>serialVersionUID:long=1L</code> and a constructor <code>JBoxplot(in model:M)</code>. It also has an operation <code>generateContent():void</code>.</p>
hierarchy	<pre> classDiagram     JAbstract2DDiagram&lt;M&gt; &lt; -- JBoxplot   </pre>
owner	<a href="#">diagrams</a>
template parameters	name kind constrainingClassifier default <b>M Class BoxplotModel</b>
ownedMember	<a href="#">generateContent</a> <a href="#">JBoxplot</a> <a href="#">serialVersionUID</a>
general	<a href="#">JAbstract2DDiagram&lt;M&gt;</a>
target of relation	ComponentRealization <a href="#">Implementierung</a>
documentation	<p>This class implements how the BoxplotModel is to be drawn.</p> <p>@author David Kaufman</p> <p>@param &lt;M&gt;</p>

## Operation JBoxplot::generateContent

owner	<a href="#">JBoxplot</a>
parameter	name direction type multiplicity default <b>return return void</b>
ownedMember	<b>return</b>

## Operation JBoxplot::JBoxplot

owner	<a href="#">JBoxplot</a>
parameter	name direction type multiplicity default <b>model in M</b>
ownedMember	<b>model</b>
documentation	<p>Constructor</p> <p>@param abstractModel the model from which the drawing information is taken</p>

## Property JBoxplot::serialVersionUID

owner	<a href="#">JBoxplot</a>
-------	--------------------------



## Class JCluster3D

diagram	<pre> classDiagram     JCluster3D &lt; -- JAbstract3DDiagram&lt;M&gt;     JCluster3D "1..&gt; M Cluster3DModel     JCluster3D &lt;&lt;final&gt;&gt; serialVersionUID: long = 1L     JCluster3D &lt;&lt;constructor&gt;&gt; JCluster3D(in model: M)     JCluster3D &lt;&lt;annotation&gt;&gt; generateContent(): void     JCluster3D &lt;&lt;annotation&gt;&gt; modifiers = Override   </pre>
hierarchy	<pre> classDiagram     JAbstract3DDiagram&lt;M&gt; &lt; -- JCluster3D   </pre>
owner	<a href="#">diagrams</a>
template parameters	name kind constrainingClassifier default <b>M Class Cluster3DModel</b>
ownedMember	<a href="#">generateContent JCluster3D serialVersionUID</a>
general	<a href="#">JAbstract3DDiagram&lt;M&gt;</a>
target of relation	ComponentRealization <a href="#">Implementierung</a>
documentation	<p>This class implements how the Cluster3DModel is to be drawn.</p> <p>@author David Kaufman</p> <p>@param &lt;M&gt;</p>

## Operation JCluster3D::generateContent

owner	<a href="#">JCluster3D</a>
parameter	name direction type multiplicity default <b>return return void</b>
ownedMember	<b>return</b>

## Operation JCluster3D::JCluster3D

owner	<a href="#">JCluster3D</a>
parameter	name direction type multiplicity default <b>model in M</b>
ownedMember	<b>model</b>
documentation	<p>Constructor</p> <p>@param abstractModel the model from which the drawing information is taken</p>

## Property JCluster3D::serialVersionUID

owner	<a href="#">JCluster3D</a>
-------	----------------------------



## Class JDataFormatHelper

diagram	<pre> classDiagram     class JDataFormatHelper {         &lt;&lt;constructor&gt;&gt; JDataFormatHelper()         byteToInt(in b:byte):int         byteToLong(in b:byte):long         buf4ToInt(in buf:byte[*], in offset:int, in blintel:boolean):int         buf4ToIntSigned(in buf:byte[*], in offset:int, in blintel:boolean):int         buf2ToInt(in buf:byte[*], in offset:int, in blintel:boolean):int         buf2ToIntBE(in buf:byte[*], in offset:int):int         buf2ToShortBE(in buf:byte[*], in offset:int):short         buf4ToIntBE(in buf:byte[*], in offset:int):int         buf4ToLong(in buf:byte[*], in offset:int, in blintel:boolean):long         buf4ToLongSigned(in buf:byte[*], in offset:int):long         buf1ToShort(in buf:byte[*], in offset:int):short         IntToBuf2(in val:int, in buf:byte[*], in offset:int, in blintel:boolean):void         IntToBuf2BE(in val:int, in buf:byte[*], in offset:int):void         IntToBuf4(in val:int, in buf:byte[*], in offset:int, in blintel:boolean):void         LongToBuf4(in val:long, in buf:byte[*], in offset:int):void         ShortToBuf1(in val:short, in buf:byte[*], in offset:int):void         IntSignedToBuf4(in val:int, in buf:byte[*], in offset:int):void     }   </pre>
owner	<b>dataformathelper</b>
ownedMember	<b>buf1ToShort buf2ToInt buf2ToIntBE buf2ToShortBE buf4ToInt buf4ToIntBE buf4ToIntSigned buf4ToLong buf4ToLongSigned byteToInt byteToLong IntSignedToBuf4 IntToBuf2 IntToBuf2BE IntToBuf4 JDataFormatHelper LongToBuf4 ShortToBuf1</b>
target of relation	ComponentRealization <a href="#">Implementierung</a>
documentation	@author reiner

### Operation JDataFormatHelper::buf1ToShort

owner	<b>JDataFormatHelper</b>
parameter	name direction type multiplicity default <b>buf in byte * offset in int return return short</b>
ownedMember	<b>buf offset return</b>

### Operation JDataFormatHelper::buf2ToInt

owner	<b>JDataFormatHelper</b>
parameter	name direction type multiplicity default <b>buf in byte * offset in int blintel in boolean return return int</b>
ownedMember	<b>blintel buf offset return</b>

### Operation JDataFormatHelper::buf2ToIntBE

owner	<b>JDataFormatHelper</b>
-------	--------------------------

parameter	name direction type multiplicity default <b>buf in byte * offset in int return return int</b>
ownedMember	<b>buf offset return</b>

#### Operation **JDataFormatHelper::buf2ToShortBE**

owner	<a href="#"><b>JDataFormatHelper</b></a>
parameter	name direction type multiplicity default <b>buf in byte * offset in int return return short</b>
ownedMember	<b>buf offset return</b>

#### Operation **JDataFormatHelper::buf4ToInt**

owner	<a href="#"><b>JDataFormatHelper</b></a>
parameter	name direction type multiplicity default <b>buf in byte * offset in int bInt in boolean return return int</b>
ownedMember	<b>bInt buf offset return</b>

#### Operation **JDataFormatHelper::buf4ToIntBE**

owner	<a href="#"><b>JDataFormatHelper</b></a>
parameter	name direction type multiplicity default <b>buf in byte * offset in int return return int</b>
ownedMember	<b>buf offset return</b>

#### Operation **JDataFormatHelper::buf4ToIntSigned**

owner	<a href="#"><b>JDataFormatHelper</b></a>
parameter	name direction type multiplicity default <b>buf in byte * offset in int bInt in boolean return return int</b>
ownedMember	<b>bInt buf offset return</b>

#### Operation **JDataFormatHelper::buf4ToLong**

owner	<a href="#"><b>JDataFormatHelper</b></a>
parameter	name direction type multiplicity default <b>buf in byte * offset in int bInt in boolean return return long</b>
ownedMember	<b>bInt buf offset return</b>

#### Operation **JDataFormatHelper::buf4ToLongSigned**

owner	<a href="#"><b>JDataFormatHelper</b></a>
parameter	name direction type multiplicity default <b>buf in byte * offset in int return return long</b>
ownedMember	<b>buf offset return</b>

#### Operation **JDataFormatHelper::byteToInt**

owner	<a href="#"><b>JDataFormatHelper</b></a>
parameter	name direction type multiplicity default <b>b in byte return return int</b>
ownedMember	<b>b return</b>

#### Operation **JDataFormatHelper::byteToLong**

owner	<a href="#">JDataFormatHelper</a>
parameter	name direction type multiplicity default <b>b in byte</b> return return long
ownedMember	<b>b return</b>

#### Operation **JDataFormatHelper::IntSignedToBuf4**

owner	<a href="#">JDataFormatHelper</a>
parameter	name direction type multiplicity default <b>val in int</b> <b>buf in byte</b> * <b>offset in int</b> return return void
ownedMember	<b>buf offset return val</b>

#### Operation **JDataFormatHelper::IntToBuf2**

owner	<a href="#">JDataFormatHelper</a>
parameter	name direction type multiplicity default <b>val in int</b> <b>buf in byte</b> * <b>offset in int</b> <b>blintel in boolean</b> return return void
ownedMember	<b>blintel buf offset return val</b>

#### Operation **JDataFormatHelper::IntToBuf2BE**

owner	<a href="#">JDataFormatHelper</a>
parameter	name direction type multiplicity default <b>val in int</b> <b>buf in byte</b> * <b>offset in int</b> return return void
ownedMember	<b>buf offset return val</b>

#### Operation **JDataFormatHelper::IntToBuf4**

owner	<a href="#">JDataFormatHelper</a>
parameter	name direction type multiplicity default <b>val in int</b> <b>buf in byte</b> * <b>offset in int</b> <b>blintel in boolean</b> return return void
ownedMember	<b>blintel buf offset return val</b>

#### Operation **JDataFormatHelper::JDataFormatHelper**

owner	<a href="#">JDataFormatHelper</a>
documentation	Creates a new instance of JDataFormatHelper

#### Operation **JDataFormatHelper::LongToBuf4**

owner	<a href="#">JDataFormatHelper</a>
parameter	name direction type multiplicity default <b>val in long</b> <b>buf in byte</b> * <b>offset in int</b> return return void
ownedMember	<b>buf offset return val</b>

### Operation **JDataFormatHelper::ShortToBuf1**

owner	<a href="#">JDataFormatHelper</a>
parameter	name direction type multiplicity default <b>val in short</b> <b>buf in byte *</b> <b>offset in int</b> <b>return return void</b>
ownedMember	<b>buf offset return val</b>

## Class JDataFormatHelper

diagram	<pre> classDiagram     class JDataFormatHelper {         &lt;&lt;constructor&gt;&gt; JDataFormatHelper()         byteToInt(in b:byte):int         byteToLong(in b:byte):long         buf4ToInt(in buf:byte[*], in offset:int, in blintel:boolean):int         buf4ToIntSigned(in buf:byte[*], in offset:int, in blintel:boolean):int         buf2ToInt(in buf:byte[*], in offset:int, in blintel:boolean):int         buf2ToIntBE(in buf:byte[*], in offset:int):int         buf2ToShortBE(in buf:byte[*], in offset:int):short         buf4ToIntBE(in buf:byte[*], in offset:int):int         buf4ToLong(in buf:byte[*], in offset:int, in blintel:boolean):long         buf4ToLongSigned(in buf:byte[*], in offset:int):long         buf1ToShort(in buf:byte[*], in offset:int):short         IntToBuf2(in val:int, in buf:byte[*], in offset:int, in blintel:boolean):void         IntToBuf2BE(in val:int, in buf:byte[*], in offset:int):void         IntToBuf4(in val:int, in buf:byte[*], in offset:int, in blintel:boolean):void         LongToBuf4(in val:long, in buf:byte[*], in offset:int):void         ShortToBuf1(in val:short, in buf:byte[*], in offset:int):void         IntSignedToBuf4(in val:int, in buf:byte[*], in offset:int):void     }   </pre>
owner	<b>dataformathelper</b>
ownedMember	<b>buf1ToShort buf2ToInt buf2ToIntBE buf2ToShortBE buf4ToInt buf4ToIntBE buf4ToIntSigned buf4ToLong buf4ToLongSigned byteToInt byteToLong IntSignedToBuf4 IntToBuf2 IntToBuf2BE IntToBuf4 JDataFormatHelper LongToBuf4 ShortToBuf1</b>
target of relation	ComponentRealization <a href="#">Implementierung</a>
documentation	@author reiner

### Operation JDataFormatHelper::buf1ToShort

owner	<b>JDataFormatHelper</b>
parameter	name direction type multiplicity default <b>buf in byte * offset in int return return short</b>
ownedMember	<b>buf offset return</b>

### Operation JDataFormatHelper::buf2ToInt

owner	<b>JDataFormatHelper</b>
parameter	name direction type multiplicity default <b>buf in byte * offset in int blintel in boolean return return int</b>
ownedMember	<b>blintel buf offset return</b>

### Operation JDataFormatHelper::buf2ToIntBE

owner	<b>JDataFormatHelper</b>
-------	--------------------------

parameter	name direction type multiplicity default <b>buf in byte * offset in int return return int</b>
ownedMember	<b>buf offset return</b>

#### Operation **JDataFormatHelper::buf2ToShortBE**

owner	<a href="#"><b>JDataFormatHelper</b></a>
parameter	name direction type multiplicity default <b>buf in byte * offset in int return return short</b>
ownedMember	<b>buf offset return</b>

#### Operation **JDataFormatHelper::buf4ToInt**

owner	<a href="#"><b>JDataFormatHelper</b></a>
parameter	name direction type multiplicity default <b>buf in byte * offset in int bInt in boolean return return int</b>
ownedMember	<b>bInt buf offset return</b>

#### Operation **JDataFormatHelper::buf4ToIntBE**

owner	<a href="#"><b>JDataFormatHelper</b></a>
parameter	name direction type multiplicity default <b>buf in byte * offset in int return return int</b>
ownedMember	<b>buf offset return</b>

#### Operation **JDataFormatHelper::buf4ToIntSigned**

owner	<a href="#"><b>JDataFormatHelper</b></a>
parameter	name direction type multiplicity default <b>buf in byte * offset in int bInt in boolean return return int</b>
ownedMember	<b>bInt buf offset return</b>

#### Operation **JDataFormatHelper::buf4ToLong**

owner	<a href="#"><b>JDataFormatHelper</b></a>
parameter	name direction type multiplicity default <b>buf in byte * offset in int bInt in boolean return return long</b>
ownedMember	<b>bInt buf offset return</b>

#### Operation **JDataFormatHelper::buf4ToLongSigned**

owner	<a href="#"><b>JDataFormatHelper</b></a>
parameter	name direction type multiplicity default <b>buf in byte * offset in int return return long</b>
ownedMember	<b>buf offset return</b>

#### Operation **JDataFormatHelper::byteToInt**

owner	<a href="#"><b>JDataFormatHelper</b></a>
parameter	name direction type multiplicity default <b>b in byte return return int</b>
ownedMember	<b>b return</b>

#### Operation **JDataFormatHelper::byteToLong**

owner	<a href="#">JDataFormatHelper</a>
parameter	name direction type multiplicity default <b>b in byte</b> return return long
ownedMember	<b>b return</b>

#### Operation **JDataFormatHelper::IntSignedToBuf4**

owner	<a href="#">JDataFormatHelper</a>
parameter	name direction type multiplicity default <b>val in int</b> <b>buf in byte</b> * <b>offset in int</b> return return void
ownedMember	<b>buf offset return val</b>

#### Operation **JDataFormatHelper::IntToBuf2**

owner	<a href="#">JDataFormatHelper</a>
parameter	name direction type multiplicity default <b>val in int</b> <b>buf in byte</b> * <b>offset in int</b> <b>blintel in boolean</b> return return void
ownedMember	<b>blintel buf offset return val</b>

#### Operation **JDataFormatHelper::IntToBuf2BE**

owner	<a href="#">JDataFormatHelper</a>
parameter	name direction type multiplicity default <b>val in int</b> <b>buf in byte</b> * <b>offset in int</b> return return void
ownedMember	<b>buf offset return val</b>

#### Operation **JDataFormatHelper::IntToBuf4**

owner	<a href="#">JDataFormatHelper</a>
parameter	name direction type multiplicity default <b>val in int</b> <b>buf in byte</b> * <b>offset in int</b> <b>blintel in boolean</b> return return void
ownedMember	<b>blintel buf offset return val</b>

#### Operation **JDataFormatHelper::JDataFormatHelper**

owner	<a href="#">JDataFormatHelper</a>
documentation	Creates a new instance of JDataFormatHelper

#### Operation **JDataFormatHelper::LongToBuf4**

owner	<a href="#">JDataFormatHelper</a>
parameter	name direction type multiplicity default <b>val in long</b> <b>buf in byte</b> * <b>offset in int</b> return return void
ownedMember	<b>buf offset return val</b>

### Operation **JDataFormatHelper::ShortToBuf1**

owner	<a href="#">JDataFormatHelper</a>
parameter	name direction type multiplicity default <b>val in short</b> <b>buf in byte *</b> <b>offset in int</b> <b>return return void</b>
ownedMember	<b>buf offset return val</b>

## Class JDiaDiagramButtons2D

diagram	<pre> JDiaDiagramButtons2D +--&lt;&gt; serialVersionUID:long=1L +--&lt;&gt; JDiaDiagramButtons2D() </pre>
hierarchy	<pre> classDiagram     JDiaDiagramButtons2D &lt; -- JPanel     JPanel &lt; -- JComponent     JComponent &lt; -- Container     Container &lt; -- Component     Component --&gt; ImageObserver     Component --&gt; MenuContainer     Component --&gt; Serializable     Serializable --&gt; Accessible </pre>
owner	<a href="#">diagrams</a>
ownedMember	<a href="#">JDiaDiagramButtons2D serialVersionUID</a>
general	<a href="#">JPanel</a>
target of relation	ComponentRealization <a href="#">Implementierung</a>
documentation	<p>This class represents the buttons which are available in ever 2D diagram.</p> <p>@author David Kaufman</p>

## Operation JDiaDiagramButtons2D::JDiaDiagramButtons2D

owner	<a href="#">JDiaDiagramButtons2D</a>
-------	--------------------------------------

## Property JDiaDiagramButtons2D::serialVersionUID

owner	<a href="#">JDiaDiagramButtons2D</a>
-------	--------------------------------------

## Class JDiaDiagramButtons3D

diagram	<pre> JDiaDiagramButtons3D +--&lt;&gt; serialVersionUID:long=1L +--&lt;&gt; JDiaDiagramButtons3D() </pre>
hierarchy	<pre> classDiagram     JDiaDiagramButtons3D &lt; -- JPanel     JPanel &lt; -- JComponent     JComponent &lt; -- Container     Container &lt; -- Component     Component &lt; -- ImageObserver     Component &lt; -- MenuContainer     Component &lt; -- Serializable     JComponent &lt; -- Serializable     JComponent &lt; -- Accessible </pre>
owner	<a href="#">diagrams</a>
ownedMember	<a href="#">JDiaDiagramButtons3D serialVersionUID</a>
general	<a href="#">JPanel</a>
target of relation	ComponentRealization <a href="#">Implementierung</a>
documentation	<p>This class represents the buttons which are available in ever 3D diagram.</p> <p>@author David Kaufman</p>

## Operation JDiaDiagramButtons3D::JDiaDiagramButtons3D

owner	<a href="#">JDiaDiagramButtons3D</a>
-------	--------------------------------------

## Property JDiaDiagramButtons3D::serialVersionUID

owner	<a href="#">JDiaDiagramButtons3D</a>
-------	--------------------------------------

## Class JDiaDiagramButtonsPlain

diagram	<pre> JDiaDiagramButtonsPlain +--&gt; serialVersionUID:long=1L +--&gt; JDiaDiagramButtonsPlain() </pre>
hierarchy	<pre> classDiagram     JDiaDiagramButtonsPlain &lt; -- JPanel     JPanel &lt; -- JComponent     JComponent &lt; -- Container     Container &lt; -- Component     Component &lt; -- ImageObserver     Component &lt; -- MenuContainer     Serializable &lt; -- JComponent     Accessible &lt; -- JPanel </pre>
owner	<a href="#">diagrams</a>
ownedMember	<a href="#">JDiaDiagramButtonsPlain serialVersionUID</a>
general	<a href="#">JPanel</a>
target of relation	ComponentRealization <a href="#">Implementierung</a>
documentation	<p>This class represents the buttons which are available in the text and table diagram.</p> <p>@author David Kaufman</p>

## Operation JDiaDiagramButtonsPlain::JDiaDiagramButtonsPlain

owner	<a href="#">JDiaDiagramButtonsPlain</a>
-------	---

## Property JDiaDiagramButtonsPlain::serialVersionUID

owner	<a href="#">JDiaDiagramButtonsPlain</a>
-------	---

## Class JDiaDiagramType

diagram	<pre> <b>JDiagramType</b>   JTextField reportname   long serialVersionUID   JList meineliste    &lt;&lt;constructor&gt;&gt; JDiaDiagramType(in titel:String, in icon:Icon, in tip:String)   int getSelectedDiagramType()   String getReportName()   String getReportDescription() </pre>
hierarchy	<pre> classDiagram     JDiaDiagramType &lt; -- JAbstractSinglePanel     JAbstractSinglePanel &lt; -- JComponent     JComponent &lt; -- Container     Container &lt; -- Component     Component &lt;--&gt; &gt; Serializable   </pre>
owner	<a href="#">reportmanagement</a>
ownedMember	<a href="#">getReportDescription</a> <a href="#">getReportName</a> <a href="#">getSelectedDiagramType</a> <a href="#">JDiaDiagramType</a> <a href="#">meineliste</a> <a href="#">reportname</a> <a href="#">serialVersionUID</a>
general	<a href="#">JAbstractSinglePanel</a>
target of relation	ComponentRealization <a href="#">Implementierung</a>
typedElements	Class <a href="#">DiagramTypeSelectController</a> Property <a href="#">view</a> Operation <a href="#">DiagramTypeSelectController</a>
documentation	<p>This class represents the panel where the user can choose the diagram type and assign the report name and report description.</p> <p>Note that this panel is present in every report compilation.</p> <p>@author David Kaufman</p>

## Operation JDiaDiagramType::getReportDescription

owner	<a href="#">JDiaDiagramType</a>
parameter	name direction type multiplicity default <b>return</b> <b>return</b> <b>String</b>
ownedMember	<b>return</b>

## Operation JDiaDiagramType::getReportName

owner	<a href="#">JDiaDiagramType</a>
-------	---------------------------------

parameter	name direction type multiplicity default <b>return return String</b>
ownedMember	<b>return</b>

#### Operation **JDiagramType::getSelectedDiagramType**

owner	<b>JDiagramType</b>
parameter	name direction type multiplicity default <b>return return int</b>
ownedMember	<b>return</b>
documentation	<p>Returns the currently selected index in the diagram type selection list</p> <p>@return the current selection index</p>

#### Operation **JDiagramType::JDiagramType**

owner	<b>JDiagramType</b>
parameter	name direction type multiplicity default <b>titel in String icon in Icon tip in String</b>
ownedMember	<b>icon tip titel</b>
documentation	<p>Constructor which initialized this diagramm selection panel</p> <p>@param titel The title which is registered with this panel.</p> <p>@param icon The icon which is registered with this panel.</p> <p>@param tip The tooltip which is registered with this panel.</p>

#### Property **JDiagramType::meineliste**

owner	<b>JDiagramType</b>
-------	---------------------

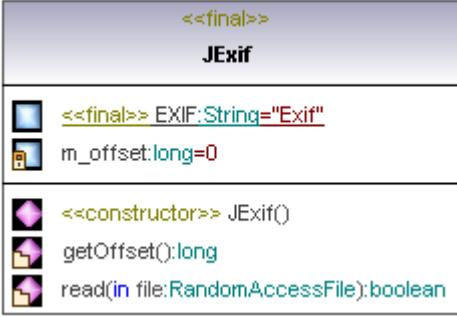
#### Property **JDiagramType::reportname**

owner	<b>JDiagramType</b>
-------	---------------------

#### Property **JDiagramType::serialVersionUID**

owner	<b>JDiagramType</b>
-------	---------------------

## Class JExif

diagram	
owner	<a href="#">jexifviewer</a>
ownedMember	<a href="#">EXIF getOffset JExif m_offset read</a>
target of relation	ComponentRealization <a href="#">Implementierung</a>
documentation	@author reiner

## Property JExif::EXIF

owner	<a href="#">JExif</a>
-------	-----------------------

## Operation JExif::getOffset

owner	<a href="#">JExif</a>
parameter	name direction type multiplicity default <b>return return long</b>
ownedMember	<b>return</b>

## Operation JExif::JExif

owner	<a href="#">JExif</a>
documentation	Creates a new instance of JExif @see <a href="http://www.exif.org/Exif2-2.PDF">Exif specification</a>

## Property JExif::m\_offset

owner	<a href="#">JExif</a>
-------	-----------------------

## Operation JExif::read

owner	<a href="#">JExif</a>
parameter	name direction type multiplicity default <b>file in RandomAccessFile return return boolean</b>
ownedMember	<b>file return</b>



## Class JExif

diagram	
owner	<a href="#">jexifviewer</a>
ownedMember	<a href="#">EXIF getOffset JExif m_offset read</a>
target of relation	ComponentRealization <a href="#">Implementierung</a>
documentation	@author reiner

## Property JExif::EXIF

owner	<a href="#">JExif</a>
-------	-----------------------

## Operation JExif::getOffset

owner	<a href="#">JExif</a>
parameter	name direction type multiplicity default <b>return return long</b>
ownedMember	<b>return</b>

## Operation JExif::JExif

owner	<a href="#">JExif</a>
documentation	Creates a new instance of JExif @see <a href="http://www.exif.org/Exif2-2.PDF">Exif specification</a>

## Property JExif::m\_offset

owner	<a href="#">JExif</a>
-------	-----------------------

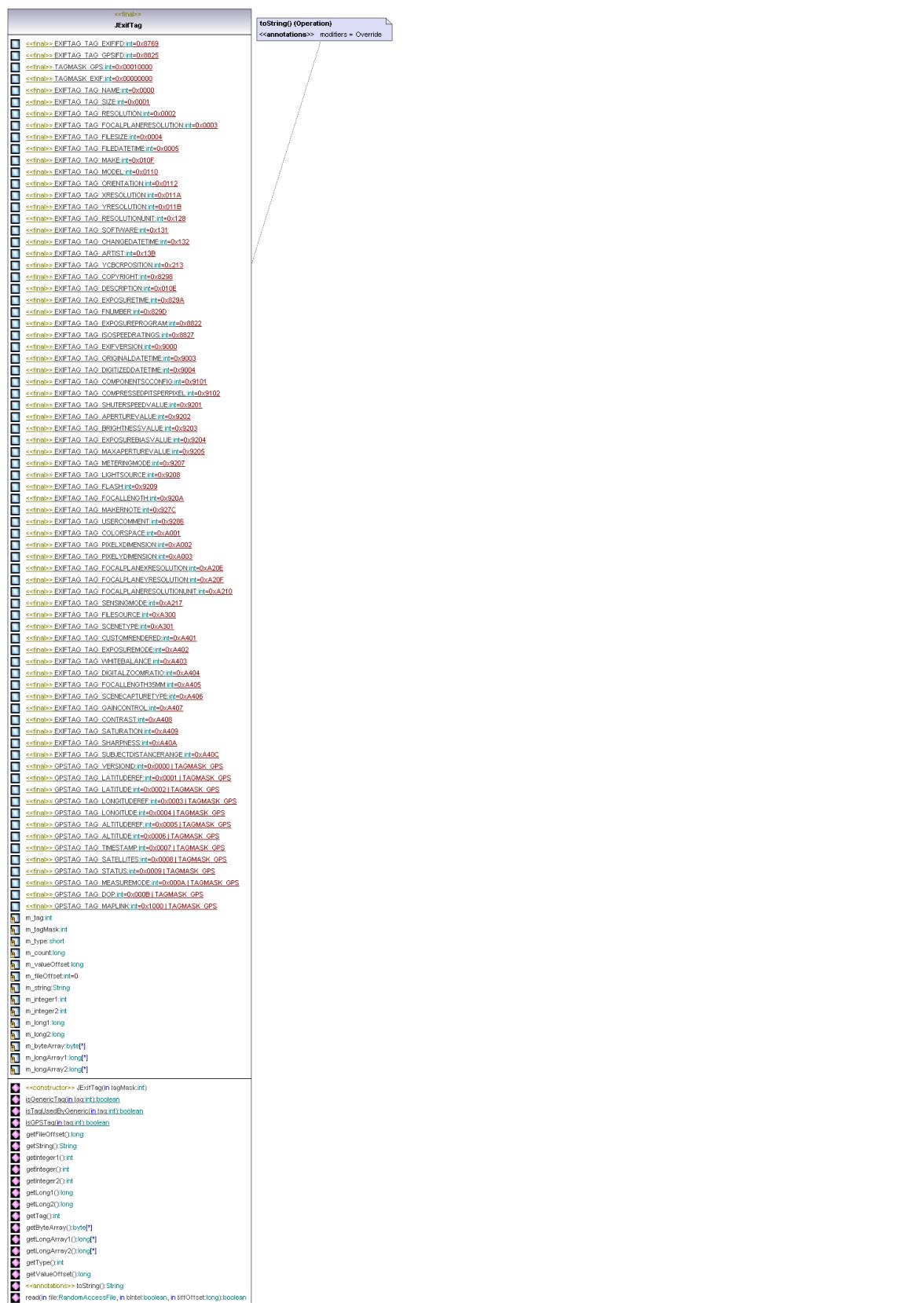
## Operation JExif::read

owner	<a href="#">JExif</a>
parameter	name direction type multiplicity default <b>file in RandomAccessFile return return boolean</b>
ownedMember	<b>file return</b>



## Class JExifTag

diagram



owner	<a href="#">jexifviewer</a>
ownedMember	<a href="#">EXIFTAG_TAG_APERTUREVALUE</a> <a href="#">EXIFTAG_TAG_ARTIST</a> <a href="#">EXIFTAG_TAG_BRIGHTNESSVALUE</a> <a href="#">EXIFTAG_TAG_CHANGEDATETIME</a> <a href="#">EXIFTAG_TAG_COLORSPACE</a> <a href="#">EXIFTAG_TAG_COMPONENTSCONFIG</a> <a href="#">EXIFTAG_TAG_COMPRESSEDPIXELSPERPIXEL</a> <a href="#">EXIFTAG_TAG_CONTRAST</a> <a href="#">EXIFTAG_TAG_COPYRIGHT</a> <a href="#">EXIFTAG_TAG_CUSTOM RENDERED</a> <a href="#">EXIFTAG_TAG_DESCRIPTION</a> <a href="#">EXIFTAG_TAG_DIGITALZOOMRATIO</a> <a href="#">EXIFTAG_TAG_DIGITIZEDDATETIME</a> <a href="#">EXIFTAG_TAG_EXIFID</a> <a href="#">EXIFTAG_TAG_EXIFVERSION</a> <a href="#">EXIFTAG_TAG_EXPOSUREBIASVALUE</a> <a href="#">EXIFTAG_TAG_EXPOSUREMODE</a> <a href="#">EXIFTAG_TAG_EXPOSUREPROGRAM</a> <a href="#">EXIFTAG_TAG_EXPOSURETIME</a> <a href="#">EXIFTAG_TAG_FILEDATETIME</a> <a href="#">EXIFTAG_TAG_FILESIZEx</a> <a href="#">EXIFTAG_TAG_FILESOURCE</a> <a href="#">EXIFTAG_TAG_FLASH</a> <a href="#">EXIFTAG_TAG_FNUMBER</a> <a href="#">EXIFTAG_TAG_FOCALLength</a> <a href="#">EXIFTAG_TAG_FOCALPLANE Resolution</a> <a href="#">EXIFTAG_TAG_FOCALPLANE Resolution Unit</a> <a href="#">EXIFTAG_TAG_FOCALPLANE Resolution</a> <a href="#">EXIFTAG_TAG_FOCALPLANE Resolution</a> <a href="#">EXIFTAG_TAG_GAINCONTROL</a> <a href="#">EXIFTAG_TAG_GPSIFD</a> <a href="#">EXIFTAG_TAG_ISOSPEEDRATINGS</a> <a href="#">EXIFTAG_TAG_LIGHTSOURCE</a> <a href="#">EXIFTAG_TAG_MAKE</a> <a href="#">EXIFTAG_TAG_MAKERNOTE</a> <a href="#">EXIFTAG_TAG_MAXAPERTUREVALUE</a> <a href="#">EXIFTAG_TAG_METERINGMODE</a> <a href="#">EXIFTAG_TAG_MODEL</a> <a href="#">EXIFTAG_TAG_NAME</a> <a href="#">EXIFTAG_TAG_ORIENTATION</a> <a href="#">EXIFTAG_TAG_ORIGINALDATETIME</a> <a href="#">EXIFTAG_TAG_PIXELDIMENSION</a> <a href="#">EXIFTAG_TAG_PIXELYDIMENSION</a> <a href="#">EXIFTAG_TAG_RESOLUTION</a> <a href="#">EXIFTAG_TAG_RESOLUTIONUNIT</a> <a href="#">EXIFTAG_TAG_SATURATION</a> <a href="#">EXIFTAG_TAG_SCENE CAPTURETYPE</a> <a href="#">EXIFTAG_TAG_SCENETYPE</a> <a href="#">EXIFTAG_TAG_SENSINGMODE</a> <a href="#">EXIFTAG_TAG_SHARPNESS</a> <a href="#">EXIFTAG_TAG_SHUTERSPEEDVALUE</a> <a href="#">EXIFTAG_TAG_SIZE</a> <a href="#">EXIFTAG_TAG_SOFTWARE</a> <a href="#">EXIFTAG_TAG SUBJECTDISTANCERANGE</a> <a href="#">EXIFTAG_TAG_USERCOMMENT</a> <a href="#">EXIFTAG_TAG_WHITEBALANCE</a> <a href="#">EXIFTAG_TAG_XRESOLUTION</a> <a href="#">EXIFTAG_TAG_YCBRPOSITION</a> <a href="#">EXIFTAG_TAG_YRESOLUTION</a> <a href="#">getByteArray</a> <a href="#">getFileOffset</a> <a href="#">getInteger</a> <a href="#">getInteger1</a> <a href="#">getInteger2</a> <a href="#">getLong1</a> <a href="#">getLong2</a> <a href="#">getLongArray1</a> <a href="#">getLongArray2</a> <a href="#">getString</a> <a href="#">getTag</a> <a href="#">getType</a> <a href="#">getValueOffset</a> <a href="#">GPSTAG_TAG_ALTITUDE</a> <a href="#">GPSTAG_TAG_ALITUDEREF</a> <a href="#">GPSTAG_TAG_DOP</a> <a href="#">GPSTAG_TAG_LATITUDE</a> <a href="#">GPSTAG_TAG_LATITUDEREf</a> <a href="#">GPSTAG_TAG_LONGITUDE</a> <a href="#">GPSTAG_TAG_LONGITUDEREf</a> <a href="#">GPSTAG_TAG_MAPLINK</a> <a href="#">GPSTAG_TAG_MEASUREMODE</a> <a href="#">GPSTAG_TAG_SATELLITES</a> <a href="#">GPSTAG_TAG_STATUS</a> <a href="#">GPSTAG_TAG_TIMESTAMP</a> <a href="#">GPSTAG_TAG_VERSIONID</a> <a href="#">isGenericTag</a> <a href="#">isGPSTag</a> <a href="#">isTagUsedByGeneric</a> <a href="#">JExifTag</a> <a href="#">m_byteArray</a> <a href="#">m_count</a> <a href="#">m_fileOffset</a> <a href="#">m_integer1</a> <a href="#">m_integer2</a> <a href="#">m_long1</a> <a href="#">m_long2</a> <a href="#">m_longArray1</a> <a href="#">m_longArray2</a> <a href="#">m_string</a> <a href="#">m_tag</a> <a href="#">m_tagMask</a> <a href="#">m_type</a> <a href="#">m_valueOffset</a> <a href="#">read</a> <a href="#">TAGMASK_EXIF TAGMASK_GPS</a> <a href="#">toString</a>
target of relation	ComponentRealization <a href="#">Implementierung</a>
typedElements	Class <a href="#">JIfd</a> Operation <a href="#">findTag</a> <a href="#">getGradFromExifTag</a>
documentation	@author reiner

#### Property JExifTag::EXIFTAG\_TAG\_APERTUREVALUE

owner	<a href="#">JExifTag</a>
-------	--------------------------

#### Property JExifTag::EXIFTAG\_TAG\_ARTIST

owner	<a href="#">JExifTag</a>
-------	--------------------------

#### Property JExifTag::EXIFTAG\_TAG\_BRIGHTNESSVALUE

owner	<a href="#">JExifTag</a>
-------	--------------------------

#### Property JExifTag::EXIFTAG\_TAG\_CHANGEDATETIME

owner	<a href="#">JExifTag</a>
-------	--------------------------

#### Property JExifTag::EXIFTAG\_TAG\_COLORSPACE

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_COMPONENTSCCONFIG**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_COMPRESSEDBITSPERPIXEL**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_CONTRAST**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_COPYRIGHT**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_CUSTOM RENDERED**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_DESCRIPTION**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_DIGITALZOOMRATIO**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_DIGITIZEDDATETIME**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_EXIFIFD**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_EXIFVERSION**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_EXPOSUREBIASVALUE**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_EXPOSUREMODE**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_EXPOSUREPROGRAM**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_EXPOSURETIME**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_FILEDATETIME**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_FILESIZE**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_FILESOURCE**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_FLASH**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_FNUMBER**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_FOCALLENGTH**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_FOCALLENGTH35MM**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_FOCALPLANERESOLUTION**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_FOCALPLANERESOLUTIONUNIT**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_FOCALPLANEXRESOLUTION**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_FOCALPLANEYRESOLUTION**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_GAINCONTROL**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_GPSIFD**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_ISOSPEEDRATINGS**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_LIGHTSOURCE**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_MAKE**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_MAKERNOTE**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_MAXAPERTUREVALUE**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_METERINGMODE**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_MODEL**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_NAME**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_ORIENTATION**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_ORIGINALDATETIME**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_PIXELXDIMENSION**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_PIXELYDIMENSION**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_RESOLUTION**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_RESOLUTIONUNIT**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_SATURATION**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_SCENECAPTURETYPE**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_SCENETYPE**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_SENSINGMODE**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_SHARPNESS**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_SHUTERSPEEDVALUE**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_SIZE**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_SOFTWARE**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_SUBJECTDISTANCERANGE**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_USERCOMMENT**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_WHITEBALANCE**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_XRESOLUTION**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_YCBCRPOSITION**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_YRESOLUTION**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Operation **JExifTag::getByteArray**

owner	<a href="#">JExifTag</a>
parameter	name direction type multiplicity default <b>return return byte *</b>
ownedMember	<b>return</b>

Operation **JExifTag::getFileOffset**

owner	<a href="#">JExifTag</a>
parameter	name direction type multiplicity default <b>return return long</b>
ownedMember	<b>return</b>

Operation **JExifTag::getInteger**

owner	<a href="#">JExifTag</a>
parameter	name direction type multiplicity default <b>return return int</b>
ownedMember	<b>return</b>

**Operation JExifTag::getInteger1**

owner	<a href="#">JExifTag</a>
parameter	name direction type multiplicity default <b>return return int</b>
ownedMember	<b>return</b>

**Operation JExifTag::getInteger2**

owner	<a href="#">JExifTag</a>
parameter	name direction type multiplicity default <b>return return int</b>
ownedMember	<b>return</b>

**Operation JExifTag::getLong1**

owner	<a href="#">JExifTag</a>
parameter	name direction type multiplicity default <b>return return long</b>
ownedMember	<b>return</b>

**Operation JExifTag::getLong2**

owner	<a href="#">JExifTag</a>
parameter	name direction type multiplicity default <b>return return long</b>
ownedMember	<b>return</b>

**Operation JExifTag::getLongArray1**

owner	<a href="#">JExifTag</a>
parameter	name direction type multiplicity default <b>return return long *</b>
ownedMember	<b>return</b>

**Operation JExifTag::getLongArray2**

owner	<a href="#">JExifTag</a>
parameter	name direction type multiplicity default <b>return return long *</b>
ownedMember	<b>return</b>

**Operation JExifTag::getString**

owner	<a href="#">JExifTag</a>
parameter	name direction type multiplicity default <b>return return String</b>
ownedMember	<b>return</b>

**Operation JExifTag::getTag**

owner	<a href="#">JExifTag</a>
-------	--------------------------

parameter	name direction type multiplicity default <b>return return int</b>
ownedMember	<b>return</b>

#### Operation **JExifTag::getType**

owner	<b>JExifTag</b>
parameter	name direction type multiplicity default <b>return return int</b>
ownedMember	<b>return</b>

#### Operation **JExifTag::getValueOffset**

owner	<b>JExifTag</b>
parameter	name direction type multiplicity default <b>return return long</b>
ownedMember	<b>return</b>

#### Property **JExifTag::GPSTAG\_TAG\_ALTITUDE**

owner	<b>JExifTag</b>
-------	-----------------

#### Property **JExifTag::GPSTAG\_TAG\_ALTITUDEREF**

owner	<b>JExifTag</b>
-------	-----------------

#### Property **JExifTag::GPSTAG\_TAG\_DOP**

owner	<b>JExifTag</b>
-------	-----------------

#### Property **JExifTag::GPSTAG\_TAG\_LATITUDE**

owner	<b>JExifTag</b>
-------	-----------------

#### Property **JExifTag::GPSTAG\_TAG\_LATITUDEREF**

owner	<b>JExifTag</b>
-------	-----------------

#### Property **JExifTag::GPSTAG\_TAG\_LONGITUDE**

owner	<b>JExifTag</b>
-------	-----------------

#### Property **JExifTag::GPSTAG\_TAG\_LONGITUDEREF**

owner	<b>JExifTag</b>
-------	-----------------

#### Property **JExifTag::GPSTAG\_TAG\_MAPLINK**

owner	<b>JExifTag</b>
-------	-----------------

Property **JExifTag::GPSTAG\_TAG\_MEASUREMODE**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::GPSTAG\_TAG\_SATELLITES**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::GPSTAG\_TAG\_STATUS**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::GPSTAG\_TAG\_TIMESTAMP**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::GPSTAG\_TAG\_VERSIONID**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Operation **JExifTag::isGenericTag**

owner	<a href="#">JExifTag</a>
parameter	name direction type multiplicity default <b>tag</b> in int return return boolean
ownedMember	<b>return tag</b>

Operation **JExifTag::isGPSTag**

owner	<a href="#">JExifTag</a>
parameter	name direction type multiplicity default <b>tag</b> in int return return boolean
ownedMember	<b>return tag</b>

Operation **JExifTag::isTagUsedByGeneric**

owner	<a href="#">JExifTag</a>
parameter	name direction type multiplicity default <b>tag</b> in int return return boolean
ownedMember	<b>return tag</b>

Operation **JExifTag::JExifTag**

owner	<a href="#">JExifTag</a>
parameter	name direction type multiplicity default <b>tagMask</b> in int
ownedMember	<b>tagMask</b>
documentation	Creates a new instance of JExifTag

Property **JExifTag::m\_byteArray**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::m\_count**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::m\_fileOffset**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::m\_integer1**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::m\_integer2**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::m\_long1**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::m\_long2**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::m\_longArray1**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::m\_longArray2**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::m\_string**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::m\_tag**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::m\_tagMask**

owner	<a href="#">JExifTag</a>
-------	--------------------------

**Property JExifTag::m\_type**

owner	<a href="#">JExifTag</a>
-------	--------------------------

**Property JExifTag::m\_valueOffset**

owner	<a href="#">JExifTag</a>
-------	--------------------------

**Operation JExifTag::read**

owner	<a href="#">JExifTag</a>
parameter	name direction type multiplicity default <b>file in RandomAccessFile bIntel in boolean tiffOffset in long return return boolean</b>
ownedMember	<b>bIntel file return tiffOffset</b>

**Property JExifTag::TAGMASK\_EXIF**

owner	<a href="#">JExifTag</a>
-------	--------------------------

**Property JExifTag::TAGMASK\_GPS**

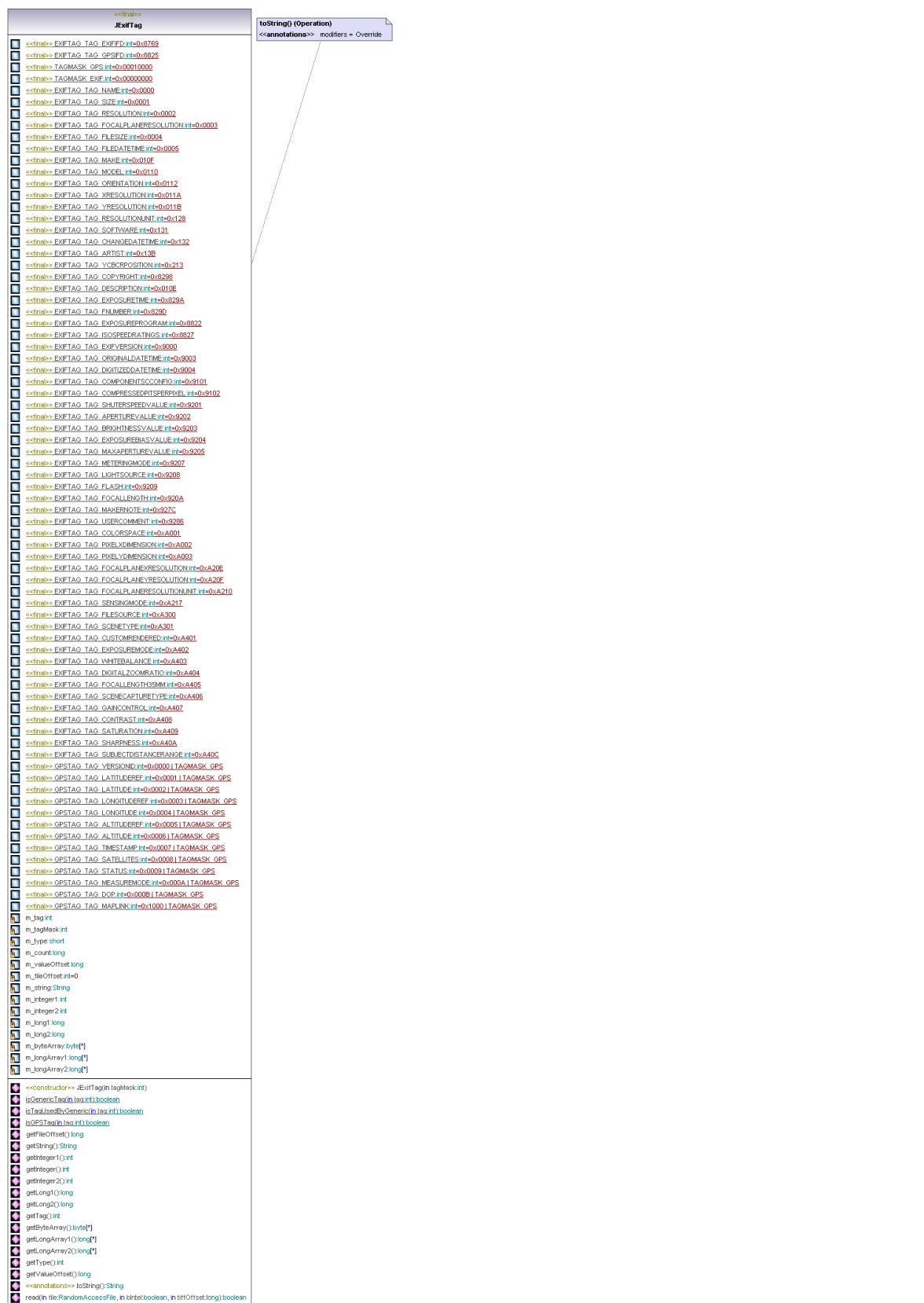
owner	<a href="#">JExifTag</a>
-------	--------------------------

**Operation JExifTag::toString**

owner	<a href="#">JExifTag</a>
parameter	name direction type multiplicity default <b>return return String</b>
ownedMember	<b>return</b>

## Class JExifTag

diagram



owner	<a href="#">jexifviewer</a>
ownedMember	<a href="#">EXIFTAG_TAG_APERTUREVALUE</a> <a href="#">EXIFTAG_TAG_ARTIST</a> <a href="#">EXIFTAG_TAG_BRIGHTNESSVALUE</a> <a href="#">EXIFTAG_TAG_CHANGEDATETIME</a> <a href="#">EXIFTAG_TAG_COLORSPACE</a> <a href="#">EXIFTAG_TAG_COMPONENTSCONFIG</a> <a href="#">EXIFTAG_TAG_COMPRESSEDPIXELSPERPIXEL</a> <a href="#">EXIFTAG_TAG_CONTRAST</a> <a href="#">EXIFTAG_TAG_COPYRIGHT</a> <a href="#">EXIFTAG_TAG_CUSTOM RENDERED</a> <a href="#">EXIFTAG_TAG_DESCRIPTION</a> <a href="#">EXIFTAG_TAG_DIGITALZOOMRATIO</a> <a href="#">EXIFTAG_TAG_DIGITIZEDDATETIME</a> <a href="#">EXIFTAG_TAG_EXIFID</a> <a href="#">EXIFTAG_TAG_EXIFVERSION</a> <a href="#">EXIFTAG_TAG_EXPOSUREBIASVALUE</a> <a href="#">EXIFTAG_TAG_EXPOSUREMODE</a> <a href="#">EXIFTAG_TAG_EXPOSUREPROGRAM</a> <a href="#">EXIFTAG_TAG_EXPOSURETIME</a> <a href="#">EXIFTAG_TAG_FILEDATETIME</a> <a href="#">EXIFTAG_TAG_FILESIZEx</a> <a href="#">EXIFTAG_TAG_FILESOURCE</a> <a href="#">EXIFTAG_TAG_FLASH</a> <a href="#">EXIFTAG_TAG_FNUMBER</a> <a href="#">EXIFTAG_TAG_FOCALLength</a> <a href="#">EXIFTAG_TAG_FOCALPLANE Resolution</a> <a href="#">EXIFTAG_TAG_FOCALPLANE Resolution Unit</a> <a href="#">EXIFTAG_TAG_FOCALPLANE Resolution</a> <a href="#">EXIFTAG_TAG_FOCALPLANE Resolution</a> <a href="#">EXIFTAG_TAG_GAINCONTROL</a> <a href="#">EXIFTAG_TAG_GPSIFD</a> <a href="#">EXIFTAG_TAG_ISOSPEEDRATINGS</a> <a href="#">EXIFTAG_TAG_LIGHTSOURCE</a> <a href="#">EXIFTAG_TAG_MAKE</a> <a href="#">EXIFTAG_TAG_MAKERNOTE</a> <a href="#">EXIFTAG_TAG_MAXAPERTUREVALUE</a> <a href="#">EXIFTAG_TAG_METERINGMODE</a> <a href="#">EXIFTAG_TAG_MODEL</a> <a href="#">EXIFTAG_TAG_NAME</a> <a href="#">EXIFTAG_TAG_ORIENTATION</a> <a href="#">EXIFTAG_TAG_ORIGINALDATETIME</a> <a href="#">EXIFTAG_TAG_PIXELDIMENSION</a> <a href="#">EXIFTAG_TAG_PIXELYDIMENSION</a> <a href="#">EXIFTAG_TAG_RESOLUTION</a> <a href="#">EXIFTAG_TAG_RESOLUTIONUNIT</a> <a href="#">EXIFTAG_TAG_SATURATION</a> <a href="#">EXIFTAG_TAG_SCENE CAPTURETYPE</a> <a href="#">EXIFTAG_TAG_SCENETYPE</a> <a href="#">EXIFTAG_TAG_SENSINGMODE</a> <a href="#">EXIFTAG_TAG_SHARPNESS</a> <a href="#">EXIFTAG_TAG_SHUTERSPEEDVALUE</a> <a href="#">EXIFTAG_TAG_SIZE</a> <a href="#">EXIFTAG_TAG_SOFTWARE</a> <a href="#">EXIFTAG_TAG_SUBJECTDISTANCE RANGE</a> <a href="#">EXIFTAG_TAG_USERCOMMENT</a> <a href="#">EXIFTAG_TAG_WHITEBALANCE</a> <a href="#">EXIFTAG_TAG_XRESOLUTION</a> <a href="#">EXIFTAG_TAG_YCBRPOSITION</a> <a href="#">EXIFTAG_TAG_YRESOLUTION</a> <a href="#">getByteArray</a> <a href="#">getFileOffset</a> <a href="#">getInteger</a> <a href="#">getInteger1</a> <a href="#">getInteger2</a> <a href="#">getLong1</a> <a href="#">getLong2</a> <a href="#">getLongArray1</a> <a href="#">getLongArray2</a> <a href="#">getString</a> <a href="#">getTag</a> <a href="#">getType</a> <a href="#">getValueOffset</a> <a href="#">GPSTAG_TAG_ALTITUDE</a> <a href="#">GPSTAG_TAG_ALITUDEREF</a> <a href="#">GPSTAG_TAG_DOP</a> <a href="#">GPSTAG_TAG_LATITUDE</a> <a href="#">GPSTAG_TAG_LATITUDEREf</a> <a href="#">GPSTAG_TAG_LONGITUDE</a> <a href="#">GPSTAG_TAG_LONGITUDEREf</a> <a href="#">GPSTAG_TAG_MAPLINK</a> <a href="#">GPSTAG_TAG_MEASUREMODE</a> <a href="#">GPSTAG_TAG_SATELLITES</a> <a href="#">GPSTAG_TAG_STATUS</a> <a href="#">GPSTAG_TAG_TIMESTAMP</a> <a href="#">GPSTAG_TAG_VERSIONID</a> <a href="#">isGenericTag</a> <a href="#">isGPSTag</a> <a href="#">isTagUsedByGeneric</a> <a href="#">JExifTag</a> <a href="#">m_byteArray</a> <a href="#">m_count</a> <a href="#">m_fileOffset</a> <a href="#">m_integer1</a> <a href="#">m_integer2</a> <a href="#">m_long1</a> <a href="#">m_long2</a> <a href="#">m_longArray1</a> <a href="#">m_longArray2</a> <a href="#">m_string</a> <a href="#">m_tag</a> <a href="#">m_tagMask</a> <a href="#">m_type</a> <a href="#">m_valueOffset</a> <a href="#">read</a> <a href="#">TAGMASK_EXIF TAGMASK_GPS</a> <a href="#">toString</a>
target of relation	ComponentRealization <a href="#">Implementierung</a>
typedElements	Class <a href="#">JIfd</a> Operation <a href="#">findTag</a> <a href="#">getGradFromExifTag</a>
documentation	@author reiner

#### Property JExifTag::EXIFTAG\_TAG\_APERTUREVALUE

owner	<a href="#">JExifTag</a>
-------	--------------------------

#### Property JExifTag::EXIFTAG\_TAG\_ARTIST

owner	<a href="#">JExifTag</a>
-------	--------------------------

#### Property JExifTag::EXIFTAG\_TAG\_BRIGHTNESSVALUE

owner	<a href="#">JExifTag</a>
-------	--------------------------

#### Property JExifTag::EXIFTAG\_TAG\_CHANGEDATETIME

owner	<a href="#">JExifTag</a>
-------	--------------------------

#### Property JExifTag::EXIFTAG\_TAG\_COLORSPACE

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_COMPONENTSCCONFIG**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_COMPRESSEDBITSPERPIXEL**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_CONTRAST**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_COPYRIGHT**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_CUSTOM RENDERED**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_DESCRIPTION**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_DIGITALZOOMRATIO**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_DIGITIZEDDATETIME**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_EXIFIFD**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_EXIFVERSION**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_EXPOSUREBIASVALUE**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_EXPOSUREMODE**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_EXPOSUREPROGRAM**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_EXPOSURETIME**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_FILEDATETIME**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_FILESIZE**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_FILESOURCE**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_FLASH**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_FNUMBER**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_FOCALLENGTH**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_FOCALLENGTH35MM**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_FOCALPLANERESOLUTION**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_FOCALPLANERESOLUTIONUNIT**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_FOCALPLANEXRESOLUTION**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_FOCALPLANEYRESOLUTION**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_GAINCONTROL**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_GPSIFD**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_ISOSPEEDRATINGS**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_LIGHTSOURCE**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_MAKE**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_MAKERNOTE**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_MAXAPERTUREVALUE**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_METERINGMODE**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_MODEL**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_NAME**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_ORIENTATION**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_ORIGINALDATETIME**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_PIXELXDIMENSION**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_PIXELYDIMENSION**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_RESOLUTION**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_RESOLUTIONUNIT**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_SATURATION**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_SCENECAPTURETYPE**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_SCENETYPE**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_SENSINGMODE**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_SHARPNESS**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_SHUTERSPEEDVALUE**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_SIZE**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_SOFTWARE**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_SUBJECTDISTANCERANGE**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_USERCOMMENT**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_WHITEBALANCE**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_XRESOLUTION**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_YCBCRPOSITION**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::EXIFTAG\_TAG\_YRESOLUTION**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Operation **JExifTag::getByteArray**

owner	<a href="#">JExifTag</a>
parameter	name direction type multiplicity default <b>return return byte *</b>
ownedMember	<b>return</b>

Operation **JExifTag::getFileOffset**

owner	<a href="#">JExifTag</a>
parameter	name direction type multiplicity default <b>return return long</b>
ownedMember	<b>return</b>

Operation **JExifTag::getInteger**

owner	<a href="#">JExifTag</a>
parameter	name direction type multiplicity default <b>return return int</b>
ownedMember	<b>return</b>

**Operation JExifTag::getInteger1**

owner	<a href="#">JExifTag</a>
parameter	name direction type multiplicity default <b>return return int</b>
ownedMember	<b>return</b>

**Operation JExifTag::getInteger2**

owner	<a href="#">JExifTag</a>
parameter	name direction type multiplicity default <b>return return int</b>
ownedMember	<b>return</b>

**Operation JExifTag::getLong1**

owner	<a href="#">JExifTag</a>
parameter	name direction type multiplicity default <b>return return long</b>
ownedMember	<b>return</b>

**Operation JExifTag::getLong2**

owner	<a href="#">JExifTag</a>
parameter	name direction type multiplicity default <b>return return long</b>
ownedMember	<b>return</b>

**Operation JExifTag::getLongArray1**

owner	<a href="#">JExifTag</a>
parameter	name direction type multiplicity default <b>return return long *</b>
ownedMember	<b>return</b>

**Operation JExifTag::getLongArray2**

owner	<a href="#">JExifTag</a>
parameter	name direction type multiplicity default <b>return return long *</b>
ownedMember	<b>return</b>

**Operation JExifTag::getString**

owner	<a href="#">JExifTag</a>
parameter	name direction type multiplicity default <b>return return String</b>
ownedMember	<b>return</b>

**Operation JExifTag::getTag**

owner	<a href="#">JExifTag</a>
-------	--------------------------

parameter	name direction type multiplicity default <b>return return int</b>
ownedMember	<b>return</b>

#### Operation **JExifTag::getType**

owner	<b>JExifTag</b>
parameter	name direction type multiplicity default <b>return return int</b>
ownedMember	<b>return</b>

#### Operation **JExifTag::getValueOffset**

owner	<b>JExifTag</b>
parameter	name direction type multiplicity default <b>return return long</b>
ownedMember	<b>return</b>

#### Property **JExifTag::GPSTAG\_TAG\_ALTITUDE**

owner	<b>JExifTag</b>
-------	-----------------

#### Property **JExifTag::GPSTAG\_TAG\_ALTITUDEREF**

owner	<b>JExifTag</b>
-------	-----------------

#### Property **JExifTag::GPSTAG\_TAG\_DOP**

owner	<b>JExifTag</b>
-------	-----------------

#### Property **JExifTag::GPSTAG\_TAG\_LATITUDE**

owner	<b>JExifTag</b>
-------	-----------------

#### Property **JExifTag::GPSTAG\_TAG\_LATITUDEREF**

owner	<b>JExifTag</b>
-------	-----------------

#### Property **JExifTag::GPSTAG\_TAG\_LONGITUDE**

owner	<b>JExifTag</b>
-------	-----------------

#### Property **JExifTag::GPSTAG\_TAG\_LONGITUDEREF**

owner	<b>JExifTag</b>
-------	-----------------

#### Property **JExifTag::GPSTAG\_TAG\_MAPLINK**

owner	<b>JExifTag</b>
-------	-----------------

Property **JExifTag::GPSTAG\_TAG\_MEASUREMODE**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::GPSTAG\_TAG\_SATELLITES**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::GPSTAG\_TAG\_STATUS**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::GPSTAG\_TAG\_TIMESTAMP**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::GPSTAG\_TAG\_VERSIONID**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Operation **JExifTag::isGenericTag**

owner	<a href="#">JExifTag</a>
parameter	name direction type multiplicity default <b>tag</b> in int return return boolean
ownedMember	<b>return tag</b>

Operation **JExifTag::isGPSTag**

owner	<a href="#">JExifTag</a>
parameter	name direction type multiplicity default <b>tag</b> in int return return boolean
ownedMember	<b>return tag</b>

Operation **JExifTag::isTagUsedByGeneric**

owner	<a href="#">JExifTag</a>
parameter	name direction type multiplicity default <b>tag</b> in int return return boolean
ownedMember	<b>return tag</b>

Operation **JExifTag::JExifTag**

owner	<a href="#">JExifTag</a>
parameter	name direction type multiplicity default <b>tagMask</b> in int
ownedMember	<b>tagMask</b>
documentation	Creates a new instance of JExifTag

Property **JExifTag::m\_byteArray**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::m\_count**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::m\_fileOffset**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::m\_integer1**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::m\_integer2**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::m\_long1**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::m\_long2**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::m\_longArray1**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::m\_longArray2**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::m\_string**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::m\_tag**

owner	<a href="#">JExifTag</a>
-------	--------------------------

Property **JExifTag::m\_tagMask**

owner	<a href="#">JExifTag</a>
-------	--------------------------

**Property JExifTag::m\_type**

owner	<a href="#">JExifTag</a>
-------	--------------------------

**Property JExifTag::m\_valueOffset**

owner	<a href="#">JExifTag</a>
-------	--------------------------

**Operation JExifTag::read**

owner	<a href="#">JExifTag</a>
parameter	name direction type multiplicity default <b>file in RandomAccessFile bIntel in boolean tiffOffset in long return return boolean</b>
ownedMember	<b>bIntel file return tiffOffset</b>

**Property JExifTag::TAGMASK\_EXIF**

owner	<a href="#">JExifTag</a>
-------	--------------------------

**Property JExifTag::TAGMASK\_GPS**

owner	<a href="#">JExifTag</a>
-------	--------------------------

**Operation JExifTag::toString**

owner	<a href="#">JExifTag</a>
parameter	name direction type multiplicity default <b>return return String</b>
ownedMember	<b>return</b>

## Class JFileHelper

diagram	
owner	<b>files</b>
ownedMember	<b>copyFile JFileHelper</b>
target of relation	ComponentRealization <b>Implementierung</b>
documentation	@author reiner

### Operation JFileHelper::copyFile

owner	<b>JFileHelper</b>
parameter	name direction type multiplicity default <b>src in File</b> <b>target in File</b> <b>return return boolean</b>
ownedMember	<b>return src target</b>

### Operation JFileHelper::JFileHelper

owner	<b>JFileHelper</b>
documentation	Creates a new instance of JFileHelper

## Class JFileHelper

diagram	
owner	<b>files</b>
ownedMember	<b>copyFile JFileHelper</b>
target of relation	ComponentRealization <b>Implementierung</b>
documentation	@author reiner

### Operation JFileHelper::copyFile

owner	<b>JFileHelper</b>
parameter	name direction type multiplicity default <b>src in File</b> <b>target in File</b> <b>return return boolean</b>
ownedMember	<b>return src target</b>

### Operation JFileHelper::JFileHelper

owner	<b>JFileHelper</b>
documentation	Creates a new instance of JFileHelper

## Class JFileSaver

diagram	A UML class diagram showing a class named "JFileSaver". Below the class name is a constructor symbol (a small square with a diamond inside) followed by the signature "<<constructor>> JFileSaver(in image: BufferedImage)".
owner	<a href="#">utils</a>
ownedMember	<a href="#">JFileSaver</a>
target of relation	ComponentRealization <a href="#">Implementierung</a>

## Operation JFileSaver::JFileSaver

owner	<a href="#">JFileSaver</a>
parameter	name direction type multiplicity default <b>image in BufferedImage</b>
ownedMember	<b>image</b>
documentation	Start a JFileDialog which lets the user save the specified image on a storage device @param image image which is to be saved

## Class JHistogram2D

diagram	 <pre> classDiagram     JHistogram2D &lt; -- M&gt;Histogram2DModel     JHistogram2D "1" --&gt; "1" M&gt;Histogram2DModel     JHistogram2D &lt;&lt;final&gt;&gt; serialVersionUID: long = 1L     JHistogram2D &lt;&lt;constructor&gt;&gt; JHistogram2D(in model:M)     JHistogram2D &lt;&lt;annotation&gt;&gt; generateContent(): void   </pre>	<b>generateContent() (Operation)</b> <<annotations>> modifiers = Override
hierarchy		
owner	<a href="#">diagrams</a>	
template parameters	name kind constrainingClassifier default <b>M Class Histogram2DModel</b>	
ownedMember	<a href="#">generateContent JHistogram2D serialVersionUID</a>	
general	<a href="#">JAbstract2DDiagram&lt;M&gt;M&gt;</a>	
target of relation		ComponentRealization <a href="#">Implementierung</a>
documentation	This class implements how the Histogram2DModel is to be drawn.	
	@author David Kaufman	
	@param <M>	

## Operation JHistogram2D::generateContent

owner	<b>JHistogram2D</b>
parameter	name direction type multiplicity default <b>return return void</b>
ownedMember	<b>return</b>

## Operation JHistogram2D::JHistogram2D

owner	<b>JHistogram2D</b>
parameter	name direction type multiplicity default <b>model in M</b>
ownedMember	<b>model</b>
documentation	<p>Constructor</p> <p><b>@param abstractModel</b> the model from which the drawing information is taken</p>

## Property JHistogram2D::serialVersionUID

owner **JHistogram2D**



## Class JHistogram3D

diagram	<p>JHistogram3D</p> <p><code>&lt;&lt;final&gt;&gt; serialVersionUID:long=1L</code></p> <p><code>&lt;&lt;constructor&gt;&gt; JHistogram3D(in model:M)</code></p> <p><code>&lt;&lt;annotation&gt;&gt; generateContent():void</code></p> <p><code>generateContent() (Operation) modifiers = Override</code></p>
hierarchy	<pre> classDiagram     JAbstract3DDiagram&lt;M&gt; &lt; -- JHistogram3D   </pre>
owner	<a href="#">diagrams</a>
template parameters	name kind constrainingClassifier default <b>M Class Histogram3DModel</b>
ownedMember	<a href="#">generateContent JHistogram3D serialVersionUID</a>
general	<a href="#">JAbstract3DDiagram&lt;M&gt;</a>
target of relation	ComponentRealization <a href="#">Implementierung</a>
documentation	<p>This class implements how the Histogram3DModel is to be drawn.</p> <p>@author David Kaufman</p> <p>@param &lt;M&gt;</p>

## Operation JHistogram3D::generateContent

owner	<a href="#">JHistogram3D</a>
parameter	name direction type multiplicity default <b>return return void</b>
ownedMember	<b>return</b>

## Operation JHistogram3D::JHistogram3D

owner	<a href="#">JHistogram3D</a>
parameter	name direction type multiplicity default <b>model in M</b>
ownedMember	<b>model</b>
documentation	<p>Constructor</p> <p>@param abstractModel the model from which the drawing information is taken</p>

## Property JHistogram3D::serialVersionUID

owner	<a href="#">JHistogram3D</a>
-------	------------------------------



## Class JIfd

diagram

JIfd	
<ul style="list-style-type: none"><li>■ m_offset:long</li><li>■ m_count:int</li><li>■ m_nextIfdOffset:long</li><li>■ m_bIsInIef:boolean</li><li>■ m_tagArray:ArrayList&lt;E&gt;&gt;JExifTag&gt;</li><li>■ &lt;&lt;final&gt;&gt; GRAD INVALID:float=999.0f</li><li>■ &lt;&lt;constructor&gt;&gt; JIfd()</li><li>■ getDateFromString(in str:String):Date</li><li>■ isTagPresent(in tag:int):boolean</li><li>■ getMake():String</li><li>■ getArtist():String</li><li>■ getDescription():String</li><li>■ getModel():String</li><li>■ getOrientation():int</li><li>■ getXResolution():float</li><li>■ getYResolution():float</li><li>■ getResolutionUnit():int</li><li>■ getSoftware():String</li><li>■ getChangeDateTime():String</li><li>■ getYCbCrPosition():int</li><li>■ getCopyright():String</li><li>■ getExposureTime(in str:String[]):int</li><li>■ getFNumber(in str:String[]):float</li><li>■ getExposureProgram():int</li><li>■ getISOSpeedRatings():int</li><li>■ getExifVersion():String</li><li>■ getOriginalDateTime():String</li><li>■ getDigitizedDateTime():String</li><li>■ getComponentsConfig():String</li><li>■ getCompressedBitsPerPixel():float</li><li>■ getShutterSpeedValue():float</li><li>■ getApertureValue():float</li><li>■ getBrightnessValue():float</li><li>■ getExposureBiasValue():float</li><li>■ getMaxApertureValue():float</li><li>■ getMeteringMode():int</li><li>■ getLightSource():int</li><li>■ getFlash():int</li><li>■ getFocalLength():float</li><li>■ getColorSpace():int</li><li>■ getPixelXDimension():int</li><li>■ getPixelYDimension():int</li><li>■ getFocalPlaneXResolution():float</li><li>■ getFocalPlaneYResolution():float</li><li>■ getFocalPlaneResolutionUnit():int</li><li>■ getSensingMode():int</li><li>■ getCustomRendered():int</li><li>■ getExposureMode():int</li><li>■ getWhiteBalance():int</li><li>■ getDigitalZoomRatio():float</li><li>■ getSceneType():int</li><li>■ getSceneCaptureType():int</li><li>■ getSharpness():int</li><li>■ getSubjectDistanceRange():int</li><li>■ getFocalLength35mm():int</li><li>■ getGainControl():int</li><li>■ getContrast():int</li><li>■ getSaturation():int</li><li>■ getFileSource():int</li><li>■ getGPSVersionID():String</li><li>■ getGPSLatitudeRef():String</li><li>■ getGradFromExifTag(in tag:JExifTag):float</li><li>■ getGPSLatitude():float</li><li>■ getGPSLongitudeRef():String</li><li>■ getGPSLongitude():float</li><li>■ getGPSAltitudeRef():int</li><li>■ getGPSAltitude():float</li><li>■ getGPSSatellites():String</li><li>■ getGPSStatus():String</li><li>■ getGPSMeasureMode():String</li><li>■ getGPSTimestamp():Date</li><li>■ findTag(in tag:int):JExifTag</li><li>■ getCount():int</li><li>■ read(in file:RandomAccessFile, in bIsInIef:boolean, in ifdOffset:long):boolean</li><li>■ readIETag(in file:RandomAccessFile, in bIsInIef:boolean, in ifdOffset:long, in tagMask:int):boolean</li><li>■ updateTag(in file:RandomAccessFile, in tag:int, in value:int):boolean</li></ul>	

hierarchy	<pre> classDiagram     class JIffd {         &lt;&lt;JIffd&gt;&gt;     }     class JIffdData {         &lt;&lt;JIffdData&gt;&gt;     }     JIffd "1" --&gt; "1" JIffdData   </pre>
owner	<b>jexifviewer</b>
ownedMember	<code>findTag getApertureValue getArtist getBrightnessValue getChangeDateTime getColorSpace getComponentsConfig getCompressedBitsPerPixel getContrast getCopyright getCount getCustomRendered getDateFromString getDescription getDigitalZoomRatio getDigitizedDateTime getExifVersion getExposureBiasValue getExposureMode getExposureProgram getExposureTime getFileSource getFlash getFNumber getFocalLength getFocalLength35mm getFocalPlaneResolutionUnit getFocalPlaneXResolution getFocalPlaneYResolution getGainControl getGPSAltitude getGPSAltitudeRef getGPSLatitude getGPSLatitudeRef getGPSLongitude getGPSLongitudeRef getGPSMeasureMode getGPSSatellites getGPSStatus getGPSTimestamp getGPSVersionID getGradFromExifTag getISOSpeedRatings getLightSource getMake getMaxApertureValue getMeteringMode getModel getOrientation getOriginalDateTime getPixelXDimension getPixelYDimension getResolutionUnit getSaturation getSceneCaptureType getSceneType getSensingMode getSharpness getShutterSpeedValue getSoftware getSubjectDistanceRange getWhiteBalance getXResolution getYCbCrPosition getYResolution GRAD_INVALID isTagPresent JIffd m_bIsIntel m_count m_nextIffdOffset m_offset m_tagArray read readIFDTag updateTag</code>
specific	<b>JIffdData</b>
target of relation	ComponentRealization <a href="#">Implementierung</a>
documentation	@author reiner

#### Operation **JIffd::findTag**

owner	<b>JIffd</b>
parameter	name direction type multiplicity default <b>tag</b> in int return return <b>JExifTag</b>
ownedMember	<b>return tag</b>

#### Operation **JIffd::getApertureValue**

owner	<b>JIffd</b>
parameter	name direction type multiplicity default <b>return return float</b>
ownedMember	<b>return</b>

#### Operation **JIffd::getArtist**

owner	<b>JIffd</b>
parameter	name direction type multiplicity default <b>return return String</b>
ownedMember	<b>return</b>

#### Operation **JIffd::getBrightnessValue**

owner	<b>JIffd</b>
parameter	name direction type multiplicity default <b>return return float</b>
ownedMember	<b>return</b>

**Operation JIfd::getChangeDateTime**

owner	JIfd
parameter	name direction type multiplicity default return return String
ownedMember	return

**Operation JIfd::getColorSpace**

owner	JIfd
parameter	name direction type multiplicity default return return int
ownedMember	return

**Operation JIfd::getComponentsConfig**

owner	JIfd
parameter	name direction type multiplicity default return return String
ownedMember	return

**Operation JIfd::getCompressedBitsPerPixel**

owner	JIfd
parameter	name direction type multiplicity default return return float
ownedMember	return

**Operation JIfd::getContrast**

owner	JIfd
parameter	name direction type multiplicity default return return int
ownedMember	return

**Operation JIfd::getCopyright**

owner	JIfd
parameter	name direction type multiplicity default return return String
ownedMember	return

**Operation JIfd::getCount**

owner	JIfd
parameter	name direction type multiplicity default return return int
ownedMember	return

**Operation JIfd::getCustomRendered**

owner	<a href="#">JIfd</a>
parameter	name direction type multiplicity default <b>return return int</b>
ownedMember	<b>return</b>

**Operation JIfd::getDateFromString**

owner	<a href="#">JIfd</a>
parameter	name direction type multiplicity default <b>str in String return return Date</b>
ownedMember	<b>return str</b>

**Operation JIfd::getDescription**

owner	<a href="#">JIfd</a>
parameter	name direction type multiplicity default <b>return return String</b>
ownedMember	<b>return</b>

**Operation JIfd::getDigitalZoomRatio**

owner	<a href="#">JIfd</a>
parameter	name direction type multiplicity default <b>return return float</b>
ownedMember	<b>return</b>

**Operation JIfd::getDigitizedDateTime**

owner	<a href="#">JIfd</a>
parameter	name direction type multiplicity default <b>return return String</b>
ownedMember	<b>return</b>

**Operation JIfd::getExifVersion**

owner	<a href="#">JIfd</a>
parameter	name direction type multiplicity default <b>return return String</b>
ownedMember	<b>return</b>

**Operation JIfd::getExposureBiasValue**

owner	<a href="#">JIfd</a>
parameter	name direction type multiplicity default <b>return return float</b>
ownedMember	<b>return</b>

**Operation JIfd::getExposureMode**

owner	<a href="#">JIfd</a>
-------	----------------------

parameter	name direction type multiplicity default <b>return return int</b>
ownedMember	<b>return</b>

#### Operation **JIfd::getExposureProgram**

owner	<b>JIfd</b>
parameter	name direction type multiplicity default <b>return return int</b>
ownedMember	<b>return</b>

#### Operation **JIfd::getExposureTime**

owner	<b>JIfd</b>
parameter	name direction type multiplicity default <b>str in String * return return int</b>
ownedMember	<b>return str</b>
documentation	returns the exposure time in $\frac{1}{2}$ sec @param str string representation @return exposure time in $\frac{1}{2}$ sec

#### Operation **JIfd::getFileSource**

owner	<b>JIfd</b>
parameter	name direction type multiplicity default <b>return return int</b>
ownedMember	<b>return</b>

#### Operation **JIfd::getFlash**

owner	<b>JIfd</b>
parameter	name direction type multiplicity default <b>return return int</b>
ownedMember	<b>return</b>

#### Operation **JIfd::getFNumber**

owner	<b>JIfd</b>
parameter	name direction type multiplicity default <b>str in String * return return float</b>
ownedMember	<b>return str</b>

#### Operation **JIfd::getFocalLength**

owner	<b>JIfd</b>
parameter	name direction type multiplicity default <b>return return float</b>
ownedMember	<b>return</b>

**Operation JIfd::getFocalLength35mm**

owner	<a href="#">JIfd</a>
parameter	name direction type multiplicity default <b>return return int</b>
ownedMember	<b>return</b>

**Operation JIfd::getFocalPlaneResolutionUnit**

owner	<a href="#">JIfd</a>
parameter	name direction type multiplicity default <b>return return int</b>
ownedMember	<b>return</b>

**Operation JIfd::getFocalPlaneXResolution**

owner	<a href="#">JIfd</a>
parameter	name direction type multiplicity default <b>return return float</b>
ownedMember	<b>return</b>

**Operation JIfd::getFocalPlaneYResolution**

owner	<a href="#">JIfd</a>
parameter	name direction type multiplicity default <b>return return float</b>
ownedMember	<b>return</b>

**Operation JIfd::getGainControl**

owner	<a href="#">JIfd</a>
parameter	name direction type multiplicity default <b>return return int</b>
ownedMember	<b>return</b>

**Operation JIfd::getGPSAltitude**

owner	<a href="#">JIfd</a>
parameter	name direction type multiplicity default <b>return return float</b>
ownedMember	<b>return</b>

**Operation JIfd::getGPSAltitudeRef**

owner	<a href="#">JIfd</a>
parameter	name direction type multiplicity default <b>return return int</b>
ownedMember	<b>return</b>

**Operation JIfd::getGPSLatitude**

owner	<a href="#">JIfd</a>
-------	----------------------

parameter	name direction type multiplicity default <b>return return float</b>
ownedMember	<b>return</b>

#### Operation **JIfd::getGPSLatitudeRef**

owner	<b>JIfd</b>
parameter	name direction type multiplicity default <b>return return String</b>
ownedMember	<b>return</b>

#### Operation **JIfd::getGPSLongitude**

owner	<b>JIfd</b>
parameter	name direction type multiplicity default <b>return return float</b>
ownedMember	<b>return</b>

#### Operation **JIfd::getGPSLongitudeRef**

owner	<b>JIfd</b>
parameter	name direction type multiplicity default <b>return return String</b>
ownedMember	<b>return</b>

#### Operation **JIfd::getGPSMeasureMode**

owner	<b>JIfd</b>
parameter	name direction type multiplicity default <b>return return String</b>
ownedMember	<b>return</b>

#### Operation **JIfd::getGPSSatellites**

owner	<b>JIfd</b>
parameter	name direction type multiplicity default <b>return return String</b>
ownedMember	<b>return</b>

#### Operation **JIfd::getGPSStatus**

owner	<b>JIfd</b>
parameter	name direction type multiplicity default <b>return return String</b>
ownedMember	<b>return</b>

#### Operation **JIfd::getGPSTimestamp**

owner	<b>JIfd</b>
parameter	name direction type multiplicity default <b>return return Date</b>
ownedMember	<b>return</b>

#### Operation **JIfd::getGPSVersionID**

owner	<b>JIfd</b>
parameter	name direction type multiplicity default <b>return return String</b>
ownedMember	<b>return</b>

#### Operation **JIfd::getGradFromExifTag**

owner	<b>JIfd</b>
parameter	name direction type multiplicity default <b>tag in JExifTag return return float</b>
ownedMember	<b>return tag</b>

#### Operation **JIfd::getISOSpeedRatings**

owner	<b>JIfd</b>
parameter	name direction type multiplicity default <b>return return int</b>
ownedMember	<b>return</b>

#### Operation **JIfd::getLightSource**

owner	<b>JIfd</b>
parameter	name direction type multiplicity default <b>return return int</b>
ownedMember	<b>return</b>

#### Operation **JIfd::getMake**

owner	<b>JIfd</b>
parameter	name direction type multiplicity default <b>return return String</b>
ownedMember	<b>return</b>

#### Operation **JIfd::getMaxApertureValue**

owner	<b>JIfd</b>
parameter	name direction type multiplicity default <b>return return float</b>
ownedMember	<b>return</b>

#### Operation **JIfd::getMeteringMode**

owner	<b>JIfd</b>
parameter	name direction type multiplicity default <b>return return int</b>
ownedMember	<b>return</b>

#### Operation **JIfd::getModel**

owner	<b>JIfd</b>
parameter	name direction type multiplicity default <b>return return String</b>
ownedMember	<b>return</b>

#### Operation **JIfd::getOrientation**

owner	<b>JIfd</b>
parameter	name direction type multiplicity default <b>return return int</b>
ownedMember	<b>return</b>

#### Operation **JIfd::getOriginalDateTime**

owner	<b>JIfd</b>
parameter	name direction type multiplicity default <b>return return String</b>
ownedMember	<b>return</b>

#### Operation **JIfd::getPixelXDimension**

owner	<b>JIfd</b>
parameter	name direction type multiplicity default <b>return return int</b>
ownedMember	<b>return</b>

#### Operation **JIfd::getPixelYDimension**

owner	<b>JIfd</b>
parameter	name direction type multiplicity default <b>return return int</b>
ownedMember	<b>return</b>

#### Operation **JIfd::getResolutionUnit**

owner	<b>JIfd</b>
parameter	name direction type multiplicity default <b>return return int</b>
ownedMember	<b>return</b>

#### Operation **JIfd::getSaturation**

owner	<b>JIfd</b>
parameter	name direction type multiplicity default <b>return return int</b>
ownedMember	<b>return</b>

#### Operation **JIfd::getSceneCaptureType**

owner	<b>JIfd</b>
-------	-------------

parameter	name direction type multiplicity default <b>return return int</b>
ownedMember	<b>return</b>

#### Operation **JIfd::getSceneType**

owner	<b>JIfd</b>
parameter	name direction type multiplicity default <b>return return int</b>
ownedMember	<b>return</b>

#### Operation **JIfd::getSensingMode**

owner	<b>JIfd</b>
parameter	name direction type multiplicity default <b>return return int</b>
ownedMember	<b>return</b>

#### Operation **JIfd::getSharpness**

owner	<b>JIfd</b>
parameter	name direction type multiplicity default <b>return return int</b>
ownedMember	<b>return</b>

#### Operation **JIfd::getShutterSpeedValue**

owner	<b>JIfd</b>
parameter	name direction type multiplicity default <b>return return float</b>
ownedMember	<b>return</b>

#### Operation **JIfd::getSoftware**

owner	<b>JIfd</b>
parameter	name direction type multiplicity default <b>return return String</b>
ownedMember	<b>return</b>

#### Operation **JIfd::getSubjectDistanceRange**

owner	<b>JIfd</b>
parameter	name direction type multiplicity default <b>return return int</b>
ownedMember	<b>return</b>

#### Operation **JIfd::getWhiteBalance**

owner	<b>JIfd</b>
parameter	name direction type multiplicity default <b>return return int</b>
ownedMember	<b>return</b>

**Operation JIfd::getXResolution**

owner	JIfd
parameter	name direction type multiplicity default <b>return return float</b>
ownedMember	<b>return</b>

**Operation JIfd::getYCbCrPosition**

owner	JIfd
parameter	name direction type multiplicity default <b>return return int</b>
ownedMember	<b>return</b>

**Operation JIfd::getYResolution**

owner	JIfd
parameter	name direction type multiplicity default <b>return return float</b>
ownedMember	<b>return</b>

**Property JIfd::GRAD\_INVALID**

owner	JIfd
-------	------

**Operation JIfd::isTagPresent**

owner	JIfd
parameter	name direction type multiplicity default <b>tag in int return return boolean</b>
ownedMember	<b>return tag</b>

**Operation JIfd::JIfd**

owner	JIfd
documentation	Creates a new instance of JIfd

**Property JIfd::m\_bIsIntel**

owner	JIfd
-------	------

**Property JIfd::m\_count**

owner	JIfd
-------	------

**Property JIfd::m\_nextIfdOffset**

owner	JIfd
-------	------

**Property JIfd::m\_offset**

owner	<a href="#">JIfd</a>
-------	----------------------

**Property JIfd::m\_tagArray**

owner	<a href="#">JIfd</a>
-------	----------------------

**Operation JIfd::read**

owner	<a href="#">JIfd</a>
parameter	name direction type multiplicity default <b>file in RandomAccessFile bintel in boolean tiffOffset in long return return boolean</b>
ownedMember	<b>bintel file return tiffOffset</b>

**Operation JIfd::readIFDTag**

owner	<a href="#">JIfd</a>
parameter	name direction type multiplicity default <b>file in RandomAccessFile bintel in boolean tiffOffset in long tagMask in int return return boolean</b>
ownedMember	<b>bintel file return tagMask tiffOffset</b>

**Operation JIfd::updateTag**

owner	<a href="#">JIfd</a>
parameter	name direction type multiplicity default <b>file in RandomAccessFile tag in int value in int return return boolean</b>
ownedMember	<b>file return tag value</b>

## Class JIfd

diagram

JIfd	
<ul style="list-style-type: none"><li>■ m_offset:long</li><li>■ m_count:int</li><li>■ m_nextIfdOffset:long</li><li>■ m_bIsInIef:boolean</li><li>■ m_tagArray:ArrayList&lt;E&gt;&gt;JExifTag&gt;</li><li>■ &lt;&lt;final&gt;&gt; GRAD INVALID:float=999.0f</li><li>■ &lt;&lt;constructor&gt;&gt; JIfd()</li><li>■ getDateFromString(in str:String):Date</li><li>■ isTagPresent(in tag:int):boolean</li><li>■ getMake():String</li><li>■ getArtist():String</li><li>■ getDescription():String</li><li>■ getModel():String</li><li>■ getOrientation():int</li><li>■ getXResolution():float</li><li>■ getYResolution():float</li><li>■ getResolutionUnit():int</li><li>■ getSoftware():String</li><li>■ getChangeDateTime():String</li><li>■ getYCbCrPosition():int</li><li>■ getCopyright():String</li><li>■ getExposureTime(in str:String[]):int</li><li>■ getFNumber(in str:String[]):float</li><li>■ getExposureProgram():int</li><li>■ getISOSpeedRatings():int</li><li>■ getExifVersion():String</li><li>■ getOriginalDateTime():String</li><li>■ getDigitizedDateTime():String</li><li>■ getComponentsConfig():String</li><li>■ getCompressedBitsPerPixel():float</li><li>■ getShutterSpeedValue():float</li><li>■ getApertureValue():float</li><li>■ getBrightnessValue():float</li><li>■ getExposureBiasValue():float</li><li>■ getMaxApertureValue():float</li><li>■ getMeteringMode():int</li><li>■ getLightSource():int</li><li>■ getFlash():int</li><li>■ getFocalLength():float</li><li>■ getColorSpace():int</li><li>■ getPixelXDimension():int</li><li>■ getPixelYDimension():int</li><li>■ getFocalPlaneXResolution():float</li><li>■ getFocalPlaneYResolution():float</li><li>■ getFocalPlaneResolutionUnit():int</li><li>■ getSensingMode():int</li><li>■ getCustomRendered():int</li><li>■ getExposureMode():int</li><li>■ getWhiteBalance():int</li><li>■ getDigitalZoomRatio():float</li><li>■ getSceneType():int</li><li>■ getSceneCaptureType():int</li><li>■ getSharpness():int</li><li>■ getSubjectDistanceRange():int</li><li>■ getFocalLength35mm():int</li><li>■ getGainControl():int</li><li>■ getContrast():int</li><li>■ getSaturation():int</li><li>■ getFileSource():int</li><li>■ getGPSVersionID():String</li><li>■ getGPSLatitudeRef():String</li><li>■ getGradFromExifTag(in tag:JExifTag):float</li><li>■ getGPSLatitude():float</li><li>■ getGPSLongitudeRef():String</li><li>■ getGPSLongitude():float</li><li>■ getGPSAltitudeRef():int</li><li>■ getGPSAltitude():float</li><li>■ getGPSSatellites():String</li><li>■ getGPSStatus():String</li><li>■ getGPSMeasureMode():String</li><li>■ getGPSTimestamp():Date</li><li>■ findTag(in tag:int):JExifTag</li><li>■ getCount():int</li><li>■ read(in file:RandomAccessFile, in bIsInIef:boolean, in ifdOffset:long):boolean</li><li>■ readIETag(in file:RandomAccessFile, in bIsInIef:boolean, in ifdOffset:long, in tagMask:int):boolean</li><li>■ updateTag(in file:RandomAccessFile, in tag:int, in value:int):boolean</li></ul>	

hierarchy	<pre> classDiagram     class JIffd {         &lt;&lt;JIffd&gt;&gt;     }     class JIffdData {         &lt;&lt;JIffdData&gt;&gt;     }     JIffd "1" --&gt; "1" JIffdData   </pre>
owner	<b>jexifviewer</b>
ownedMember	<code>findTag getApertureValue getArtist getBrightnessValue getChangeDateTime getColorSpace getComponentsConfig getCompressedBitsPerPixel getContrast getCopyright getCount getCustomRendered getDateFromString getDescription getDigitalZoomRatio getDigitizedDateTime getExifVersion getExposureBiasValue getExposureMode getExposureProgram getExposureTime getFileSource getFlash getFNumber getFocalLength getFocalLength35mm getFocalPlaneResolutionUnit getFocalPlaneXResolution getFocalPlaneYResolution getGainControl getGPSAltitude getGPSAltitudeRef getGPSLatitude getGPSLatitudeRef getGPSLongitude getGPSLongitudeRef getGPSMeasureMode getGPSSatellites getGPSStatus getGPSTimestamp getGPSVersionID getGradFromExifTag getISOSpeedRatings getLightSource getMake getMaxApertureValue getMeteringMode getModel getOrientation getOriginalDateTime getPixelXDimension getPixelYDimension getResolutionUnit getSaturation getSceneCaptureType getSceneType getSensingMode getSharpness getShutterSpeedValue getSoftware getSubjectDistanceRange getWhiteBalance getXResolution getYCbCrPosition getYResolution GRAD_INVALID isTagPresent JIffd m_bIsIntel m_count m_nextIffdOffset m_offset m_tagArray read readIFDTag updateTag</code>
specific	<b>JIffdData</b>
target of relation	ComponentRealization <a href="#">Implementierung</a>
documentation	@author reiner

#### Operation **JIffd::findTag**

owner	<b>JIffd</b>
parameter	name direction type multiplicity default <b>tag</b> in int return return <b>JExifTag</b>
ownedMember	<b>return tag</b>

#### Operation **JIffd::getApertureValue**

owner	<b>JIffd</b>
parameter	name direction type multiplicity default <b>return return float</b>
ownedMember	<b>return</b>

#### Operation **JIffd::getArtist**

owner	<b>JIffd</b>
parameter	name direction type multiplicity default <b>return return String</b>
ownedMember	<b>return</b>

#### Operation **JIffd::getBrightnessValue**

owner	<b>JIffd</b>
parameter	name direction type multiplicity default <b>return return float</b>
ownedMember	<b>return</b>

**Operation JIfd::getChangeDateTime**

owner	JIfd
parameter	name direction type multiplicity default return return String
ownedMember	return

**Operation JIfd::getColorSpace**

owner	JIfd
parameter	name direction type multiplicity default return return int
ownedMember	return

**Operation JIfd::getComponentsConfig**

owner	JIfd
parameter	name direction type multiplicity default return return String
ownedMember	return

**Operation JIfd::getCompressedBitsPerPixel**

owner	JIfd
parameter	name direction type multiplicity default return return float
ownedMember	return

**Operation JIfd::getContrast**

owner	JIfd
parameter	name direction type multiplicity default return return int
ownedMember	return

**Operation JIfd::getCopyright**

owner	JIfd
parameter	name direction type multiplicity default return return String
ownedMember	return

**Operation JIfd::getCount**

owner	JIfd
parameter	name direction type multiplicity default return return int
ownedMember	return

**Operation JIfd::getCustomRendered**

owner	<a href="#">JIfd</a>
parameter	name direction type multiplicity default <b>return return int</b>
ownedMember	<b>return</b>

**Operation JIfd::getDateFromString**

owner	<a href="#">JIfd</a>
parameter	name direction type multiplicity default <b>str in String return return Date</b>
ownedMember	<b>return str</b>

**Operation JIfd::getDescription**

owner	<a href="#">JIfd</a>
parameter	name direction type multiplicity default <b>return return String</b>
ownedMember	<b>return</b>

**Operation JIfd::getDigitalZoomRatio**

owner	<a href="#">JIfd</a>
parameter	name direction type multiplicity default <b>return return float</b>
ownedMember	<b>return</b>

**Operation JIfd::getDigitizedDateTime**

owner	<a href="#">JIfd</a>
parameter	name direction type multiplicity default <b>return return String</b>
ownedMember	<b>return</b>

**Operation JIfd::getExifVersion**

owner	<a href="#">JIfd</a>
parameter	name direction type multiplicity default <b>return return String</b>
ownedMember	<b>return</b>

**Operation JIfd::getExposureBiasValue**

owner	<a href="#">JIfd</a>
parameter	name direction type multiplicity default <b>return return float</b>
ownedMember	<b>return</b>

**Operation JIfd::getExposureMode**

owner	<a href="#">JIfd</a>
-------	----------------------

parameter	name direction type multiplicity default <b>return return int</b>
ownedMember	<b>return</b>

#### Operation **JIfd::getExposureProgram**

owner	<b>JIfd</b>
parameter	name direction type multiplicity default <b>return return int</b>
ownedMember	<b>return</b>

#### Operation **JIfd::getExposureTime**

owner	<b>JIfd</b>
parameter	name direction type multiplicity default <b>str in String * return return int</b>
ownedMember	<b>return str</b>
documentation	returns the exposure time in $\frac{1}{2}$ sec @param str string representation @return exposure time in $\frac{1}{2}$ sec

#### Operation **JIfd::getFileSource**

owner	<b>JIfd</b>
parameter	name direction type multiplicity default <b>return return int</b>
ownedMember	<b>return</b>

#### Operation **JIfd::getFlash**

owner	<b>JIfd</b>
parameter	name direction type multiplicity default <b>return return int</b>
ownedMember	<b>return</b>

#### Operation **JIfd::getFNumber**

owner	<b>JIfd</b>
parameter	name direction type multiplicity default <b>str in String * return return float</b>
ownedMember	<b>return str</b>

#### Operation **JIfd::getFocalLength**

owner	<b>JIfd</b>
parameter	name direction type multiplicity default <b>return return float</b>
ownedMember	<b>return</b>

**Operation JIfd::getFocalLength35mm**

owner	<a href="#">JIfd</a>
parameter	name direction type multiplicity default <b>return return int</b>
ownedMember	<b>return</b>

**Operation JIfd::getFocalPlaneResolutionUnit**

owner	<a href="#">JIfd</a>
parameter	name direction type multiplicity default <b>return return int</b>
ownedMember	<b>return</b>

**Operation JIfd::getFocalPlaneXResolution**

owner	<a href="#">JIfd</a>
parameter	name direction type multiplicity default <b>return return float</b>
ownedMember	<b>return</b>

**Operation JIfd::getFocalPlaneYResolution**

owner	<a href="#">JIfd</a>
parameter	name direction type multiplicity default <b>return return float</b>
ownedMember	<b>return</b>

**Operation JIfd::getGainControl**

owner	<a href="#">JIfd</a>
parameter	name direction type multiplicity default <b>return return int</b>
ownedMember	<b>return</b>

**Operation JIfd::getGPSAltitude**

owner	<a href="#">JIfd</a>
parameter	name direction type multiplicity default <b>return return float</b>
ownedMember	<b>return</b>

**Operation JIfd::getGPSAltitudeRef**

owner	<a href="#">JIfd</a>
parameter	name direction type multiplicity default <b>return return int</b>
ownedMember	<b>return</b>

**Operation JIfd::getGPSLatitude**

owner	<a href="#">JIfd</a>
-------	----------------------

parameter	name direction type multiplicity default <b>return return float</b>
ownedMember	<b>return</b>

#### Operation **JIfd::getGPSLatitudeRef**

owner	<b>JIfd</b>
parameter	name direction type multiplicity default <b>return return String</b>
ownedMember	<b>return</b>

#### Operation **JIfd::getGPSLongitude**

owner	<b>JIfd</b>
parameter	name direction type multiplicity default <b>return return float</b>
ownedMember	<b>return</b>

#### Operation **JIfd::getGPSLongitudeRef**

owner	<b>JIfd</b>
parameter	name direction type multiplicity default <b>return return String</b>
ownedMember	<b>return</b>

#### Operation **JIfd::getGPSMeasureMode**

owner	<b>JIfd</b>
parameter	name direction type multiplicity default <b>return return String</b>
ownedMember	<b>return</b>

#### Operation **JIfd::getGPSSatellites**

owner	<b>JIfd</b>
parameter	name direction type multiplicity default <b>return return String</b>
ownedMember	<b>return</b>

#### Operation **JIfd::getGPSStatus**

owner	<b>JIfd</b>
parameter	name direction type multiplicity default <b>return return String</b>
ownedMember	<b>return</b>

#### Operation **JIfd::getGPSTimestamp**

owner	<b>JIfd</b>
parameter	name direction type multiplicity default <b>return return Date</b>
ownedMember	<b>return</b>

#### Operation **JIfd::getGPSVersionID**

owner	<b>JIfd</b>
parameter	name direction type multiplicity default <b>return return String</b>
ownedMember	<b>return</b>

#### Operation **JIfd::getGradFromExifTag**

owner	<b>JIfd</b>
parameter	name direction type multiplicity default <b>tag in JExifTag return return float</b>
ownedMember	<b>return tag</b>

#### Operation **JIfd::getISOSpeedRatings**

owner	<b>JIfd</b>
parameter	name direction type multiplicity default <b>return return int</b>
ownedMember	<b>return</b>

#### Operation **JIfd::getLightSource**

owner	<b>JIfd</b>
parameter	name direction type multiplicity default <b>return return int</b>
ownedMember	<b>return</b>

#### Operation **JIfd::getMake**

owner	<b>JIfd</b>
parameter	name direction type multiplicity default <b>return return String</b>
ownedMember	<b>return</b>

#### Operation **JIfd::getMaxApertureValue**

owner	<b>JIfd</b>
parameter	name direction type multiplicity default <b>return return float</b>
ownedMember	<b>return</b>

#### Operation **JIfd::getMeteringMode**

owner	<b>JIfd</b>
parameter	name direction type multiplicity default <b>return return int</b>
ownedMember	<b>return</b>

#### Operation **JIfd::getModel**

owner	<b>JIfd</b>
parameter	name direction type multiplicity default <b>return return String</b>
ownedMember	<b>return</b>

#### Operation **JIfd::getOrientation**

owner	<b>JIfd</b>
parameter	name direction type multiplicity default <b>return return int</b>
ownedMember	<b>return</b>

#### Operation **JIfd::getOriginalDateTime**

owner	<b>JIfd</b>
parameter	name direction type multiplicity default <b>return return String</b>
ownedMember	<b>return</b>

#### Operation **JIfd::getPixelXDimension**

owner	<b>JIfd</b>
parameter	name direction type multiplicity default <b>return return int</b>
ownedMember	<b>return</b>

#### Operation **JIfd::getPixelYDimension**

owner	<b>JIfd</b>
parameter	name direction type multiplicity default <b>return return int</b>
ownedMember	<b>return</b>

#### Operation **JIfd::getResolutionUnit**

owner	<b>JIfd</b>
parameter	name direction type multiplicity default <b>return return int</b>
ownedMember	<b>return</b>

#### Operation **JIfd::getSaturation**

owner	<b>JIfd</b>
parameter	name direction type multiplicity default <b>return return int</b>
ownedMember	<b>return</b>

#### Operation **JIfd::getSceneCaptureType**

owner	<b>JIfd</b>
-------	-------------

parameter	name direction type multiplicity default <b>return return int</b>
ownedMember	<b>return</b>

#### Operation **JIfd::getSceneType**

owner	<b>JIfd</b>
parameter	name direction type multiplicity default <b>return return int</b>
ownedMember	<b>return</b>

#### Operation **JIfd::getSensingMode**

owner	<b>JIfd</b>
parameter	name direction type multiplicity default <b>return return int</b>
ownedMember	<b>return</b>

#### Operation **JIfd::getSharpness**

owner	<b>JIfd</b>
parameter	name direction type multiplicity default <b>return return int</b>
ownedMember	<b>return</b>

#### Operation **JIfd::getShutterSpeedValue**

owner	<b>JIfd</b>
parameter	name direction type multiplicity default <b>return return float</b>
ownedMember	<b>return</b>

#### Operation **JIfd::getSoftware**

owner	<b>JIfd</b>
parameter	name direction type multiplicity default <b>return return String</b>
ownedMember	<b>return</b>

#### Operation **JIfd::getSubjectDistanceRange**

owner	<b>JIfd</b>
parameter	name direction type multiplicity default <b>return return int</b>
ownedMember	<b>return</b>

#### Operation **JIfd::getWhiteBalance**

owner	<b>JIfd</b>
parameter	name direction type multiplicity default <b>return return int</b>
ownedMember	<b>return</b>

**Operation JIfd::getXResolution**

owner	<b>JIfd</b>
parameter	name direction type multiplicity default <b>return return float</b>
ownedMember	<b>return</b>

**Operation JIfd::getYCbCrPosition**

owner	<b>JIfd</b>
parameter	name direction type multiplicity default <b>return return int</b>
ownedMember	<b>return</b>

**Operation JIfd::getYResolution**

owner	<b>JIfd</b>
parameter	name direction type multiplicity default <b>return return float</b>
ownedMember	<b>return</b>

**Property JIfd::GRAD\_INVALID**

owner	<b>JIfd</b>
-------	-------------

**Operation JIfd::isTagPresent**

owner	<b>JIfd</b>
parameter	name direction type multiplicity default <b>tag in int return return boolean</b>
ownedMember	<b>return tag</b>

**Operation JIfd::JIfd**

owner	<b>JIfd</b>
documentation	Creates a new instance of JIfd

**Property JIfd::m\_bIsIntel**

owner	<b>JIfd</b>
-------	-------------

**Property JIfd::m\_count**

owner	<b>JIfd</b>
-------	-------------

**Property JIfd::m\_nextIfdOffset**

owner	<b>JIfd</b>
-------	-------------

**Property JIfd::m\_offset**

owner	<a href="#">JIfd</a>
-------	----------------------

**Property JIfd::m\_tagArray**

owner	<a href="#">JIfd</a>
-------	----------------------

**Operation JIfd::read**

owner	<a href="#">JIfd</a>
parameter	name direction type multiplicity default <b>file in RandomAccessFile bintel in boolean tiffOffset in long return return boolean</b>
ownedMember	<b>bintel file return tiffOffset</b>

**Operation JIfd::readIFDTag**

owner	<a href="#">JIfd</a>
parameter	name direction type multiplicity default <b>file in RandomAccessFile bintel in boolean tiffOffset in long tagMask in int return return boolean</b>
ownedMember	<b>bintel file return tagMask tiffOffset</b>

**Operation JIfd::updateTag**

owner	<a href="#">JIfd</a>
parameter	name direction type multiplicity default <b>file in RandomAccessFile tag in int value in int return return boolean</b>
ownedMember	<b>file return tag value</b>

## Class JIfdData

diagram	<pre> &lt;&lt;final&gt;&gt; JIfdData  m_fileName:String m_fileSize:long=-1  &lt;&lt;constructor&gt;&gt; JIfdData() &lt;&lt;constructor&gt;&gt; JIfdData(<b>in</b> imageFile:<b>File</b>) &lt;&lt;constructor&gt;&gt; JIfdData(<b>in</b> imageFile:<b>String</b>) getFileSize():long getFileDate():<b>Date</b> getFileName():<b>String</b> getFullFileName():<b>String</b> getFilePath():<b>String</b> setFilePath(<b>in</b> fileName:<b>String</b>):void readFromFile(<b>in</b> imageFile:<b>String</b>):boolean readFromFile(<b>in</b> imageFile:<b>File</b>):boolean </pre>
hierarchy	<pre> classDiagram     JIfd &lt; -- JIfdData </pre>
owner	jexifviewer
ownedMember	<a href="#">getFileDialog</a> <a href="#">getFileName</a> <a href="#">getFilePath</a> <a href="#">getFileSize</a> <a href="#">getFullFileName</a> <a href="#">JIfdData</a> <a href="#">JIfdData</a> <a href="#">m_fileName</a> <a href="#">m_fileSize</a> <a href="#">readFromFile</a> <a href="#">readFromFile</a> <a href="#">setFilePath</a>
general	<a href="#">JIfd</a>
target of relation	ComponentRealization <a href="#">Implementierung</a>
typedElements	Class <a href="#">ExifAdapter</a> Property <a href="#">exifData</a>
documentation	@author reiner

## Operation JIfdData::getFileDialog

owner	<a href="#">JIfdData</a>
parameter	name direction type multiplicity default <b>return return Date</b>
ownedMember	<b>return</b>

## Operation JIfdData::getFileName

owner	<a href="#">JIfdData</a>
parameter	name direction type multiplicity default <b>return return String</b>
ownedMember	<b>return</b>

**Operation JIfdData::getFilePath**

owner	<a href="#">JIfdData</a>
parameter	name direction type multiplicity default <b>return return String</b>
ownedMember	<b>return</b>

**Operation JIfdData::getFileSize**

owner	<a href="#">JIfdData</a>
parameter	name direction type multiplicity default <b>return return long</b>
ownedMember	<b>return</b>

**Operation JIfdData::getFullFileName**

owner	<a href="#">JIfdData</a>
parameter	name direction type multiplicity default <b>return return String</b>
ownedMember	<b>return</b>

**Operation JIfdData::JIfdData**

owner	<a href="#">JIfdData</a>
documentation	Creates a new instance of JIfdData

**Operation JIfdData::JIfdData**

owner	<a href="#">JIfdData</a>
parameter	name direction type multiplicity default <b>imageFile in File</b>
ownedMember	<b>imageFile</b>

**Operation JIfdData::JIfdData**

owner	<a href="#">JIfdData</a>
parameter	name direction type multiplicity default <b>imageFile in String</b>
ownedMember	<b>imageFile</b>

**Property JIfdData::m\_fileName**

owner	<a href="#">JIfdData</a>
-------	--------------------------

**Property JIfdData::m\_fileSize**

owner	<a href="#">JIfdData</a>
-------	--------------------------

#### Operation **JIfdData::readFromFile**

owner	<b>JIfdData</b>
parameter	name direction type multiplicity default <b>imageFile in String return return boolean</b>
ownedMember	<b>imageFile return</b>

#### Operation **JIfdData::readFromFile**

owner	<b>JIfdData</b>
parameter	name direction type multiplicity default <b>imageFile in File return return boolean</b>
ownedMember	<b>imageFile return</b>

#### Operation **JIfdData::setFilePath**

owner	<b>JIfdData</b>
parameter	name direction type multiplicity default <b>fileName in String return return void</b>
ownedMember	<b>fileName return</b>

## Class JIfdData

diagram	<pre> &lt;&lt;final&gt;&gt; JIfdData  m_fileName:String m_fileSize:long=-1  &lt;&lt;constructor&gt;&gt; JIfdData() &lt;&lt;constructor&gt;&gt; JIfdData(<b>in</b> imageFile:<b>File</b>) &lt;&lt;constructor&gt;&gt; JIfdData(<b>in</b> imageFile:<b>String</b>) getFileSize():long getFileDate():<b>Date</b> getFileName():<b>String</b> getFullFileName():<b>String</b> getFilePath():<b>String</b> setFilePath(<b>in</b> fileName:<b>String</b>):void readFromFile(<b>in</b> imageFile:<b>String</b>):boolean readFromFile(<b>in</b> imageFile:<b>File</b>):boolean </pre>
hierarchy	<pre> classDiagram     JIfd &lt; -- JIfdData </pre>
owner	jexifviewer
ownedMember	<a href="#">getFileDialog</a> <a href="#">getFileName</a> <a href="#">getFilePath</a> <a href="#">getFileSize</a> <a href="#">getFullFileName</a> <a href="#">JIfdData</a> <a href="#">JIfdData</a> <a href="#">m_fileName</a> <a href="#">m_fileSize</a> <a href="#">readFromFile</a> <a href="#">readFromFile</a> <a href="#">setFilePath</a>
general	<a href="#">JIfd</a>
target of relation	ComponentRealization <a href="#">Implementierung</a>
typedElements	Class <a href="#">ExifAdapter</a> Property <a href="#">exifData</a>
documentation	@author reiner

## Operation JIfdData::getFileDialog

owner	<a href="#">JIfdData</a>
parameter	name direction type multiplicity default <b>return return Date</b>
ownedMember	<b>return</b>

## Operation JIfdData::getFileName

owner	<a href="#">JIfdData</a>
parameter	name direction type multiplicity default <b>return return String</b>
ownedMember	<b>return</b>

**Operation JIfdData::getFilePath**

owner	<a href="#">JIfdData</a>
parameter	name direction type multiplicity default <b>return return String</b>
ownedMember	<b>return</b>

**Operation JIfdData::getFileSize**

owner	<a href="#">JIfdData</a>
parameter	name direction type multiplicity default <b>return return long</b>
ownedMember	<b>return</b>

**Operation JIfdData::getFullFileName**

owner	<a href="#">JIfdData</a>
parameter	name direction type multiplicity default <b>return return String</b>
ownedMember	<b>return</b>

**Operation JIfdData::JIfdData**

owner	<a href="#">JIfdData</a>
documentation	Creates a new instance of JIfdData

**Operation JIfdData::JIfdData**

owner	<a href="#">JIfdData</a>
parameter	name direction type multiplicity default <b>imageFile in File</b>
ownedMember	<b>imageFile</b>

**Operation JIfdData::JIfdData**

owner	<a href="#">JIfdData</a>
parameter	name direction type multiplicity default <b>imageFile in String</b>
ownedMember	<b>imageFile</b>

**Property JIfdData::m\_fileName**

owner	<a href="#">JIfdData</a>
-------	--------------------------

**Property JIfdData::m\_fileSize**

owner	<a href="#">JIfdData</a>
-------	--------------------------

#### Operation **JIfdData::readFromFile**

owner	<b>JIfdData</b>
parameter	name direction type multiplicity default <b>imageFile in String return return boolean</b>
ownedMember	<b>imageFile return</b>

#### Operation **JIfdData::readFromFile**

owner	<b>JIfdData</b>
parameter	name direction type multiplicity default <b>imageFile in File return return boolean</b>
ownedMember	<b>imageFile return</b>

#### Operation **JIfdData::setFilePath**

owner	<b>JIfdData</b>
parameter	name direction type multiplicity default <b>fileName in String return return void</b>
ownedMember	<b>fileName return</b>

## Class JJPEGHelper

diagram	<pre> &lt;&lt;final&gt;&gt; JJPEGHelper  &lt;&lt;final&gt;&gt; SOI:int=0xFFD8 &lt;&lt;final&gt;&gt; JFIF:int=0xFFE0 &lt;&lt;final&gt;&gt; COM:int=0xFFFF &lt;&lt;final&gt;&gt; COP:int=0xFFEE &lt;&lt;final&gt;&gt; APP1:int=0xFFE1 &lt;&lt;final&gt;&gt; APP2:int=0xFFE2 &lt;&lt;final&gt;&gt; DQT:int=0xFFDB &lt;&lt;final&gt;&gt; DHT:int=0xFFC4 &lt;&lt;final&gt;&gt; DRI:int=0xFFDD &lt;&lt;final&gt;&gt; SOF:int=0xFFC0 &lt;&lt;final&gt;&gt; SOS:int=0xFFDA &lt;&lt;final&gt;&gt; EOI:int=0xFFD9 m_size:int m_offset:int  getSize():int getOffset():int &lt;&lt;constructor&gt;&gt; JJPEGHelper() isJpegFile(<b>in</b> file:RandomAccessFile):boolean findMarker(<b>in</b> file:RandomAccessFile, <b>in</b> marker:int):boolean isMarker(<b>in</b> buf:byte[*], <b>in</b> marker:int):boolean   </pre>
owner	<a href="#">jexifviewer</a>
ownedMember	<a href="#">APP1</a> <a href="#">APP2</a> <a href="#">COM</a> <a href="#">COP</a> <a href="#">DHT</a> <a href="#">DQT</a> <a href="#">DRI</a> <a href="#">EOI</a> <a href="#">findMarker</a> <a href="#">getOffset</a> <a href="#">getSize</a> <a href="#">isJpegFile</a> <a href="#">isMarker</a> <a href="#">JFIF</a> <a href="#">JJPEGHelper</a> <a href="#">m_offset</a> <a href="#">m_size</a> <a href="#">SOF</a> <a href="#">SOS</a>
target of relation	ComponentRealization <a href="#">Implementierung</a>
documentation	@author reiner

## Property JJPEGHelper::APP1

owner	<a href="#">JJPEGHelper</a>
-------	-----------------------------

## Property JJPEGHelper::APP2

owner	<a href="#">JJPEGHelper</a>
-------	-----------------------------

## Property JJPEGHelper::COM

owner	<a href="#">JJPEGHelper</a>
-------	-----------------------------

Property **JJPEGHelper::COP**

owner	<a href="#">JJPEGHelper</a>
-------	-----------------------------

Property **JJPEGHelper::DHT**

owner	<a href="#">JJPEGHelper</a>
-------	-----------------------------

Property **JJPEGHelper::DQT**

owner	<a href="#">JJPEGHelper</a>
-------	-----------------------------

Property **JJPEGHelper::DRI**

owner	<a href="#">JJPEGHelper</a>
-------	-----------------------------

Property **JJPEGHelper::EOI**

owner	<a href="#">JJPEGHelper</a>
-------	-----------------------------

Operation **JJPEGHelper::findMarker**

owner	<a href="#">JJPEGHelper</a>
parameter	name direction type multiplicity default <b>file in RandomAccessFile</b> <b>marker in int</b> <b>return return boolean</b>
ownedMember	<b>file marker return</b>

Operation **JJPEGHelper::getOffset**

owner	<a href="#">JJPEGHelper</a>
parameter	name direction type multiplicity default <b>return return int</b>
ownedMember	<b>return</b>

Operation **JJPEGHelper::getSize**

owner	<a href="#">JJPEGHelper</a>
parameter	name direction type multiplicity default <b>return return int</b>
ownedMember	<b>return</b>
documentation	Creates a new instance of JJPEGHelper

Operation **JJPEGHelper::isJpegFile**

owner	<a href="#">JJPEGHelper</a>
parameter	name direction type multiplicity default <b>file in RandomAccessFile</b> <b>return return boolean</b>
ownedMember	<b>file return</b>

**Operation JJPEGHelper::isMarker**

owner	<a href="#">JJPEGHelper</a>
parameter	name direction type multiplicity default <b>buf in byte * marker in int return return boolean</b>
ownedMember	<b>buf marker return</b>

**Property JJPEGHelper::JFIF**

owner	<a href="#">JJPEGHelper</a>
-------	-----------------------------

**Operation JJPEGHelper::JJPEGHelper**

owner	<a href="#">JJPEGHelper</a>
-------	-----------------------------

**Property JJPEGHelper::m\_offset**

owner	<a href="#">JJPEGHelper</a>
-------	-----------------------------

**Property JJPEGHelper::m\_size**

owner	<a href="#">JJPEGHelper</a>
-------	-----------------------------

**Property JJPEGHelper::SOF**

owner	<a href="#">JJPEGHelper</a>
-------	-----------------------------

**Property JJPEGHelper::SOI**

owner	<a href="#">JJPEGHelper</a>
-------	-----------------------------

**Property JJPEGHelper::SOS**

owner	<a href="#">JJPEGHelper</a>
-------	-----------------------------

## Class JJPEGHelper

diagram	<pre> &lt;&lt;final&gt;&gt; JJPEGHelper  &lt;&lt;final&gt;&gt; SOI:int=0xFFD8 &lt;&lt;final&gt;&gt; JFIF:int=0xFFE0 &lt;&lt;final&gt;&gt; COM:int=0xFFFF &lt;&lt;final&gt;&gt; COP:int=0xFFEE &lt;&lt;final&gt;&gt; APP1:int=0xFFE1 &lt;&lt;final&gt;&gt; APP2:int=0xFFE2 &lt;&lt;final&gt;&gt; DQT:int=0xFFDB &lt;&lt;final&gt;&gt; DHT:int=0xFFC4 &lt;&lt;final&gt;&gt; DRI:int=0xFFDD &lt;&lt;final&gt;&gt; SOF:int=0xFFC0 &lt;&lt;final&gt;&gt; SOS:int=0xFFDA &lt;&lt;final&gt;&gt; EOI:int=0xFFD9 m_size:int m_offset:int  getSize():int getOffset():int &lt;&lt;constructor&gt;&gt; JJPEGHelper() isJpegFile(in file:RandomAccessFile):boolean findMarker(in file:RandomAccessFile, in marker:int):boolean isMarker(in buf:byte[], in marker:int):boolean </pre>
owner	jexifviewer
ownedMember	APP1 APP2 COM COP DHT DQT DRI EOI findMarker getSize isJpegFile isMarker JFIF JJPEGHelper m_offset m_size SOF SOI SOS
target of relation	ComponentRealization <a href="#">Implementierung</a>
documentation	@author reiner

### Property JJPEGHelper::APP1

owner	<a href="#">JJPEGHelper</a>
-------	-----------------------------

### Property JJPEGHelper::APP2

owner	<a href="#">JJPEGHelper</a>
-------	-----------------------------

### Property JJPEGHelper::COM

owner	<a href="#">JJPEGHelper</a>
-------	-----------------------------

Property **JJPEGHelper::COP**

owner	<a href="#">JJPEGHelper</a>
-------	-----------------------------

Property **JJPEGHelper::DHT**

owner	<a href="#">JJPEGHelper</a>
-------	-----------------------------

Property **JJPEGHelper::DQT**

owner	<a href="#">JJPEGHelper</a>
-------	-----------------------------

Property **JJPEGHelper::DRI**

owner	<a href="#">JJPEGHelper</a>
-------	-----------------------------

Property **JJPEGHelper::EOI**

owner	<a href="#">JJPEGHelper</a>
-------	-----------------------------

Operation **JJPEGHelper::findMarker**

owner	<a href="#">JJPEGHelper</a>
parameter	name direction type multiplicity default <b>file in RandomAccessFile</b> <b>marker in int</b> <b>return return boolean</b>
ownedMember	<b>file marker return</b>

Operation **JJPEGHelper::getOffset**

owner	<a href="#">JJPEGHelper</a>
parameter	name direction type multiplicity default <b>return return int</b>
ownedMember	<b>return</b>

Operation **JJPEGHelper::getSize**

owner	<a href="#">JJPEGHelper</a>
parameter	name direction type multiplicity default <b>return return int</b>
ownedMember	<b>return</b>
documentation	Creates a new instance of JJPEGHelper

Operation **JJPEGHelper::isJpegFile**

owner	<a href="#">JJPEGHelper</a>
parameter	name direction type multiplicity default <b>file in RandomAccessFile</b> <b>return return boolean</b>
ownedMember	<b>file return</b>

**Operation JJPEGHelper::isMarker**

owner	<a href="#">JJPEGHelper</a>
parameter	name direction type multiplicity default <b>buf in byte * marker in int return return boolean</b>
ownedMember	<b>buf marker return</b>

**Property JJPEGHelper::JFIF**

owner	<a href="#">JJPEGHelper</a>
-------	-----------------------------

**Operation JJPEGHelper::JJPEGHelper**

owner	<a href="#">JJPEGHelper</a>
-------	-----------------------------

**Property JJPEGHelper::m\_offset**

owner	<a href="#">JJPEGHelper</a>
-------	-----------------------------

**Property JJPEGHelper::m\_size**

owner	<a href="#">JJPEGHelper</a>
-------	-----------------------------

**Property JJPEGHelper::SOF**

owner	<a href="#">JJPEGHelper</a>
-------	-----------------------------

**Property JJPEGHelper::SOI**

owner	<a href="#">JJPEGHelper</a>
-------	-----------------------------

**Property JJPEGHelper::SOS**

owner	<a href="#">JJPEGHelper</a>
-------	-----------------------------

## Class JMyFileFilter

diagram	<pre> classDiagram     class JMyFileFilter {         m_extList:ArrayList&lt;E-&gt;String&gt;         m_description:String          &lt;&lt;constructor&gt;&gt; JMyFileFilter()         addExtension(in ext:String):void         accept(in file:File):boolean         getDescription():String         setDescription(in description:String):void     }   </pre>
hierarchy	<pre> classDiagram     class FileFilter     class JMyFileFilter {         &lt; -- FileFilter     }   </pre>
owner	<a href="#">files</a>
ownedMember	<a href="#">accept</a> <a href="#">addExtension</a> <a href="#">getDescription</a> <a href="#">JMyFileFilter</a> <a href="#">m_description</a> <a href="#">m_extList</a> <a href="#">setDescription</a>
general	<a href="#">FileFilter</a>
target of relation	ComponentRealization <a href="#">Implementierung</a>
documentation	@author Reiner

### Operation JMyFileFilter::accept

owner	<a href="#">JMyFileFilter</a>
parameter	name direction type multiplicity default <b>file</b> in <b>File</b> return return <b>boolean</b>
ownedMember	<b>file return</b>

### Operation JMyFileFilter::addExtension

owner	<a href="#">JMyFileFilter</a>
parameter	name direction type multiplicity default <b>ext</b> in <b>String</b> return return <b>void</b>
ownedMember	<b>ext return</b>

### Operation JMyFileFilter::getDescription

owner	<a href="#">JMyFileFilter</a>
parameter	name direction type multiplicity default <b>return</b> return <b>String</b>
ownedMember	<b>return</b>

**Operation JMyFileFilter::JMyFileFilter**

owner	<a href="#">JMyFileFilter</a>
documentation	Creates a new instance of JMyFileFilter

**Property JMyFileFilter::m\_description**

owner	<a href="#">JMyFileFilter</a>
-------	-------------------------------

**Property JMyFileFilter::m\_extList**

owner	<a href="#">JMyFileFilter</a>
-------	-------------------------------

**Operation JMyFileFilter::setDescription**

owner	<a href="#">JMyFileFilter</a>
parameter	name direction type multiplicity default <b>description in String return return void</b>
ownedMember	<b>description return</b>

## Class JMyFileFilter

diagram	<pre> classDiagram     class JMyFileFilter {         m_extList:ArrayList&lt;E-&gt;String&gt;         m_description:String          &lt;&lt;constructor&gt;&gt; JMyFileFilter()         addExtension(in ext:String):void         accept(in file:File):boolean         getDescription():String         setDescription(in description:String):void     }   </pre>
hierarchy	<pre> classDiagram     class FileFilter     class JMyFileFilter {         &lt; -- FileFilter     }   </pre>
owner	<a href="#">files</a>
ownedMember	<a href="#">accept</a> <a href="#">addExtension</a> <a href="#">getDescription</a> <a href="#">JMyFileFilter</a> <a href="#">m_description</a> <a href="#">m_extList</a> <a href="#">setDescription</a>
general	<a href="#">FileFilter</a>
target of relation	ComponentRealization <a href="#">Implementierung</a>
documentation	@author Reiner

### Operation JMyFileFilter::accept

owner	<a href="#">JMyFileFilter</a>
parameter	name direction type multiplicity default <b>file</b> in <b>File</b> return return <b>boolean</b>
ownedMember	<b>file return</b>

### Operation JMyFileFilter::addExtension

owner	<a href="#">JMyFileFilter</a>
parameter	name direction type multiplicity default <b>ext</b> in <b>String</b> return return <b>void</b>
ownedMember	<b>ext return</b>

### Operation JMyFileFilter::getDescription

owner	<a href="#">JMyFileFilter</a>
parameter	name direction type multiplicity default <b>return</b> return <b>String</b>
ownedMember	<b>return</b>

**Operation JMyFileFilter::JMyFileFilter**

owner	<a href="#">JMyFileFilter</a>
documentation	Creates a new instance of JMyFileFilter

**Property JMyFileFilter::m\_description**

owner	<a href="#">JMyFileFilter</a>
-------	-------------------------------

**Property JMyFileFilter::m\_extList**

owner	<a href="#">JMyFileFilter</a>
-------	-------------------------------

**Operation JMyFileFilter::setDescription**

owner	<a href="#">JMyFileFilter</a>
parameter	name direction type multiplicity default <b>description in String return return void</b>
ownedMember	<b>description return</b>

## Class JParameters

diagram	<pre> classDiagram     class JParameters {         &lt;&lt;final&gt;&gt; serialVersionUID:long=1L         &lt;&lt;constructor&gt;&gt; JParameters(in titel:String, in icon:Icon, in tip:String)         getAxes():ArrayList&lt;E-&gt;Axis&gt;     }     JParameters &lt; -- ImageObserver     JParameters &lt; -- MenuContainer     JParameters &lt; -- Serializable     JParameters &lt; -- Component     JParameters &lt; -- Container     JParameters &lt; -- JComponent     JParameters &lt; -- JAbstractSinglePanel   </pre>
hierarchy	<pre> classDiagram     class ImageObserver     class MenuContainer     class Serializable     class Component     class Container     class JComponent     class JAbstractSinglePanel     class JParameters {         &lt;&lt;final&gt;&gt; serialVersionUID:long=1L         &lt;&lt;constructor&gt;&gt; JParameters(in titel:String, in icon:Icon, in tip:String)         getAxes():ArrayList&lt;E-&gt;Axis&gt;     }     JParameters &lt; -- ImageObserver     JParameters &lt; -- MenuContainer     JParameters &lt; -- Serializable     JParameters &lt; -- Component     JParameters &lt; -- Container     JParameters &lt; -- JComponent     JParameters &lt; -- JAbstractSinglePanel   </pre>
owner	<a href="#">reportmanagement</a>
ownedMember	<a href="#">getAxes</a> <a href="#">JParameters</a> <a href="#">serialVersionUID</a>
general	<a href="#">JAbstractSinglePanel</a>
target of relation	ComponentRealization <a href="#">Implementierung</a>
documentation	<p>This class represents the panel where the user is able to assign a parameter with an optional description to each available axis.</p> <p>@author David Kaufman</p>

## Operation JParameters::getAxes

owner	<a href="#">JParameters</a>
parameter	name direction type multiplicity default <b>return return ArrayList&lt;E-&gt;Axis&gt;</b>
ownedMember	<b>return</b>
documentation	Returns the Exif-parameters and axes description specified @return the Exif-parameters and axes description

## Operation JParameters::JParameters

owner	<a href="#">JParameters</a>
parameter	name direction type multiplicity default <b>titel in String icon in Icon tip in String</b>
ownedMember	<b>icon tip titel</b>
documentation	

	on	Constructor which initialized this parameter panel @param titel The title which is registered with this panel. @param icon The icon which is registered with this panel. @param tip The tooltip which is registered with this panel.
--	----	---

### Property **JParameters::serialVersionUID**

owner	<a href="#">JParameters</a>
documentati on	

## Class JPathHelper

diagram	<pre> <b>JPathHelper</b> +--&gt;&lt;&lt;constructor&gt;&gt; JPathHelper() +--&gt; getFileExtension(<i>in file:File</i>):String +--&gt; getFileExtension(<i>in fileName:String</i>):String +--&gt; getBaseFileName(<i>in file:File</i>):String +--&gt; getBaseFileName(<i>in fileName:String</i>):String +--&gt; getFolder(<i>in file:File</i>):String +--&gt; getFolder(<i>in fileName:String</i>):String +--&gt; getSubFolder(<i>in file:File</i>):String +--&gt; getSubFolder(<i>in fileName:String</i>):String +--&gt; addSeparator(<i>in stringBuilder:StringBuilder</i>):void +--&gt; addSeparator(<i>in str:String</i>):String </pre>
owner	<b>files</b>
ownedMember	<b>addSeparator addSeparator getBaseFileName getBaseFileName getFileExtension getFileExtension getFolder getFolder getSubFolder getSubFolder JPathHelper</b>
target of relation	ComponentRealization <b>Implementierung</b>
documentation	@author Reiner

### Operation JPathHelper::addSeparator

owner	<b>JPathHelper</b>
parameter	name direction type multiplicity default <b>stringBuilder in StringBuilder return return void</b>
ownedMember	<b>return stringBuilder</b>

### Operation JPathHelper::addSeparator

owner	<b>JPathHelper</b>
parameter	name direction type multiplicity default <b>str in String return return String</b>
ownedMember	<b>return str</b>

### Operation JPathHelper::getBaseFileName

owner	<b>JPathHelper</b>
parameter	name direction type multiplicity default <b>file in File return return String</b>
ownedMember	<b>file return</b>

### Operation JPathHelper::getBaseFileName

owner	<b>JPathHelper</b>
parameter	name direction type multiplicity default <b>fileName in String return return String</b>

ownedMember	<b>fileName return</b>
-------------	------------------------

#### Operation **JPathHelper::getFileExtension**

owner	<b>JPathHelper</b>
parameter	name direction type multiplicity default <b>file in File return return String</b>
ownedMember	<b>file return</b>

#### Operation **JPathHelper::getFileExtension**

owner	<b>JPathHelper</b>
parameter	name direction type multiplicity default <b>fileName in String return return String</b>
ownedMember	<b>fileName return</b>

#### Operation **JPathHelper::getFolder**

owner	<b>JPathHelper</b>
parameter	name direction type multiplicity default <b>file in File return return String</b>
ownedMember	<b>file return</b>

#### Operation **JPathHelper::getFolder**

owner	<b>JPathHelper</b>
parameter	name direction type multiplicity default <b>fileName in String return return String</b>
ownedMember	<b>fileName return</b>

#### Operation **JPathHelper::getSubFolder**

owner	<b>JPathHelper</b>
parameter	name direction type multiplicity default <b>file in File return return String</b>
ownedMember	<b>file return</b>

#### Operation **JPathHelper::getSubFolder**

owner	<b>JPathHelper</b>
parameter	name direction type multiplicity default <b>fileName in String return return String</b>
ownedMember	<b>fileName return</b>

#### Operation **JPathHelper::JPathHelper**

owner	<b>JPathHelper</b>
documentation	Creates a new instance of JPathHelper



## Class JPathHelper

diagram	<pre> <b>JPathHelper</b> +--&gt;&lt;&lt;constructor&gt;&gt; JPathHelper() +--&gt; getFileExtension(<i>in file:File</i>):String +--&gt; getFileExtension(<i>in fileName:String</i>):String +--&gt; getBaseFileName(<i>in file:File</i>):String +--&gt; getBaseFileName(<i>in fileName:String</i>):String +--&gt; getFolder(<i>in file:File</i>):String +--&gt; getFolder(<i>in fileName:String</i>):String +--&gt; getSubFolder(<i>in file:File</i>):String +--&gt; getSubFolder(<i>in fileName:String</i>):String +--&gt; addSeparator(<i>in stringBuilder:StringBuilder</i>):void +--&gt; addSeparator(<i>in str:String</i>):String </pre>
owner	<b>files</b>
ownedMember	<a href="#">addSeparator</a> <a href="#">addSeparator</a> <a href="#">getBaseFileName</a> <a href="#">getBaseFileName</a> <a href="#">getFileExtension</a> <a href="#">getFileExtension</a> <a href="#">getFolder</a> <a href="#">getFolder</a> <a href="#">getSubFolder</a> <a href="#">getSubFolder</a> <a href="#">JPathHelper</a>
target of relation	ComponentRealization <a href="#">Implementierung</a>
documentation	@author Reiner

### Operation JPathHelper::addSeparator

owner	<b>JPathHelper</b>
parameter	name direction type multiplicity default <b>stringBuilder</b> <i>in StringBuilder</i> return return void
ownedMember	<b>return stringBuilder</b>

### Operation JPathHelper::addSeparator

owner	<b>JPathHelper</b>
parameter	name direction type multiplicity default <b>str</b> <i>in String</i> return return String
ownedMember	<b>return str</b>

### Operation JPathHelper::getBaseFileName

owner	<b>JPathHelper</b>
parameter	name direction type multiplicity default <b>file</b> <i>in File</i> return return String
ownedMember	<b>file return</b>

### Operation JPathHelper::getBaseFileName

owner	<b>JPathHelper</b>
parameter	name direction type multiplicity default <b>fileName</b> <i>in String</i> return return String

ownedMember	<b>fileName return</b>
-------------	------------------------

#### Operation **JPathHelper::getFileExtension**

owner	<b>JPathHelper</b>
parameter	name direction type multiplicity default <b>file in File return return String</b>
ownedMember	<b>file return</b>

#### Operation **JPathHelper::getFileExtension**

owner	<b>JPathHelper</b>
parameter	name direction type multiplicity default <b>fileName in String return return String</b>
ownedMember	<b>fileName return</b>

#### Operation **JPathHelper::getFolder**

owner	<b>JPathHelper</b>
parameter	name direction type multiplicity default <b>file in File return return String</b>
ownedMember	<b>file return</b>

#### Operation **JPathHelper::getFolder**

owner	<b>JPathHelper</b>
parameter	name direction type multiplicity default <b>fileName in String return return String</b>
ownedMember	<b>fileName return</b>

#### Operation **JPathHelper::getSubFolder**

owner	<b>JPathHelper</b>
parameter	name direction type multiplicity default <b>file in File return return String</b>
ownedMember	<b>file return</b>

#### Operation **JPathHelper::getSubFolder**

owner	<b>JPathHelper</b>
parameter	name direction type multiplicity default <b>fileName in String return return String</b>
ownedMember	<b>fileName return</b>

#### Operation **JPathHelper::JPathHelper**

owner	<b>JPathHelper</b>
documentation	Creates a new instance of JPathHelper



## Class JPictureSetExif

diagram	<pre> classDiagram     class JPictureSetExif {         &lt;&lt;final&gt;&gt; serialVersionUID:long=1L         pictureContainer:PictureSet[*]         exifFilterKeywords:String[*]         availablePictureSets:JList         associatedPictureSets:JList          &lt;&lt;constructor&gt;&gt; JPictureSetExif(in title:String, in icon:Icon, in tip:String)         setPictureContainer(in pictureContainer:PictureSet[*]):void         getPictureContainer():PictureSet[*]         setExifFilterKeywords(in exifFilterKeywords:String[*]):void         getExifFilterKeywords():String[*]     }   </pre>
hierarchy	<pre> classDiagram     class ImageObserver     class MenuContainer     class Serializable     class Component     class Container     class JComponent     class JAbstractSinglePanel     class JPictureSetExif      ImageObserver &lt; -- JPictureSetExif     MenuContainer &lt; -- JPictureSetExif     Serializable &lt; -- JPictureSetExif     Component &lt; -- JPictureSetExif     Container &lt; -- JPictureSetExif     JComponent &lt; -- JPictureSetExif     JAbstractSinglePanel &lt; -- JPictureSetExif   </pre>
owner	<a href="#">reportmanagement</a>
ownedMember	<a href="#">associatedPictureSets</a> <a href="#">availablePictureSets</a> <a href="#">exifFilterKeywords</a> <a href="#">getExifFilterKeywords</a> <a href="#">getPictureContainer</a> <a href="#">JPictureSetExif</a> <a href="#">pictureContainer</a> <a href="#">serialVersionUID</a> <a href="#">setExifFilterKeywords</a> <a href="#">setPictureContainer</a>
general	<a href="#">JAbstractSinglePanel</a>
target of relation	ComponentRealization <a href="#">Implementierung</a>
documentation	<p>This class represents the panel where the user is able to assign one or more picture sets to the specified report if available. Also the user has the option to assign one ore more Exif tags to the current report if possible.</p> <p>@author David Kaufman</p>

### Property JPictureSetExif::associatedPictureSets

owner	<a href="#">JPictureSetExif</a>
-------	---------------------------------

### Property JPictureSetExif::availablePictureSets

owner	<a href="#">JPictureSetExif</a>
-------	---------------------------------

### Property **JPictureSetExif::exifFilterKeywords**

owner	<a href="#">JPictureSetExif</a>
-------	---------------------------------

### Operation **JPictureSetExif::getExifFilterKeywords**

owner	<a href="#">JPictureSetExif</a>
parameter	name direction type multiplicity default <b>return return String *</b>
ownedMember	<b>return</b>

### Operation **JPictureSetExif::getPictureContainer**

owner	<a href="#">JPictureSetExif</a>
parameter	name direction type multiplicity default <b>return return PictureSet *</b>
ownedMember	<b>return</b>

### Operation **JPictureSetExif::JPictureSetExif**

owner	<a href="#">JPictureSetExif</a>
parameter	name direction type multiplicity default <b>titel in String icon in Icon tip in String</b>
ownedMember	<b>icon tip titel</b>
documentation	Constructor which initialized this picture set management and the exifkeyword management panel @param titel The title which is registered with this panel. @param icon The icon which is registered with this panel. @param tip The tooltip which is registered with this panel.

### Property **JPictureSetExif::pictureContainer**

owner	<a href="#">JPictureSetExif</a>
-------	---------------------------------

### Property **JPictureSetExif::serialVersionUID**

owner	<a href="#">JPictureSetExif</a>
documentation	

### Operation **JPictureSetExif::setExifFilterKeywords**

owner	<a href="#">JPictureSetExif</a>
parameter	name direction type multiplicity default <b>exifFilterKeywords in String * return return void</b>
ownedMember	<b>exifFilterKeywords return</b>

Operation **JPictureSetExif::setPictureContainer**

owner	<b>JPictureSetExif</b>
parameter	name direction type multiplicity default <b>pictureContainer</b> in <b>PictureSet</b> * return void
ownedMember	<b>pictureContainer return</b>

## Class JProjectManagement

diagram	<p>The diagram shows the class <b>JProjectManagement</b> with its associated interface <b>M&gt;ProjectManagementModel</b>. The class has the following attributes:</p> <ul style="list-style-type: none"> <li><code>&lt;&lt;final&gt;&gt; serialVersionUID:long=2746903025575471227L</code></li> <li><code>iContentPane:JPanel=null</code></li> <li><code>iButtonCopyProject:JButton=null</code></li> <li><code>iButtonCreateProject:JButton=null</code></li> <li><code>iButtonDeleteProject:JButton=null</code></li> <li><code>iButtonOpenProject:JButton=null</code></li> <li><code>iListProject:JList=null</code></li> <li><code>iScrollPaneProjectList:JScrollPane=null</code></li> </ul> <p>The class also implements the following operations:</p> <ul style="list-style-type: none"> <li><code>&lt;&lt;constructor&gt;&gt; JProjectManagement(<b>in</b> model:<b>M</b>)</code></li> <li><code>initialize():void</code></li> <li><code>getJContentPane():JPanel</code></li> <li><code>getJButtonCopyProject():JButton</code></li> <li><code>getJButtonCreateProject():JButton</code></li> <li><code>getJButtonDeleteProject():JButton</code></li> <li><code>getJButtonOpenProject():JButton</code></li> <li><code>getJListProject():JList</code></li> <li><code>getJScrollPaneProjectList():JScrollPane</code></li> <li><code>getSelectedProjects():int[]</code></li> <li><code>&lt;&lt;annotations&gt;&gt; update(<b>in</b> o:Observable, <b>in</b> arg:Object):void</code></li> </ul>
hierarchy	<pre> classDiagram     JAbstractView&lt;M-M&gt; &lt; -- JProjectManagement   </pre>
owner	<b>projectmanagement</b>
template parameters	name kind constrainingClassifier default <b>M</b> Class <a href="#">ProjectManagementModel</a>
ownedMember	<a href="#">getJButtonCopyProject</a> <a href="#">getJButtonCreateProject</a> <a href="#">getJButtonDeleteProject</a> <a href="#">getJButtonOpenProject</a> <a href="#">getJContentPane</a> <a href="#">getJListProject</a> <a href="#">getJScrollPaneProjectList</a> <a href="#">getSelectedProjects</a> <a href="#">initialize</a> <a href="#">jButtonCopyProject</a> <a href="#">jButtonCreateProject</a> <a href="#">jButtonDeleteProject</a> <a href="#">jButtonOpenProject</a> <a href="#">jContentPane</a> <a href="#">jListProject</a> <a href="#">JProjectManagement</a> <a href="#">JScrollPaneProjectList</a> <a href="#">serialVersionUID</a> <a href="#">update</a>
general	<a href="#">JAbstractView&lt;M-M&gt;</a>
target of relation	ComponentRealization <a href="#">Implementierung</a>
documentation	<p>Represents the management for all user projects.</p> <p>Sets all GUI Elements which are described in our Pflichtenheft.</p>

## Operation JProjectManagement::getJButtonCopyProject

owner	<b>JProjectManagement</b>
parameter	name direction type multiplicity default <b>return return JButton</b>
ownedMember	<b>return</b>

documentati on	This method initializes jButtonCopyProject.  @return the button.
-------------------	--

#### Operation **JProjectManagement::getJButtonCreateProject**

owner	<b>JProjectManagement</b>
parameter	name direction type multiplicity default <b>return return JButton</b>
ownedMem ber	<b>return</b>
documentati on	This method initializes jButtonCreateProject.  @return the button.

#### Operation **JProjectManagement::getJButtonDeleteProject**

owner	<b>JProjectManagement</b>
parameter	name direction type multiplicity default <b>return return JButton</b>
ownedMem ber	<b>return</b>
documentati on	This method initializes jButtonDeleteProject.  @return the button.

#### Operation **JProjectManagement::getJButtonOpenProject**

owner	<b>JProjectManagement</b>
parameter	name direction type multiplicity default <b>return return JButton</b>
ownedMem ber	<b>return</b>
documentati on	This method initializes jButtonOpenProject.  @return the button.

#### Operation **JProjectManagement::getJContentPane**

owner	<b>JProjectManagement</b>
parameter	name direction type multiplicity default <b>return return JPanel</b>
ownedMem ber	<b>return</b>
documentati on	This method initializes jContentPane.  @return the panel.

#### Operation **JProjectManagement::getListProject**

owner	<b>JProjectManagement</b>
parameter	name direction type multiplicity default <b>return return JList</b>
ownedMember	<b>return</b>
documentation	This method initializes jListProject.  @return the list.

#### Operation **JProjectManagement::getScrollPaneProjectList**

owner	<b>JProjectManagement</b>
parameter	name direction type multiplicity default <b>return return JScrollPane</b>
ownedMember	<b>return</b>
documentation	This method initializes jScrollPaneProjectList.  @return the scrollpane.

#### Operation **JProjectManagement::getSelectedProjects**

owner	<b>JProjectManagement</b>
parameter	name direction type multiplicity default <b>return return int *</b>
ownedMember	<b>return</b>
documentation	This method checks which entries are marked in the list of projects and gives them back as indices.  @return array of indices.

#### Operation **JProjectManagement::initialize**

owner	<b>JProjectManagement</b>
parameter	name direction type multiplicity default <b>return return void</b>
ownedMember	<b>return</b>

#### Property **JProjectManagement::jButtonCopyProject**

owner	<b>JProjectManagement</b>
-------	---------------------------

#### Property **JProjectManagement::jButtonCreateProject**

owner	<b>JProjectManagement</b>
-------	---------------------------

#### Property **JProjectManagement::jButtonDeleteProject**

owner	<b>JProjectManagement</b>
-------	---------------------------

**Property JProjectManagement::jButtonOpenProject**

owner	<a href="#">JProjectManagement</a>
-------	------------------------------------

**Property JProjectManagement::jContentPane**

owner	<a href="#">JProjectManagement</a>
-------	------------------------------------

**Property JProjectManagement::jListProject**

owner	<a href="#">JProjectManagement</a>
-------	------------------------------------

**Operation JProjectManagement::JProjectManagement**

owner	<a href="#">JProjectManagement</a>
parameter	name direction type multiplicity default <b>model in M</b>
ownedMember	<b>model</b>
documentation	Create a new view which is connected to a appropriate model.  @param model the model which will be connected.

**Property JProjectManagement::jScrollPaneProjectList**

owner	<a href="#">JProjectManagement</a>
-------	------------------------------------

**Property JProjectManagement::serialVersionUID**

owner	<a href="#">JProjectManagement</a>
-------	------------------------------------

**Operation JProjectManagement::update**

owner	<a href="#">JProjectManagement</a>
parameter	name direction type multiplicity default <b>o in Observable arg in Object return return void</b>
ownedMember	<b>arg o return</b>

## Class JProjectView

diagram



hierarchy	<pre> classDiagram     JAbstractView&lt;M-&gt;M&gt; &lt; -- JProjectView   </pre>
owner	<b>projectview</b>
template parameters	name kind constrainingClassifier default <b>M Class ProjectModel</b>
ownedMember	<a href="#">get JButtonPictureSetContentAdd</a> <a href="#">get JButtonPictureSetContentDelete</a> <a href="#">get JButtonPictureSetContentRefresh</a> <a href="#">get JButtonPictureSetCopy</a> <a href="#">get JButtonPictureSetCreate</a> <a href="#">get JButtonPictureSetDelete</a> <a href="#">get JButtonProjectChange</a> <a href="#">get JButtonProjectSave</a> <a href="#">get JButtonReportCreate</a> <a href="#">get JButtonReportDelete</a> <a href="#">get JButtonReportOpen</a> <a href="#">get JPanelContent</a> <a href="#">get JPanelPictureProjectDescription</a> <a href="#">get JPanelPictureSet</a> <a href="#">get JPanelPictureSetActive</a> <a href="#">get JPanelPictureSetContent</a> <a href="#">get JPanelReport</a> <a href="#">get JPanelExif</a> <a href="#">get JPanelPictureSet</a> <a href="#">get JPanelPictureSetActive</a> <a href="#">get JPanelPictureSetContent</a> <a href="#">get JPanelPictureSetContentOptions</a> <a href="#">get JPanelPictureSetOptions</a> <a href="#">get JPanelProjectDescription</a> <a href="#">get JPanelProjectOptions</a> <a href="#">get JPanelReport</a> <a href="#">get JPanelReportOptions</a> <a href="#">get JPanelExif</a> <a href="#">get JScrollPaneProjectDescription</a> <a href="#">get jScrollPaneExif</a> <a href="#">get jScrollPaneProjectDescription</a> <a href="#">get jScrollPaneExif</a> <a href="#">get JTextFieldProjectName</a> <a href="#">get serialVersionUID</a> <a href="#">update</a>
general	<a href="#">JAbstractView&lt;M-&gt;M&gt;</a>
target of relation	ComponentRealization <a href="#">Implementierung</a>
documentation	<p>Represents the view for an active project.</p> <p>Sets all GUI Elements which are described in our Pflichtenheft.</p>

#### Operation **JProjectView::get JButtonPictureSetContentAdd**

owner	<a href="#">JProjectView</a>
parameter	name direction type multiplicity default <b>return return JButton</b>
ownedMember	<b>return</b>
documentation	<p>This method initializes JButtonPictureSetContentAdd.</p> <p>@return the button.</p>

#### Operation **JProjectView::get JButtonPictureSetContentDelete**

owner	<a href="#">JProjectView</a>
parameter	name direction type multiplicity default <b>return return JButton</b>
ownedMember	<b>return</b>
documentation	<p>This method initializes JButtonPictureSetContentDelete.</p> <p>@return the button.</p>

#### Operation **JProjectView::getJButtonPictureSetContentRefresh**

owner	<b>JProjectView</b>
parameter	name direction type multiplicity default <b>return return JButton</b>
ownedMember	<b>return</b>
documentation	This method initializes jButtonPictureSetContentRefresh.  @return the button.

#### Operation **JProjectView::getJButtonPictureSetCopy**

owner	<b>JProjectView</b>
parameter	name direction type multiplicity default <b>return return JButton</b>
ownedMember	<b>return</b>
documentation	This method initializes jButtonPictureSetCopy.  @return the button.

#### Operation **JProjectView::getJButtonPictureSetCreate**

owner	<b>JProjectView</b>
parameter	name direction type multiplicity default <b>return return JButton</b>
ownedMember	<b>return</b>
documentation	This method initializes jButtonPictureSetCreate.  @return the button.

#### Operation **JProjectView::getJButtonPictureSetDelete**

owner	<b>JProjectView</b>
parameter	name direction type multiplicity default <b>return return JButton</b>
ownedMember	<b>return</b>
documentation	This method initializes jButtonPictureSetDelete.  @return the button.

#### Operation **JProjectView::getJButtonProjectChange**

owner	<b>JProjectView</b>
parameter	name direction type multiplicity default <b>return return JButton</b>
ownedMember	<b>return</b>
documentation	This method initializes jButtonProjectChange.

	@return the button.
--	---------------------

#### Operation **JProjectView::getJButtonProjectSave**

owner	<a href="#">JProjectView</a>
parameter	name direction type multiplicity default <b>return return JButton</b>
ownedMember	<b>return</b>
documentation	This method initializes jButtonProjectSave.  @return the button.

#### Operation **JProjectView::getJButtonReportCreate**

owner	<a href="#">JProjectView</a>
parameter	name direction type multiplicity default <b>return return JButton</b>
ownedMember	<b>return</b>
documentation	This method initializes jButtonReportCreate.  @return the button.

#### Operation **JProjectView::getJButtonReportDelete**

owner	<a href="#">JProjectView</a>
parameter	name direction type multiplicity default <b>return return JButton</b>
ownedMember	<b>return</b>
documentation	This method initializes jButtonReportDelete.  @return the button.

#### Operation **JProjectView::getJButtonReportOpen**

owner	<a href="#">JProjectView</a>
parameter	name direction type multiplicity default <b>return return JButton</b>
ownedMember	<b>return</b>
documentation	This method initializes jButtonReportOpen.  @return the button.

#### Operation **JProjectView::getJContentPane**

owner	<a href="#">JProjectView</a>
parameter	name direction type multiplicity default <b>return return JPanel</b>

ownedMember	<b>return</b>
documentation	This method initializes jContentPane.  @return the button.

#### Operation **JProjectView::getJEditorPaneProjectDescription**

owner	<a href="#"><b>JProjectView</b></a>
parameter	name direction type multiplicity default <b>return return JEditorPane</b>
ownedMember	<b>return</b>
documentation	This method initializes jEditorPaneProjectDescription.  @return the editor pane.

#### Operation **JProjectView::getListPictureSet**

owner	<a href="#"><b>JProjectView</b></a>
parameter	name direction type multiplicity default <b>return return JList</b>
ownedMember	<b>return</b>
documentation	This method initializes jListPictureSet.  @return the list.

#### Operation **JProjectView::getListPictureSetActive**

owner	<a href="#"><b>JProjectView</b></a>
parameter	name direction type multiplicity default <b>return return JList</b>
ownedMember	<b>return</b>
documentation	This method initializes jListPictureSetActive.  @return the list.

#### Operation **JProjectView::getListPicturesetContent**

owner	<a href="#"><b>JProjectView</b></a>
parameter	name direction type multiplicity default <b>return return JList</b>
ownedMember	<b>return</b>
documentation	This method initializes jListPicturesetContent.  @return the list.

#### Operation **JProjectView::getListReport**

owner	<b>JProjectView</b>
parameter	name direction type multiplicity default <b>return return JList</b>
ownedMember	<b>return</b>
documentation	This method initializes jListReport.  @return the list.

#### Operation **JProjectView::get JPanelExif**

owner	<b>JProjectView</b>
parameter	name direction type multiplicity default <b>return return JPanel</b>
ownedMember	<b>return</b>
documentation	This method initializes jPanelExif.  @return the panel.

#### Operation **JProjectView::get JPanelPictureSet**

owner	<b>JProjectView</b>
parameter	name direction type multiplicity default <b>return return JPanel</b>
ownedMember	<b>return</b>
documentation	This method initializes jPanelPictureSet.  @return the panel.

#### Operation **JProjectView::get JPanelPictureSetActive**

owner	<b>JProjectView</b>
parameter	name direction type multiplicity default <b>return return JPanel</b>
ownedMember	<b>return</b>
documentation	This method initializes jPanelPictureSetActive.  @return the panel.

#### Operation **JProjectView::get JPanelPictureSetContent**

owner	<b>JProjectView</b>
parameter	name direction type multiplicity default <b>return return JPanel</b>
ownedMember	<b>return</b>
documentation	This method initializes jPanelPictureSetContent.

	@return the panel.
--	--------------------

#### Operation **JProjectView::getJPanelPictureSetContentOptions**

owner	<a href="#">JProjectView</a>
parameter	name direction type multiplicity default <b>return return JPanel</b>
ownedMember	<b>return</b>
documentation	This method initializes jPanelPictureSetContentOptions.  @return the panel.

#### Operation **JProjectView::getJPanelPictureSetOptions**

owner	<a href="#">JProjectView</a>
parameter	name direction type multiplicity default <b>return return JPanel</b>
ownedMember	<b>return</b>
documentation	This method initializes jPanelPictureSetOptions.  @return the panel.

#### Operation **JProjectView::getJPanelProjectDescription**

owner	<a href="#">JProjectView</a>
parameter	name direction type multiplicity default <b>return return JPanel</b>
ownedMember	<b>return</b>
documentation	This method initializes jPanelProjectDescription.  @return the panel.

#### Operation **JProjectView::getJPanelProjectOptions**

owner	<a href="#">JProjectView</a>
parameter	name direction type multiplicity default <b>return return JPanel</b>
ownedMember	<b>return</b>
documentation	This method initializes jPanelProjectOptions.  @return the panel.

#### Operation **JProjectView::getJPanelReport**

owner	<a href="#">JProjectView</a>
parameter	name direction type multiplicity default <b>return return JPanel</b>

ownedMember	<b>return</b>
documentation	This method initializes JPanelReport.  @return the panel.

#### Operation **JProjectView::getJPanelReportOptions**

owner	<a href="#"><b>JProjectView</b></a>
parameter	name direction type multiplicity default <b>return return JPanel</b>
ownedMember	<b>return</b>
documentation	This method initializes JPanelPictureSetContentOptions.  @return the panel.

#### Operation **JProjectView::getJScrollPaneExif**

owner	<a href="#"><b>JProjectView</b></a>
parameter	name direction type multiplicity default <b>return return JScrollPane</b>
ownedMember	<b>return</b>
documentation	This method initializes jScrollPaneExif.  @return the scrollpane.

#### Operation **JProjectView::getJScrollPaneProjectDescription**

owner	<a href="#"><b>JProjectView</b></a>
parameter	name direction type multiplicity default <b>return return JScrollPane</b>
ownedMember	<b>return</b>
documentation	This method initializes jScrollPaneProjectDescription.  @return the scrollpane.

#### Operation **JProjectView::getJTableExif**

owner	<a href="#"><b>JProjectView</b></a>
parameter	name direction type multiplicity default <b>return return JTable</b>
ownedMember	<b>return</b>
documentation	This method initializes jTableExif.  @return the table.

#### Operation **JProjectView::getJTextField projectName**

owner	<b>JProjectView</b>
parameter	name direction type multiplicity default <b>return return JTextField</b>
ownedMember	<b>return</b>
documentation	This method initializes jTextFieldProjectDescription.  @return the textfield.

#### Operation **JProjectView::getProjectDescription**

owner	<b>JProjectView</b>
parameter	name direction type multiplicity default <b>return return String</b>
ownedMember	<b>return</b>
documentation	Returns the project description.  @return the project description.

#### Operation **JProjectView::getProjectName**

owner	<b>JProjectView</b>
parameter	name direction type multiplicity default <b>return return String</b>
ownedMember	<b>return</b>
documentation	Returns the project name.  @return the project name.

#### Operation **JProjectView::getSelectedPictures**

owner	<b>JProjectView</b>
parameter	name direction type multiplicity default <b>return return int *</b>
ownedMember	<b>return</b>
documentation	This method checks which entries are marked in the Picture list of a PictureSet and gives them back as indices.  @return array of indices.

#### Operation **JProjectView::getSelectedPictureSetContents**

owner	<b>JProjectView</b>
parameter	name direction type multiplicity default <b>return return int *</b>
ownedMember	<b>return</b>
documentation	This method checks which entries are marked in the PictureSetContent list and gives them back as indices.

	@return array of indices.
--	---------------------------

#### Operation **JProjectView::getSelectedPictureSets**

owner	<a href="#">JProjectView</a>
parameter	name direction type multiplicity default <b>return return int *</b>
ownedMember	<b>return</b>
documentation	This method checks which entries are marked in the PictureSet list and gives them back as indices.  @return array of indices.

#### Operation **JProjectView::getSelectedReports**

owner	<a href="#">JProjectView</a>
parameter	name direction type multiplicity default <b>return return int *</b>
ownedMember	<b>return</b>
documentation	This method checks which entries are marked in the Report list and gives them back as indices.  @return array of indices.

#### Operation **JProjectView::initialize**

owner	<a href="#">JProjectView</a>
parameter	name direction type multiplicity default <b>return return void</b>
ownedMember	<b>return</b>
documentation	This method initializes this.  @return void

#### Property **JProjectView::jButtonPictureSetContentAdd**

owner	<a href="#">JProjectView</a>
-------	------------------------------

#### Property **JProjectView::jButtonPictureSetContentDelete**

owner	<a href="#">JProjectView</a>
-------	------------------------------

#### Property **JProjectView::jButtonPictureSetContentRefresh**

owner	<a href="#">JProjectView</a>
-------	------------------------------

Property **JProjectView::jButtonPictureSetCopy**

owner	<a href="#">JProjectView</a>
-------	------------------------------

Property **JProjectView::jButtonPictureSetCreate**

owner	<a href="#">JProjectView</a>
-------	------------------------------

Property **JProjectView::jButtonPictureSetDelete**

owner	<a href="#">JProjectView</a>
-------	------------------------------

Property **JProjectView::jButtonProjectChange**

owner	<a href="#">JProjectView</a>
-------	------------------------------

Property **JProjectView::jButtonProjectSave**

owner	<a href="#">JProjectView</a>
-------	------------------------------

Property **JProjectView::jButtonReportCreate**

owner	<a href="#">JProjectView</a>
-------	------------------------------

Property **JProjectView::jButtonReportDelete**

owner	<a href="#">JProjectView</a>
-------	------------------------------

Property **JProjectView::jButtonReportOpen**

owner	<a href="#">JProjectView</a>
-------	------------------------------

Property **JProjectView::jContentPane**

owner	<a href="#">JProjectView</a>
-------	------------------------------

Property **JProjectView::jEditorPaneProjectDescription**

owner	<a href="#">JProjectView</a>
-------	------------------------------

Property **JProjectView::jListPictureSet**

owner	<a href="#">JProjectView</a>
-------	------------------------------

Property **JProjectView::jListPictureSetActive**

owner	<a href="#">JProjectView</a>
-------	------------------------------

**Property [JProjectView::jListPictureSetContent](#)**

owner	<a href="#">JProjectView</a>
-------	------------------------------

**Property [JProjectView::jListReport](#)**

owner	<a href="#">JProjectView</a>
-------	------------------------------

**Property [JProjectView::jPanelExif](#)**

owner	<a href="#">JProjectView</a>
-------	------------------------------

**Property [JProjectView::jPanelPictureSet](#)**

owner	<a href="#">JProjectView</a>
-------	------------------------------

**Property [JProjectView::jPanelPictureSetActive](#)**

owner	<a href="#">JProjectView</a>
-------	------------------------------

**Property [JProjectView::jPanelPictureSetContent](#)**

owner	<a href="#">JProjectView</a>
-------	------------------------------

**Property [JProjectView::jPanelPictureSetContentOptions](#)**

owner	<a href="#">JProjectView</a>
-------	------------------------------

**Property [JProjectView::jPanelPictureSetOptions](#)**

owner	<a href="#">JProjectView</a>
-------	------------------------------

**Property [JProjectView::jPanelProjectDescription](#)**

owner	<a href="#">JProjectView</a>
-------	------------------------------

**Property [JProjectView::jPanelProjectOptions](#)**

owner	<a href="#">JProjectView</a>
-------	------------------------------

**Property [JProjectView::jPanelReport](#)**

owner	<a href="#">JProjectView</a>
-------	------------------------------

**Property [JProjectView::jPanelReportOptions](#)**

owner	<a href="#">JProjectView</a>
-------	------------------------------

**Operation JProjectView::JProjectView**

owner	<a href="#">JProjectView</a>
parameter	name direction type multiplicity default <b>model in M</b>
ownedMember	<b>model</b>
documentation	Creates a project view connected with an appropriate model.

**Property JProjectView::jScrollPaneExif**

owner	<a href="#">JProjectView</a>
-------	------------------------------

**Property JProjectView::jScrollPaneProjectDescription**

owner	<a href="#">JProjectView</a>
-------	------------------------------

**Property JProjectView::jTableExif**

owner	<a href="#">JProjectView</a>
-------	------------------------------

**Property JProjectView::jTextFieldProjectName**

owner	<a href="#">JProjectView</a>
-------	------------------------------

**Property JProjectView::serialVersionUID**

owner	<a href="#">JProjectView</a>
-------	------------------------------

**Operation JProjectView::update**

owner	<a href="#">JProjectView</a>
parameter	name direction type multiplicity default <b>o in Observable arg in Object return void</b>
ownedMember	<b>arg o return</b>

## Class JRelPathHelper

diagram	<pre> <b>JRelPathHelper</b> +--&gt; &lt;&lt;constructor&gt;&gt; JRelPathHelper() +--&gt; getAbsolutePath(<b>in refFile:File</b>, <b>in relPath:String</b>):String +--&gt; getAbsolutePath(<b>in refPath:String</b>, <b>in relPath:String</b>):String +--&gt; getRelativePath(<b>in refFile:File</b>, <b>in fullPath:String</b>):String +--&gt; getRelativePath(<b>in refPath:String</b>, <b>in fullPath:String</b>):String </pre>
owner	<b>files</b>
ownedMember	<b>getAbsolutePath</b> <b>getAbsolutePath</b> <b>getRelativePath</b> <b>getRelativePath</b> <b>JRelPathHelper</b>
target of relation	ComponentRealization <b>Implementierung</b>
documentation	@author Reiner

### Operation JRelPathHelper::getAbsolutePath

owner	<b>JRelPathHelper</b>
parameter	name direction type multiplicity default <b>refFile</b> <b>in File</b> <b>relPath</b> <b>in String</b> <b>return return String</b>
ownedMember	<b>refFile</b> <b>relPath</b> <b>return</b>

### Operation JRelPathHelper::getAbsolutePath

owner	<b>JRelPathHelper</b>
parameter	name direction type multiplicity default <b>refPath</b> <b>in String</b> <b>relPath</b> <b>in String</b> <b>return return String</b>
ownedMember	<b>refPath</b> <b>relPath</b> <b>return</b>

### Operation JRelPathHelper::getRelativePath

owner	<b>JRelPathHelper</b>
parameter	name direction type multiplicity default <b>refFile</b> <b>in File</b> <b>fullPath</b> <b>in String</b> <b>return return String</b>
ownedMember	<b>fullPath</b> <b>refFile</b> <b>return</b>

### Operation JRelPathHelper::getRelativePath

owner	<b>JRelPathHelper</b>
parameter	name direction type multiplicity default <b>refPath</b> <b>in String</b> <b>fullPath</b> <b>in String</b> <b>return return String</b>
ownedMember	<b>fullPath</b> <b>refPath</b> <b>return</b>

### Operation JRelPathHelper::JRelPathHelper

owner	<b>JRelPathHelper</b>
documentation	Creates a new instance of JRelPathHelper

	on	
--	----	--

## Class JRelPathHelper

diagram	<pre> classDiagram     class JRelPathHelper {         &lt;&lt;constructor&gt;&gt; JRelPathHelper()         getAbsolutePath(in refFile:File, in relPath:String):String         getAbsolutePath(in refPath:String, in relPath:String):String         getRelativePath(in refFile:File, in fullPath:String):String         getRelativePath(in refPath:String, in fullPath:String):String     }   </pre>
owner	<a href="#">files</a>
ownedMember	<a href="#">getAbsolutePath</a> <a href="#">getAbsolutePath</a> <a href="#">getRelativePath</a> <a href="#">getRelativePath</a> <a href="#">JRelPathHelper</a>
target of relation	ComponentRealization <a href="#">Implementierung</a>
documentation	@author Reiner

### Operation JRelPathHelper::getAbsolutePath

owner	<a href="#">JRelPathHelper</a>
parameter	name direction type multiplicity default <b>refFile</b> in File <b>relPath</b> in String return return String
ownedMember	<b>refFile</b> <b>relPath</b> return

### Operation JRelPathHelper::getAbsolutePath

owner	<a href="#">JRelPathHelper</a>
parameter	name direction type multiplicity default <b>refPath</b> in String <b>relPath</b> in String return return String
ownedMember	<b>refPath</b> <b>relPath</b> return

### Operation JRelPathHelper::getRelativePath

owner	<a href="#">JRelPathHelper</a>
parameter	name direction type multiplicity default <b>refFile</b> in File <b>fullPath</b> in String return return String
ownedMember	<b>fullPath</b> <b>refFile</b> return

### Operation JRelPathHelper::getRelativePath

owner	<a href="#">JRelPathHelper</a>
parameter	name direction type multiplicity default <b>refPath</b> in String <b>fullPath</b> in String return return String
ownedMember	<b>fullPath</b> <b>refPath</b> return

### Operation JRelPathHelper::JRelPathHelper

owner	<a href="#">JRelPathHelper</a>
documentation	Creates a new instance of JRelPathHelper

	on	
--	----	--

## Class JReportConfig

diagram	<p>The diagram shows the class <b>JReportConfig</b> with the following details:</p> <ul style="list-style-type: none"> <li><b>Attributes:</b> <ul style="list-style-type: none"> <li><code>serialVersionUID:long=1L</code></li> <li><code>reportCompilation:AbstractReportCompilation&lt;M-&gt;AbstractReportModel&gt;</code></li> <li><code>tabbedPane:JTabbedPane</code></li> <li><code>basic: JPanel</code></li> <li><code>mainPanel: JPanel</code></li> <li><code>mysize:int[]={800,600}</code></li> </ul> </li> <li><b>Operations:</b> <ul style="list-style-type: none"> <li><code>JReportConfig() (Operation)</code></li> <li><code>setReportType() (Operation)</code></li> <li><code>update() (Operation)</code></li> </ul> </li> </ul>
hierarchy	<pre> classDiagram     class JAbstractReportUtil {         &lt;&lt;M-&gt;M, V-&gt;V&gt;     }     class JReportConfig {         &lt;&lt;Annotations, constructor&gt;&gt; JReportConfig(in model:M, in view:V)         getTabbedPane():JTabbedPane         initialize():void         &lt;&lt;Annotations&gt;&gt; setReportType(in reportconfig:AbstractReportCompilation&lt;M-&gt;?&gt;):void         &lt;&lt;Annotations&gt;&gt; update(in model:Observable, in argument:Object):void     }     JAbstractReportUtil &lt; -- JReportConfig   </pre>
owner	<b>reportmanagement</b>
template parameters	name kind constrainingClassifier default <b>M</b> Class <a href="#">AbstractReportModel</a> <b>V</b> Class <a href="#">AbstractReportCompilation&lt;M-&gt;M&gt;</a>
ownedMember	<a href="#">basic getJTabbedPane</a> <a href="#">initialize JReportConfig</a> <a href="#">mainpanel</a> <a href="#">mysize</a> <a href="#">reportCompilation</a> <a href="#">serialVersionUID</a> <a href="#">setReportType</a> <a href="#">tabbedPane</a> <a href="#">update</a>
general	<b>JAbstractReportUtil&lt;M-&gt;M,V-&gt;V&gt;</b>
target of relation	ComponentRealization <a href="#">Implementierung</a>
documentation	<p>This class represents the report configuration utility for an existing report. It offers the user the functionality to alter an existing report in any possible way.</p> <p>@author David Kaufman</p> <p>@param &lt;M&gt; @param &lt;V&gt;</p>

## Property JReportConfig::basic

owner	<a href="#">JReportConfig</a>
-------	-------------------------------

## Operation JReportConfig::getJTabbedPane

owner	<a href="#">JReportConfig</a>
parameter	name direction type multiplicity default <b>return return JTabbedPane</b>
ownedMember	<b>return</b>

## Operation JReportConfig::initialize

owner	<a href="#">JReportConfig</a>
parameter	name direction type multiplicity default <b>return return void</b>
ownedMember	<b>return</b>

**Operation JReportConfig::JReportConfig**

owner	<a href="#">JReportConfig</a>
parameter	name direction type multiplicity default <b>model in M view in V</b>
ownedMember	<b>model view</b>

**Property JReportConfig::mainpanel**

owner	<a href="#">JReportConfig</a>
-------	-------------------------------

**Property JReportConfig::mysize**

owner	<a href="#">JReportConfig</a>
-------	-------------------------------

**Property JReportConfig::reportCompilation**

owner	<a href="#">JReportConfig</a>
-------	-------------------------------

**Property JReportConfig::serialVersionUID**

owner	<a href="#">JReportConfig</a>
-------	-------------------------------

**Operation JReportConfig::setReportType**

owner	<a href="#">JReportConfig</a>
parameter	name direction type multiplicity default <b>reportconfig in AbstractReportCompilation&lt;M-&gt;?&gt;</b> return return void
ownedMember	<b>reportconfig return</b>

**Property JReportConfig::tabbedPane**

owner	<a href="#">JReportConfig</a>
-------	-------------------------------

**Operation JReportConfig::update**

owner	<a href="#">JReportConfig</a>
parameter	name direction type multiplicity default <b>model in Observable argument in Object return return void</b>
ownedMember	<b>argument model return</b>

## Class JReportWizard

diagram	<p>The diagram shows the JReportWizard class with its attributes and operations. Attributes include serialVersionUID, reportCompilation, wizardcounter, and basic. Operations include initialize(), nextPanel(), previousPanel(), update(), and setReportType(). The class is constrained by M Class AbstractReportModel and V Class AbstractReportCompilation&lt;M-&gt;M&gt;.</p>
hierarchy	<p>JReportWizard inherits from JAbstractReportUtil&lt;M-&gt;M,V-&gt;V&gt;.</p>
owner	<a href="#">reportmanagement</a>
template parameters	name kind constrainingClassifier default M Class <a href="#">AbstractReportModel</a> V Class <a href="#">AbstractReportCompilation&lt;M-&gt;M&gt;</a>
ownedMember	<a href="#">basic initialize JReportWizard nextPanel previousPanel reportCompilation serialVersionUID setReportType update wizardcounter</a>
general	<a href="#">JAbstractReportUtil&lt;M-&gt;M,V-&gt;V&gt;</a>
target of relation	ComponentRealization <a href="#">Implementierung</a>
documentation	<p>This class represents the report creation wizard. It is responsible for creating a ReportModel.</p> <p>@author David Kaufman</p> <p>@param &lt;M&gt; @param &lt;V&gt;</p>

### Property JReportWizard::basic

owner	<a href="#">JReportWizard</a>
-------	-------------------------------

### Operation JReportWizard::initialize

owner	<a href="#">JReportWizard</a>
parameter	name direction type multiplicity default <b>return return void</b>
ownedMember	<b>return</b>

### Operation JReportWizard::JReportWizard

owner	<a href="#">JReportWizard</a>
documentation	<p>Starts the wizard utility on a specified report configuration</p> <p>@param reportconfig the report configuration to operate on</p>

#### Operation **JReportWizard::nextPanel**

owner	<a href="#">JReportWizard</a>
parameter	name direction type multiplicity default <b>return return void</b>
ownedMember	<b>return</b>
documentation	Switches the current panel to the next panel if possible

#### Operation **JReportWizard::previousPanel**

owner	<a href="#">JReportWizard</a>
parameter	name direction type multiplicity default <b>return return void</b>
ownedMember	<b>return</b>
documentation	Switches the current panel to the previous panel if possible

#### Property **JReportWizard::reportCompilation**

owner	<a href="#">JReportWizard</a>
-------	-------------------------------

#### Property **JReportWizard::serialVersionUID**

owner	<a href="#">JReportWizard</a>
documentation	

#### Operation **JReportWizard::setReportType**

owner	<a href="#">JReportWizard</a>
parameter	name direction type multiplicity default <b>reportconfig in AbstractReportCompilation&lt;M-&gt;?&gt; return return void</b>
ownedMember	<b>reportconfig return</b>

#### Operation **JReportWizard::update**

owner	<a href="#">JReportWizard</a>
parameter	name direction type multiplicity default <b>model in Observable argument in Object return return void</b>
ownedMember	<b>argument model return</b>

#### Property **JReportWizard::wizardcounter**

owner	<a href="#">JReportWizard</a>
-------	-------------------------------



## Class JTableDiagram

diagram	<p><b>JTableDiagram</b></p> <ul style="list-style-type: none"> <li><b>Attributes:</b> <ul style="list-style-type: none"> <li><code>serialVersionUID:long=1L</code> (final)</li> <li><code>model:M</code> (constructor)</li> <li><code>getDiagramScreenshot():BufferedImage</code> (annotation)</li> </ul> </li> <li><b>Operations:</b> <ul style="list-style-type: none"> <li><code>getDiagramScreenshot() (Operation)</code> (modifiers = Override)</li> </ul> </li> </ul>
hierarchy	<pre> classDiagram     JAbstractDiagram&lt;M-M&gt; &lt; -- JTableDiagram   </pre>
owner	<b>diagrams</b>
template parameters	name kind constrainingClassifier default <b>M</b> Class <b>TableModel</b>
ownedMember	<b>getDiagramScreenshot</b> <b>JTableDiagram</b> <b>serialVersionUID</b>
general	<b>JAbstractDiagram&lt;M-&gt;M&gt;</b>
target of relation	ComponentRealization <b>Implementierung</b>
documentation	<p>This class implements how the TableModel is to be drawn.</p> <p>@author David Kaufman</p> <p>@param &lt;M&gt;</p>

## Operation JTableDiagram::getDiagramScreenshot

owner	<b>JTableDiagram</b>
parameter	name direction type multiplicity default <b>return return BufferedImage</b>
ownedMember	<b>return</b>

## Operation JTableDiagram::JTableDiagram

owner	<b>JTableDiagram</b>
parameter	name direction type multiplicity default <b>model in M</b>
ownedMember	<b>model</b>
documentation	<p>Constructor</p> <p>@param abstractModel the model from which the drawing information is taken</p>

## Property JTableDiagram::serialVersionUID

owner	<b>JTableDiagram</b>
-------	----------------------



## Class JTextDiagram

	<p><b>JTextDiagram</b></p> <ul style="list-style-type: none"> <li><b>Attributes:</b> <ul style="list-style-type: none"> <li><code>serialVersionUID:long=1L</code> (final)</li> <li><code>model:M</code> (constructor)</li> <li><code>getDiagramScreenshot():BufferedImage</code> (annotation)</li> </ul> </li> <li><b>Operations:</b> <ul style="list-style-type: none"> <li><code>getDiagramScreenshot() (Operation)</code> (modifiers = Override)</li> </ul> </li> </ul>
diagram	
hierarchy	<pre> JAbstractDiagram&lt;M-&gt;M&gt;           +-- JTextDiagram   </pre>
owner	<b>diagrams</b>
template parameters	name kind constrainingClassifier default <b>M</b> <b>Class TextModel</b>
ownedMember	<b>getDiagramScreenshot</b> <b>JTextDiagram</b> <b>serialVersionUID</b>
general	<b>JAbstractDiagram&lt;M-&gt;M&gt;</b>
target of relation	ComponentRealization <b>Implementierung</b>
documentation	<p>This class implements how the TextModel is to be drawn.</p> <p>@author David Kaufman</p> <p>@param &lt;M&gt;</p>

## Operation JTextDiagram::getDiagramScreenshot

owner	<b>JTextDiagram</b>
parameter	name direction type multiplicity default <b>return return BufferedImage</b>
ownedMember	<b>return</b>

## Operation JTextDiagram::JTextDiagram

owner	<b>JTextDiagram</b>
parameter	name direction type multiplicity default <b>model in M</b>
ownedMember	<b>model</b>
documentation	<p>Constructor</p> <p>@param abstractModel the model from which the drawing information is taken</p>

## Property JTextDiagram::serialVersionUID

owner	<b>JTextDiagram</b>
-------	---------------------



## Class JTiffHeader

diagram	<pre> &lt;&lt;final&gt;&gt; <b>JTiffHeader</b>  <b>m_ifdOffset:int=0</b> <b>m_offset:long=0</b> <b>m_bIntel:boolean=true</b>  <b>getOffset():long</b> <b>getIFDOffset():int</b> <b>isIntel():boolean</b> <b>read(<i>in</i> file:RandomAccessFile):boolean</b> &lt;&lt;constructor&gt;&gt; <b>JTiffHeader()</b> </pre>
owner	<b>jexifviewer</b>
ownedMember	<b>getIFDOffset</b> <b>getOffset</b> <b>isIntel</b> <b>JTiffHeader</b> <b>m_bIntel</b> <b>m_ifdOffset</b> <b>m_offset</b> <b>read</b>
target of relation	ComponentRealization <b>Implementierung</b>
documentation	@author reiner

## Operation JTiffHeader::getIFDOffset

owner	<b>JTiffHeader</b>
parameter	name direction type multiplicity default <b>return</b> <b>return int</b>
ownedMember	<b>return</b>

## Operation JTiffHeader::getOffset

owner	<b>JTiffHeader</b>
parameter	name direction type multiplicity default <b>return</b> <b>return long</b>
ownedMember	<b>return</b>

## Operation JTiffHeader::isIntel

owner	<b>JTiffHeader</b>
parameter	name direction type multiplicity default <b>return</b> <b>return boolean</b>
ownedMember	<b>return</b>

## Operation JTiffHeader::JTiffHeader

owner	<b>JTiffHeader</b>
documentation	Creates a new instance of JTiffHeader

**Property JTiffHeader::m\_bIntel**

owner	<a href="#">JTiffHeader</a>
-------	-----------------------------

**Property JTiffHeader::m\_ifdOffset**

owner	<a href="#">JTiffHeader</a>
-------	-----------------------------

**Property JTiffHeader::m\_offset**

owner	<a href="#">JTiffHeader</a>
-------	-----------------------------

**Operation JTiffHeader::read**

owner	<a href="#">JTiffHeader</a>
parameter	name direction type multiplicity default <b>file in RandomAccessFile return return boolean</b>
ownedMember	<b>file return</b>
documentation	reads the TIFF header from a file @param file the RandomAccessFile @return true if success else false

## Class JTiffHeader

diagram	<pre> &lt;&lt;final&gt;&gt; <b>JTiffHeader</b>  <b>m_ifdOffset:int=0</b> <b>m_offset:long=0</b> <b>m_bIntel:boolean=true</b>  <b>getOffset():long</b> <b>getIFDOffset():int</b> <b>isIntel():boolean</b> <b>read(<i>in</i> file:RandomAccessFile):boolean</b> &lt;&lt;constructor&gt;&gt; <b>JTiffHeader()</b> </pre>
owner	<b>jexifviewer</b>
ownedMember	<b>getIFDOffset</b> <b>getOffset</b> <b>isIntel</b> <b>JTiffHeader</b> <b>m_bIntel</b> <b>m_ifdOffset</b> <b>m_offset</b> <b>read</b>
target of relation	ComponentRealization <b>Implementierung</b>
documentation	@author reiner

## Operation JTiffHeader::getIFDOffset

owner	<b>JTiffHeader</b>
parameter	name direction type multiplicity default <b>return</b> <b>return int</b>
ownedMember	<b>return</b>

## Operation JTiffHeader::getOffset

owner	<b>JTiffHeader</b>
parameter	name direction type multiplicity default <b>return</b> <b>return long</b>
ownedMember	<b>return</b>

## Operation JTiffHeader::isIntel

owner	<b>JTiffHeader</b>
parameter	name direction type multiplicity default <b>return</b> <b>return boolean</b>
ownedMember	<b>return</b>

## Operation JTiffHeader::JTiffHeader

owner	<b>JTiffHeader</b>
documentation	Creates a new instance of JTiffHeader

**Property JTiffHeader::m\_bIntel**

owner	<a href="#">JTiffHeader</a>
-------	-----------------------------

**Property JTiffHeader::m\_ifdOffset**

owner	<a href="#">JTiffHeader</a>
-------	-----------------------------

**Property JTiffHeader::m\_offset**

owner	<a href="#">JTiffHeader</a>
-------	-----------------------------

**Operation JTiffHeader::read**

owner	<a href="#">JTiffHeader</a>
parameter	name direction type multiplicity default <b>file in RandomAccessFile return return boolean</b>
ownedMember	<b>file return</b>
documentation	reads the TIFF header from a file @param file the RandomAccessFile @return true if success else false

## Class JWilcoxon

diagram	<pre> classDiagram     class JWilcoxon {         &lt;&lt;final&gt;&gt; serialVersionUID:long=1L         &lt;&lt;constructor&gt;&gt; JWilcoxon(in titel:String, in icon:Icon, in tip:String)         getStatus():boolean         getTestType():WilcoxonTestType         getStatisticalSignificance():float     }     ImageObserver &lt; -- JWilcoxon     MenuContainer &lt; -- JWilcoxon     Serializable &lt; -- JWilcoxon     Component &lt; -- JWilcoxon     Container &lt; -- JWilcoxon     JComponent &lt; -- JWilcoxon     JAbstractSinglePanel &lt; -- JWilcoxon     </pre>
hierarchy	<pre> classDiagram     class ImageObserver     class MenuContainer     class Serializable     class Component     class Container     class JComponent     class JAbstractSinglePanel     class JWilcoxon      ImageObserver &lt; -- JWilcoxon     MenuContainer &lt; -- JWilcoxon     Serializable &lt; -- JWilcoxon     Component &lt; -- JWilcoxon     Container &lt; -- JWilcoxon     JComponent &lt; -- JWilcoxon     JAbstractSinglePanel &lt; -- JWilcoxon     </pre>
owner	<a href="#">reportmanagement</a>
ownedMember	<a href="#">getStatisticalSignificance</a> <a href="#">getStatus</a> <a href="#">getTestType</a> <a href="#">JWilcoxon</a> <a href="#">serialVersionUID</a>
general	<a href="#">JAbstractSinglePanel</a>
target of relation	ComponentRealization <a href="#">Implementierung</a>
documentation	<p>This class represents the panel where the user is able to configure the wilcoxon test for the boxplot report.</p> <p>@author David Kaufman</p>

## Operation JWilcoxon::getStatisticalSignificance

owner	<a href="#">JWilcoxon</a>
parameter	name direction type multiplicity default <b>return return float</b>
ownedMember	<b>return</b>

## Operation JWilcoxon::getStatus

owner	<a href="#">JWilcoxon</a>
parameter	name direction type multiplicity default <b>return return boolean</b>
ownedMember	<b>return</b>

**Operation JWilcoxon::getTestType**

owner	<a href="#">JWilcoxon</a>
parameter	name direction type multiplicity default <b>return return WilcoxonTestType</b>
ownedMember	<b>return</b>

**Operation JWilcoxon::JWilcoxon**

owner	<a href="#">JWilcoxon</a>
parameter	name direction type multiplicity default <b>titel in String icon in Icon tip in String</b>
ownedMember	<b>icon tip titel</b>

**Property JWilcoxon::serialVersionUID**

owner	<a href="#">JWilcoxon</a>
-------	---------------------------

## Class Klient

diagram	
owner	<b>Kompositum</b>
ownedMember	<b>main</b>
target of relation	ComponentRealization <b>Implementierung</b>

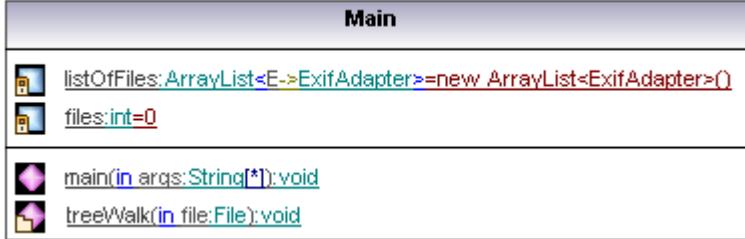
## Operation Klient::main

owner	<b>Klient</b>
parameter	name direction type multiplicity default <b>args in String * return return void</b>
ownedMember	<b>args return</b>
documentation	<b>@param args</b>

## Package Kompositum

owner	<a href="#">studien</a>
ownedMember	<a href="#">ArrayList&lt;E-&gt;?&gt;</a> <a href="#">Directory</a> <a href="#">Klient</a> <a href="#">List&lt;E-&gt;Picture&gt;</a> <a href="#">List&lt;E-&gt;PictureContainer&gt;</a> <a href="#">Picture</a> <a href="#">PictureContainer</a> <a href="#">PictureSet</a>

## Class Main

diagram	
owner	<a href="#">exifAdapter</a>
ownedMember	<a href="#">files</a> <a href="#">listOfFiles</a> <a href="#">main</a> <a href="#">treeWalk</a>
target of relation	ComponentRealization <a href="#">Implementierung</a>

## Property Main::files

owner	<a href="#">Main</a>
-------	----------------------

## Property Main::listOfFiles

owner	<a href="#">Main</a>
-------	----------------------

## Operation Main::main

owner	<a href="#">Main</a>
parameter	name direction type multiplicity default <b>args in String * return return void</b>
ownedMember	<b>args return</b>
documentation	@param args

## Operation Main::treeWalk

owner	<a href="#">Main</a>
parameter	name direction type multiplicity default <b>file in File return return void</b>
ownedMember	<b>file return</b>

## Class MyPictureListCellRenderer

diagram	<pre> classDiagram     class ListCellRenderer     class MyPictureListCellRenderer {         &lt;&lt;ListCellRenderer&gt;&gt;     } </pre>
hierarchy	<pre> classDiagram     class ListCellRenderer     class MyPictureListCellRenderer {         &lt;&lt;ListCellRenderer&gt;&gt;     }     ListCellRenderer &lt; -- MyPictureListCellRenderer </pre>
owner	<a href="#">projectview</a>
ownedMember	<a href="#">defaultRenderer getListCellRendererComponent</a>
implemented Interfaces	<a href="#">ListCellRenderer</a>
source of relation	InterfaceRealization <a href="#">ListCellRenderer</a>
target of relation	ComponentRealization <a href="#">Implementierung</a>
documentation	This nested class renders a picture cell for the picture list of an active picture container.

### Property MyPictureListCellRenderer::defaultRenderer

owner	<a href="#">MyPictureListCellRenderer</a>
-------	---

### Operation MyPictureListCellRenderer::getListCellRendererComponent

owner	<a href="#">MyPictureListCellRenderer</a>
parameter	name direction type multiplicity default list in JList value in Object index in int isSelected in boolean cellHasFocus in boolean return return Component
ownedMember	<b>cellHasFocus index isSelected list return value</b>
documentation	<p>Renders the cell.</p> <p> <b>@param list</b>  the JList we're painting.  <b>@param value</b>  the value returned by list.getModel().getElementAt(index).  <b>@param index</b>  the cells index.  <b>@param isSelected</b>  true if the specified cell was selected.  <b>@param cellHasFocus</b>  true if the specified cell has the focus.  <b>@return</b> the representation of the cell. </p>



## Class MyPictureSetContentListCellRenderer

diagram	<pre> classDiagram     class ListCellRenderer     class MyPictureSetContentListCellRenderer {         &lt;&lt;ListCellRenderer&gt;&gt;         defaultRenderer() --&gt; DefaultListCellRenderer         getListCellRendererComponent(in list: JList, in value: Object, in index: int, in isSelected: boolean, in cellHasFocus: boolean) --&gt; Component     } </pre>
hierarchy	<pre> classDiagram     class ListCellRenderer     class MyPictureSetContentListCellRenderer {         &lt;&lt;ListCellRenderer&gt;&gt;         defaultRenderer() --&gt; DefaultListCellRenderer         getListCellRendererComponent(in list: JList, in value: Object, in index: int, in isSelected: boolean, in cellHasFocus: boolean) --&gt; Component     } </pre>
owner	<a href="#">projectview</a>
ownedMember	<a href="#">defaultRenderer getListCellRendererComponent</a>
implemented Interfaces	<a href="#">ListCellRenderer</a>
source of relation	InterfaceRealization <a href="#">ListCellRenderer</a>
target of relation	ComponentRealization <a href="#">Implementierung</a>
documentation	This nested class renders a picture set content cell for the picture set content list of an active project.

## Property MyPictureSetContentListCellRenderer::defaultRenderer

owner	<a href="#">MyPictureSetContentListCellRenderer</a>
-------	---

## Operation MyPictureSetContentListCellRenderer::getListCellRendererComponent

owner	<a href="#">MyPictureSetContentListCellRenderer</a>
parameter	name direction type multiplicity default list in JList value in Object index in int isSelected in boolean cellHasFocus in boolean return return Component
ownedMember	<b>cellHasFocus index isSelected list return value</b>
documentation	<p>Renders the cell.</p> <p> <b>@param list</b>  the JList we're painting.  <b>@param value</b>  the value returned by list.getModel().getElementAt(index).  <b>@param index</b>  the cells index.  <b>@param isSelected</b>  true if the specified cell was selected.  <b>@param cellHasFocus</b>  true if the specified cell has the focus.  <b>@return</b> the representation of the cell. </p>



## Class MyPictureSetListCellRenderer

diagram	<pre> classDiagram     class ListCellRenderer     class MyPictureSetListCellRenderer {         &lt;&lt;ListCellRenderer&gt;&gt;     } </pre>
hierarchy	<pre> classDiagram     class ListCellRenderer     class MyPictureSetListCellRenderer {         &lt;&lt;ListCellRenderer&gt;&gt;     }     ListCellRenderer &lt; -- MyPictureSetListCellRenderer </pre>
owner	<a href="#">projectview</a>
ownedMember	<a href="#">defaultRenderer getListCellRendererComponent</a>
implemented interfaces	<a href="#">ListCellRenderer</a>
source of relation	InterfaceRealization <a href="#">ListCellRenderer</a>
target of relation	ComponentRealization <a href="#">Implementierung</a>
documentation	This nested class renders a picture set cell for the picture set list.

### Property MyPictureSetListCellRenderer::defaultRenderer

owner	<a href="#">MyPictureSetListCellRenderer</a>
-------	--

### Operation MyPictureSetListCellRenderer::getListCellRendererComponent

owner	<a href="#">MyPictureSetListCellRenderer</a>
parameter	name direction type multiplicity default list in <a href="#">JList</a> value in <a href="#">Object</a> index in <a href="#">int</a> isSelected in <a href="#">boolean</a> cellHasFocus in <a href="#">boolean</a> return return <a href="#">Component</a>
ownedMember	<a href="#">cellHasFocus index isSelected list return value</a>
documentation	<p>Renders the cell.</p> <p> <b>@param list</b>  the <a href="#">JList</a> we're painting.  <b>@param value</b>  the value returned by <code>list.getModel().getElementAt(index)</code>.  <b>@param index</b>  the cells index.  <b>@param isSelected</b>  true if the specified cell was selected.  <b>@param cellHasFocus</b>  true if the specified cell has the focus.  <b>@return</b> the representation of the cell. </p>



## Class MyProjectListCellRenderer

diagram	<pre> classDiagram     class ListCellRenderer {         &lt;&gt; defaultRenderer: DefaultListCellRenderer = new DefaultListCellRenderer()         &lt;&gt; getListCellRendererComponent(list: JList, value: Object, index: int, isSelected: boolean, cellHasFocus: boolean): Component     }     class MyProjectListCellRenderer {         &lt;&gt; ListCellRenderer     } </pre>
hierarchy	<pre> classDiagram     class ListCellRenderer {         &lt;&gt; defaultRenderer: DefaultListCellRenderer = new DefaultListCellRenderer()         &lt;&gt; getListCellRendererComponent(list: JList, value: Object, index: int, isSelected: boolean, cellHasFocus: boolean): Component     }     class MyProjectListCellRenderer {         &lt;&gt; ListCellRenderer     } </pre>
owner	<a href="#">projectmanagement</a>
ownedMember	<a href="#">defaultRenderer getListCellRendererComponent</a>
implemented Interfaces	<a href="#">ListCellRenderer</a>
source of relation	InterfaceRealization <a href="#">ListCellRenderer</a>
target of relation	ComponentRealization <a href="#">Implementierung</a>
documentation	This nested class renders a project cell for the project list.

## Property MyProjectListCellRenderer::defaultRenderer

owner	<a href="#">MyProjectListCellRenderer</a>
-------	---

## Operation MyProjectListCellRenderer::getListCellRendererComponent

owner	<a href="#">MyProjectListCellRenderer</a>
parameter	name direction type multiplicity default list in JList value in Object index in int isSelected in boolean cellHasFocus in boolean return return Component
ownedMember	<b>cellHasFocus index isSelected list return value</b>
documentation	<p>Renders the cell.</p> <p> <b>@param list</b>  the JList we're painting.  <b>@param value</b>  the value returned by list.getModel().getElementAt(index).  <b>@param index</b>  the cells index.  <b>@param isSelected</b>  true if the specified cell was selected.  <b>@param cellHasFocus</b>  true if the specified cell has the focus.  <b>@return</b> the representation of the cell. </p>



## Class MyReportListCellRenderer

diagram	<pre> classDiagram     class MyReportListCellRenderer {         &lt;&lt;MyReportListCellRenderer&gt;&gt;         &lt;&lt;ListCellRenderer&gt;&gt;         &lt;&lt;DefaultListCellRenderer&gt;&gt;         &lt;&lt;Component&gt;&gt;     } </pre>
hierarchy	<pre> classDiagram     class ListCellRenderer     class MyReportListCellRenderer {         &lt;&lt;ListCellRenderer&gt;&gt;     }     ListCellRenderer &lt; -- MyReportListCellRenderer </pre>
owner	<a href="#">projectview</a>
ownedMember	<a href="#">defaultRenderer getListCellRendererComponent</a>
implemented interfaces	<a href="#">ListCellRenderer</a>
source of relation	InterfaceRealization <a href="#">ListCellRenderer</a>
target of relation	ComponentRealization <a href="#">Implementierung</a>
documentation	This nested class renders a report cell for the report list of an active project.

## Property MyReportListCellRenderer::defaultRenderer

owner	<a href="#">MyReportListCellRenderer</a>
-------	--

## Operation MyReportListCellRenderer::getListCellRendererComponent

owner	<a href="#">MyReportListCellRenderer</a>
parameter	name direction type multiplicity default list in <a href="#">JList</a> value in <a href="#">Object</a> index in <a href="#">int</a> isSelected in <a href="#">boolean</a> cellHasFocus in <a href="#">boolean</a> return return <a href="#">Component</a>
ownedMember	<a href="#">cellHasFocus index isSelected list return value</a>
documentation	<p>Renders the cell.</p> <p> @param list  the <a href="#">JList</a> we're painting.  @param value  the value returned by <code>list.getModel().getElementAt(index)</code>.  @param index  the cells index.  @param isSelected  true if the specified cell was selected.  @param cellHasFocus  true if the specified cell has the focus.  @return the representation of the cell. </p>



## Class MyTableCellRenderer

diagram	<p>The diagram shows the class <b>MyTableCellRenderer</b> with its constructor and one operation. The constructor is annotated with <code>&lt;&lt;final&gt;&gt; serialVersionUID:long=1L</code>. The operation is <code>getTableCellRendererComponent(in table:JTable, in value:Object, in isSelected:boolean, in hasFocus:boolean, in rowIndex:int, in vColIndex:int):Component</code>.</p>
hierarchy	<pre> classDiagram     class MyTableCellRenderer {         &lt;&lt;final&gt;&gt; serialVersionUID:long=1L         getTableCellRendererComponent(in table:JTable, in value:Object, in isSelected:boolean, in hasFocus:boolean, in rowIndex:int, in vColIndex:int):Component     }     class ImageObserver     class MenuContainer     class Serializable     class Component     class Container     class JComponent     class SwingConstants     class Accessible     class JLabel     class TableCellRenderer      MyTableCellRenderer &lt; -- JLabel     JLabel &lt; -- TableCellRenderer     ImageObserver &lt; -- Component     MenuContainer &lt; -- Component     Serializable &lt; -- Component     Serializable &lt; -- JComponent     Container &lt; -- JComponent     JComponent &lt; -- SwingConstants     JComponent &lt; -- Accessible     </pre> <p>The hierarchy diagram illustrates the inheritance path of <b>MyTableCellRenderer</b>. It starts with <b>ImageObserver</b>, <b>MenuContainer</b>, and <b>Serializable</b> as superclasses of <b>Component</b>. <b>Component</b> is a superclass of <b>Container</b>, which is a superclass of <b>JComponent</b>. <b>JComponent</b> implements <b>SwingConstants</b> and <b>Accessible</b>. Finally, <b>JLabel</b> is a subclass of <b>JComponent</b>, and <b>MyTableCellRenderer</b> is a subclass of <b>JLabel</b>.</p>
owner	<a href="#">projectview</a>
ownedMember	<a href="#">getTableCellRendererComponent serialVersionUID</a>
general	<a href="#">JLabel</a>
implemented interfaces	<a href="#">TableCellRenderer</a>
source of relation	InterfaceRealization <a href="#">TableCellRenderer</a>
target of relation	ComponentRealization <a href="#">Implementierung</a>
documentation	This nested class renders a table cell for the table which shows the exif parameters of an active picture.

## Operation MyTableCellRenderer::getTableCellRendererComponent

owner	<a href="#">MyTableCellRenderer</a>
parameter	name direction type multiplicity default <b>table</b> in <b>JTable</b> <b>value</b> in <b>Object</b> <b>isSelected</b> in <b>boolean</b> <b>hasFocus</b> in <b>boolean</b> <b>rowIndex</b> in <b>int</b> <b>vColIndex</b> in <b>int</b> return <b>Component</b>
ownedMember	<b>hasFocus</b> <b>isSelected</b> return <b>rowIndex</b> <b>table</b> <b>value</b> <b>vColIndex</b>
documentation	<p>Renders the cell.</p> <p> <b>@param list</b>  the <b>JList</b> we're painting.  <b>@param value</b>  the value returned by <code>list.getModel().getElementAt(index)</code>.  <b>@param index</b>  the cells index.  <b>@param isSelected</b>  true if the specified cell was selected.  <b>@param cellHasFocus</b>  true if the specified cell has the focus.  <b>@return</b> the representation of the cell. </p>

Property **MyTableCellRenderer::serialVersionUID**

owner	<a href="#">MyTableCellRenderer</a>
-------	-------------------------------------

## Class Object2XML

diagram	
owner	<a href="#">XML</a>
ownedMember	<a href="#">main</a>
target of relation	ComponentRealization <a href="#">Implementierung</a>
documentation	This shows how to use JAXB to create a content tree and populate it and marshal to an xml file

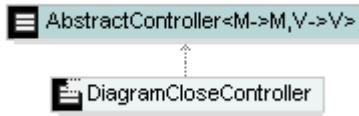
## Operation Object2XML::main

owner	<a href="#">Object2XML</a>
parameter	name direction type multiplicity default <b>args</b> <b>in</b> <b>String</b> * <b>return</b> <b>return void</b>
ownedMember	<b>args return</b>

## Package **Package1**

owner	<a href="#">Root</a>
ownedMember	<a href="#">org</a>
source of relation	ProfileApplication <b>Java Profile</b>

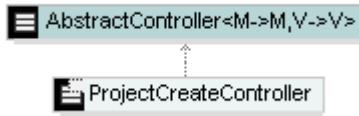
## Class AbstractController<M->M,V->V>

diagram	
hierarchy	
owner	<a href="#">diagrams</a>
template	<a href="#">AbstractController</a>
templ.par.su bstitution	formal actual <b>M MV V</b>
specific	<a href="#">DiagramCloseController</a>

## Class AbstractController<M->M,V->V>

diagram	
hierarchy	
owner	<a href="#">diagrams</a>
template	<a href="#">AbstractController</a>
templ.par.su bstitution	formal actual <b>M MV V</b>
specific	<a href="#">DiagramExportAsJPGController</a>

## Class AbstractController<M->M,V->V>

diagram	
hierarchy	
owner	<a href="#">projectmanagement</a>
template	<a href="#">AbstractController</a>
templ.par.su bstitution	formal actual <b>M MV V</b>
specific	<a href="#">ProjectCreateController</a>

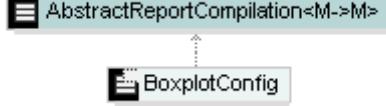
## Class AbstractReportCompilation<M->M>

diagram	
owner	<a href="#">reportmanagement</a>
template	<a href="#">AbstractReportCompilation</a>
templ.par.su bstitution	formal actual <b>M M</b>

### Class AbstractReportCompilation<M->M>

diagram	
hierarchy	
owner	<a href="#">reportmanagement</a>
template	<a href="#">AbstractReportCompilation</a>
templ.par.su bstitution	formal actual <b>M M</b>
specific	<a href="#">TableConfig</a>

## Class AbstractReportCompilation<M->M>

diagram	
hierarchy	
owner	<a href="#">reportmanagement</a>
template	<a href="#">AbstractReportCompilation</a>
templ.par.su bstitution	formal actual <b>M M</b>
specific	<a href="#">BoxplotConfig</a>

## Class ArrayList<E>->Axis>

diagram	
owner	<a href="#">reportmanagement</a>
template	<b>ArrayList</b>
templ.par.su bstitution	formal actual <b>E Axis</b>
typedEleme nts	Class <a href="#">JParameters</a> Operation <a href="#">getAxes</a>

## Class ArrayList<E>-JAbstractSinglePanel>

diagram	
owner	<a href="#">reportmanagement</a>
template	<a href="#">ArrayList</a>
templ.par.su bstitution	formal actual <b>E JAbstractSinglePanel</b>
typedEleme nts	Class <a href="#">AbstractReportCompilation</a> Property <a href="#">registeredPanels</a> Operation <a href="#">getRegisteredPanels</a>

## Class JAbstractDiagram<M>->AbstractReportModel>

diagram	
owner	<a href="#">reportmanagement</a>
template	<a href="#">JAbstractDiagram</a>
templ.par.substitution	formal actual M <a href="#">AbstractReportModel</a>
typedElements	Enumeration <a href="#">ReportHelper</a> Operation <a href="#">getDiagram</a>

Class **JAbstractReportUtil<M->AbstractReportModel,V->AbstractReportCompilation<M->AbstractReportModel>>**

diagram	
owner	<a href="#">reportmanagement</a>
template	<a href="#">JAbstractReportUtil</a>
templ.par.su bstitution	formal actual <b>M</b> <a href="#">AbstractReportModel</a> <b>V</b> <a href="#">AbstractReportCompilation&lt;M-&gt;AbstractReportModel&gt;</a>
typedElems	Enumeration <a href="#">ReportHelper</a> Property <a href="#">currentReportUtil</a>

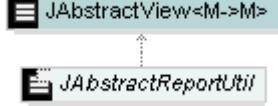
## Class JAbstractReportUtil<M->M,V->V>

diagram	
hierarchy	 <pre>graph TD; A[JAbstractReportUtil&lt;M-&gt;M,V-&gt;V&gt;] --&gt; B[JReportConfig]</pre>
owner	<a href="#">reportmanagement</a>
template	<a href="#">JAbstractReportUtil</a>
templ.par.su bstitution	formal actual <b>M MV V</b>
specific	<a href="#">JReportConfig</a>

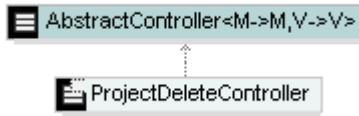
## Class JAbstractReportUtil<M->M,V->V>

diagram	
hierarchy	 <pre>graph TD; A[JAbstractReportUtil&lt;M-&gt;M,V-&gt;V&gt;] --&gt; B[JReportWizard]</pre>
owner	<a href="#">reportmanagement</a>
template	<a href="#">JAbstractReportUtil</a>
templ.par.su bstitution	formal actual <b>M MV V</b>
specific	<a href="#">JReportWizard</a>

## Class JAbstractView<M->M>

diagram	
hierarchy	
owner	<b>reportmanagement</b>
template	<b>JAbstractView</b>
templ.par.su bstitution	formal actual <b>M M</b>
specific	<b>JAbstractReportUtil</b>

## Class AbstractController<M->M,V->V>

diagram	
hierarchy	 <pre>graph TD; AbstractController["AbstractController&lt;M-&gt;M,V-&gt;V&gt;"] --&gt; ProjectDeleteController["ProjectDeleteController"]</pre>
owner	<a href="#">projectmanagement</a>
template	<a href="#">AbstractController</a>
templ.par.su bstitution	formal actual <b>M MV V</b>
specific	<a href="#">ProjectDeleteController</a>

## Class **ArrayList<E>->ExifAdapter>**

diagram	
owner	<a href="#">exifAdapter</a>
template	<a href="#">ArrayList</a>
templ.par.su bstitution	formal actual <b>E</b> <a href="#">ExifAdapter</a>
typedEleme nts	Class <a href="#">Main</a> Property <a href="#">listOfFiles</a>

## Class ArrayList<E->JExifTag>

diagram	
owner	<a href="#">jexifviewer</a>
template	<b>ArrayList</b>
templ.par.su bstitution	formal actual <b>E JExifTag</b>
typedEleme nts	Class <a href="#">JIfd</a> Property <a href="#">m_tagArray</a>

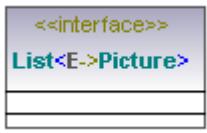
## Class ArrayList<E>->String>

diagram	
owner	<a href="#">files</a>
template	<b>ArrayList</b>
templ.par.su bstitution	formal actual <b>E String</b>
typedEleme nts	Class <a href="#">JMyFileFilter</a> Property <a href="#">m_extList</a>

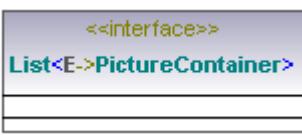
## Class ArrayList<E->?>

diagram	
owner	<a href="#">Kompositum</a>
template	<a href="#">ArrayList</a>
templ.par.su bstitution	formal actual E ?
typedEleme nts	Class <a href="#">Directory</a> Operation <a href="#">getItems</a> Class <a href="#">Picture</a> Operation <a href="#">getItems</a> Interface <a href="#">PictureContainer</a> Operation <a href="#">getItems</a> Class <a href="#">PictureSet</a> Operation <a href="#">getItems</a>

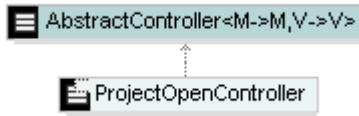
## Interface List<E>-Picture

diagram	 A UML interface diagram showing a box labeled '<<interface>>' containing the text 'List<E>-Picture'.
owner	<a href="#">Kompositum</a>
template	<a href="#">List</a>
templ.par.su bstitution	formal actual <b>E Picture</b>
typedEleme nts	Class <a href="#">Directory</a> Property <a href="#">pictures</a>

## Interface **List<E>PictureContainer**

diagram	 A UML interface diagram showing a light blue rounded rectangle labeled '<<interface>>' at the top, followed by the name 'List<E>PictureContainer' in bold black text.
owner	<a href="#">Kompositum</a>
template	<a href="#">List</a>
templ.par.su bstitution	formal actual <b>E PictureContainer</b>
typedEleme nts	Class <a href="#">PictureSet</a> Property <a href="#">childs</a>

## Class AbstractController<M->M,V->V>

diagram	
hierarchy	 <pre>classDiagram     class AbstractController&lt;M-&gt;M,V-&gt;V&gt;     class ProjectOpenController     ProjectOpenController &lt; -- AbstractController</pre>
owner	<a href="#">projectmanagement</a>
template	<a href="#">AbstractController</a>
templ.par.su bstitution	formal actual <b>M MV V</b>
specific	<a href="#">ProjectOpenController</a>

## Class JProjectManagement<M->?>

diagram	
owner	<a href="#">projectmanagement</a>
template	<a href="#">JProjectManagement</a>
templ.par.su bstitution	formal actual M ?

## Class JProjectManagement<M->M>

diagram	
owner	<a href="#">projectmanagement</a>
template	<a href="#">JProjectManagement</a>
templ.par.su bstitution	formal actual <b>M M</b>

## Class JProjectManagement<M->M>

diagram	
owner	<a href="#">projectmanagement</a>
template	<a href="#">JProjectManagement</a>
templ.par.su bstitution	formal actual <b>M M</b>

## Class JProjectManagement<M->M>

diagram	
owner	<a href="#">projectmanagement</a>
template	<a href="#">JProjectManagement</a>
templ.par.su bstitution	formal actual <b>M M</b>

## Class AbstractController<M->M,V->V>

diagram	
hierarchy	
owner	<a href="#">projectview</a>
template	<a href="#">AbstractController</a>
templ.par.su bstitution	formal actual <b>M MV V</b>
specific	<a href="#">PictureListClickOnController</a>

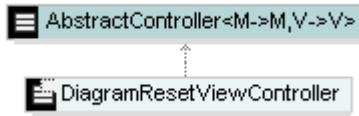
## Class AbstractController<M->M,V->V>

diagram	
hierarchy	
owner	<a href="#">projectview</a>
template	<a href="#">AbstractController</a>
templ.par.su bstitution	formal actual <b>M MV V</b>
specific	<a href="#">PictureSetContentListAddController</a>

## Class AbstractController<M->M,V->V>

diagram	
hierarchy	
owner	<a href="#">projectview</a>
template	<a href="#">AbstractController</a>
templ.par.su bstitution	formal actual <b>M MV V</b>
specific	<a href="#">PictureSetContentListClickOnController</a>

## Class AbstractController<M->M,V->V>

diagram	
hierarchy	
owner	<a href="#">diagrams</a>
template	<a href="#">AbstractController</a>
templ.par.su bstitution	formal actual <b>M MV V</b>
specific	<a href="#">DiagramResetViewController</a>

## Class AbstractController<M->M,V->V>

diagram	
hierarchy	
owner	<a href="#">projectview</a>
template	<a href="#">AbstractController</a>
templ.par.su bstitution	formal actual <b>M MV V</b>
specific	<a href="#">PictureSetContentListDeleteController</a>

## Class AbstractController<M->M,V->V>

diagram	
hierarchy	
owner	<a href="#">projectview</a>
template	<a href="#">AbstractController</a>
templ.par.su bstitution	formal actual <b>M MV V</b>
specific	<a href="#">PictureSetContentListDropController</a>

## Class AbstractController<M->M,V->V>

diagram	
hierarchy	
owner	<a href="#">projectview</a>
template	<a href="#">AbstractController</a>
templ.par.su bstitution	formal actual <b>M MV V</b>
specific	<a href="#">PictureSetContentListRefreshController</a>

## Class AbstractController<M->M,V->V>

diagram	
hierarchy	
owner	<a href="#">projectview</a>
template	<a href="#">AbstractController</a>
templ.par.su bstitution	formal actual <b>M MV V</b>
specific	<a href="#">PictureSetListClickOnController</a>

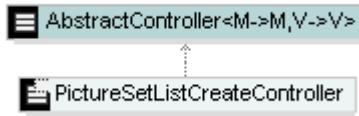
## Class AbstractController<M->M,V->V>

diagram	
hierarchy	
owner	<a href="#">projectview</a>
template	<a href="#">AbstractController</a>
templ.par.su bstitution	formal actual <b>M MV V</b>
specific	<a href="#">PictureSetListDeleteController</a>

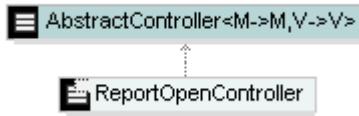
## Class AbstractController<M->M,V->V>

diagram	
hierarchy	
owner	<a href="#">projectview</a>
template	<a href="#">AbstractController</a>
templ.par.su bstitution	formal actual <b>M MV V</b>
specific	<a href="#">PictureSetListCopyController</a>

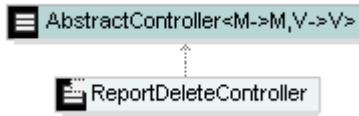
## Class AbstractController<M->M,V->V>

diagram	
hierarchy	
owner	<a href="#">projectview</a>
template	<a href="#">AbstractController</a>
templ.par.su bstitution	formal actual <b>M MV V</b>
specific	<a href="#">PictureSetListCreateController</a>

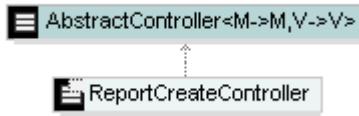
## Class AbstractController<M->M,V->V>

diagram	
hierarchy	
owner	<a href="#">projectview</a>
template	<a href="#">AbstractController</a>
templ.par.su bstitution	formal actual <b>M MV V</b>
specific	<a href="#">ReportOpenController</a>

## Class AbstractController<M->M,V->V>

diagram	
hierarchy	
owner	<a href="#">projectview</a>
template	<a href="#">AbstractController</a>
templ.par.su bstitution	formal actual <b>M MV V</b>
specific	<a href="#">ReportDeleteController</a>

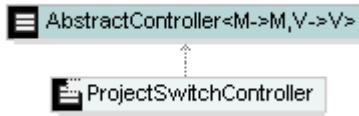
## Class AbstractController<M->M,V->V>

diagram	
hierarchy	
owner	<a href="#">projectview</a>
template	<a href="#">AbstractController</a>
templ.par.su bstitution	formal actual <b>M MV V</b>
specific	<a href="#">ReportCreateController</a>

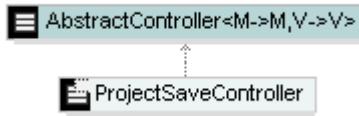
### Class AbstractController<M->M,V->V>

diagram	
hierarchy	
owner	<a href="#">diagrams</a>
template	<a href="#">AbstractController</a>
templ.par.su bstitution	formal actual <b>M MV V</b>
specific	<a href="#">ReportEditController</a>

## Class AbstractController<M->M,V->V>

diagram	
hierarchy	
owner	<a href="#">projectview</a>
template	<a href="#">AbstractController</a>
templ.par.su bstitution	formal actual <b>M MV V</b>
specific	<a href="#">ProjectSwitchController</a>

## Class AbstractController<M->M,V->V>

diagram	
hierarchy	 <pre>graph TD; AbstractController["AbstractController&lt;M-&gt;M,V-&gt;V&gt;"] --&gt; ProjectSaveController["ProjectSaveController"]</pre>
owner	<a href="#">projectview</a>
template	<a href="#">AbstractController</a>
templ.par.su bstitution	formal actual <b>M MV V</b>
specific	<a href="#">ProjectSaveController</a>

## Class AbstractController<M->M,V->V>

diagram	
hierarchy	
owner	<a href="#">projectview</a>
template	<a href="#">AbstractController</a>
templ.par.su bstitution	formal actual <b>M MV V</b>
specific	<a href="#">ProjectEditNameController</a>

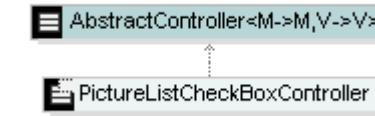
## Class AbstractController<M->M,V->V>

diagram	
hierarchy	
owner	<a href="#">projectview</a>
template	<a href="#">AbstractController</a>
templ.par.su bstitution	formal actual <b>M MV V</b>
specific	<a href="#">ProjectEditDescriptionController</a>

## Class AbstractController<M->M,V->V>

diagram	
hierarchy	
owner	<a href="#">projectview</a>
template	<a href="#">AbstractController</a>
templ.par.su bstitution	formal actual <b>M MV V</b>
specific	<a href="#">PictureSetListDragController</a>

## Class AbstractController<M->M,V->V>

diagram	
hierarchy	
owner	<a href="#">projectview</a>
template	<a href="#">AbstractController</a>
templ.par.su bstitution	formal actual <b>M MV V</b>
specific	<a href="#">PictureListCheckBoxController</a>

## Class JProjectView<M->M>

diagram	
owner	<a href="#">projectview</a>
template	<a href="#">JProjectView</a>
templ.par.su bstitution	formal actual <b>M M</b>

## Class JProjectView<M->M>

diagram	
owner	<a href="#">projectview</a>
template	<a href="#">JProjectView</a>
templ.par.su bstitution	formal actual <b>M M</b>

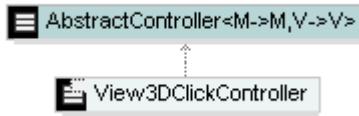
## Class JProjectView<M->M>

diagram	
owner	<a href="#">projectview</a>
template	<a href="#">JProjectView</a>
templ.par.su bstitution	formal actual <b>M M</b>

## Class JProjectView<M->M>

diagram	
owner	<a href="#">projectview</a>
template	<a href="#">JProjectView</a>
templ.par.su bstitution	formal actual <b>M M</b>

### Class AbstractController<M->M,V->V>

diagram	
hierarchy	
owner	<a href="#">diagrams</a>
template	<a href="#">AbstractController</a>
templ.par.su bstitution	formal actual <b>M MV V</b>
specific	<a href="#">View3DClickController</a>

## Class JProjectView<M->M>

diagram	
owner	<a href="#">projectview</a>
template	<a href="#">JProjectView</a>
templ.par.su bstitution	formal actual <b>M M</b>

## Class JProjectView<M->M>

diagram	
owner	<a href="#">projectview</a>
template	<a href="#">JProjectView</a>
templ.par.su bstitution	formal actual <b>M M</b>

## Class JProjectView<M->M>

diagram	
owner	<a href="#">projectview</a>
template	<a href="#">JProjectView</a>
templ.par.su bstitution	formal actual <b>M M</b>

## Class JProjectView<M->M>

diagram	
owner	<a href="#">projectview</a>
template	<a href="#">JProjectView</a>
templ.par.su bstitution	formal actual <b>M M</b>

## Class JProjectView<M->M>

diagram	
owner	<a href="#">projectview</a>
template	<a href="#">JProjectView</a>
templ.par.su bstitution	formal actual <b>M M</b>

## Class JProjectView<M->M>

diagram	
owner	<a href="#">projectview</a>
template	<a href="#">JProjectView</a>
templ.par.su bstitution	formal actual <b>M M</b>

## Class JProjectView<M->M>

diagram	
owner	<a href="#">projectview</a>
template	<a href="#">JProjectView</a>
templ.par.su bstitution	formal actual <b>M M</b>

## Class JProjectView<M->M>

diagram	
owner	<a href="#">projectview</a>
template	<a href="#">JProjectView</a>
templ.par.su bstitution	formal actual <b>M M</b>

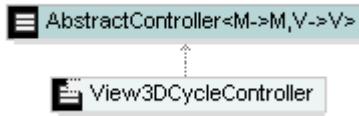
## Class JProjectView<M->M>

diagram	
owner	<a href="#">projectview</a>
template	<a href="#">JProjectView</a>
templ.par.su bstitution	formal actual <b>M M</b>

## Class JProjectView<M->M>

diagram	
owner	<a href="#">projectview</a>
template	<a href="#">JProjectView</a>
templ.par.su bstitution	formal actual <b>M M</b>

## Class AbstractController<M->M,V->V>

diagram	
hierarchy	
owner	<a href="#">diagrams</a>
template	<a href="#">AbstractController</a>
templ.par.su bstitution	formal actual <b>M MV V</b>
specific	<a href="#">View3DCycleController</a>

## Class JProjectView<M->M>

diagram	
owner	<a href="#">projectview</a>
template	<a href="#">JProjectView</a>
templ.par.su bstitution	formal actual <b>M M</b>

## Class JProjectView<M->M>

diagram	
owner	<a href="#">projectview</a>
template	<a href="#">JProjectView</a>
templ.par.su bstitution	formal actual <b>M M</b>

## Class JProjectView<M->M>

diagram	
owner	<a href="#">projectview</a>
template	<a href="#">JProjectView</a>
templ.par.su bstitution	formal actual <b>M M</b>

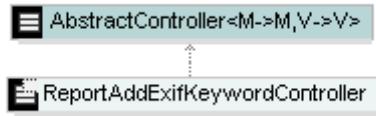
## Class JProjectView<M->M>

diagram	
owner	<a href="#">projectview</a>
template	<a href="#">JProjectView</a>
templ.par.su bstitution	formal actual <b>M M</b>

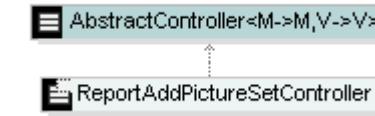
## Class JProjectView<M->M>

diagram	
owner	<a href="#">projectview</a>
template	<a href="#">JProjectView</a>
templ.par.su bstitution	formal actual <b>M M</b>

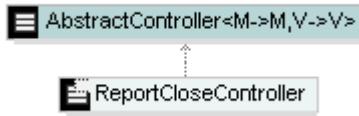
## Class AbstractController<M->M,V->V>

diagram	
hierarchy	
owner	<b>reportmanagement</b>
template	<b>AbstractController</b>
templ.par.su bstitution	formal actual <b>M MV V</b>
specific	<b>ReportAddExifKeywordController</b>

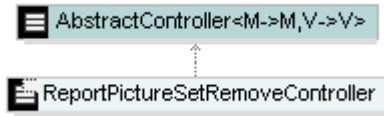
## Class AbstractController<M->M,V->V>

diagram	
hierarchy	
owner	<a href="#">reportmanagement</a>
template	<a href="#">AbstractController</a>
templ.par.su bstitution	formal actual <b>M MV V</b>
specific	<a href="#">ReportAddPictureSetController</a>

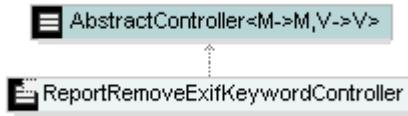
## Class AbstractController<M->M,V->V>

diagram	
hierarchy	
owner	<a href="#">reportmanagement</a>
template	<a href="#">AbstractController</a>
templ.par.su bstitution	formal actual <b>M MV V</b>
specific	<a href="#">ReportCloseController</a>

## Class AbstractController<M->M,V->V>

diagram	
hierarchy	
owner	<a href="#">reportmanagement</a>
template	<a href="#">AbstractController</a>
templ.par.su bstitution	formal actual <b>M MV V</b>
specific	<a href="#">ReportPictureSetRemoveController</a>

## Class AbstractController<M->M,V->V>

diagram	
hierarchy	
owner	<b>reportmanagement</b>
template	<b>AbstractController</b>
templ.par.su bstitution	formal actual <b>M MV V</b>
specific	<b>ReportRemoveExifKeywordController</b>

## Class JAbstract3DView<M->?>

diagram	
owner	<a href="#">diagrams</a>
template	<a href="#">JAbstract3DView</a>
templ.par.su bstitution	formal actual M ?

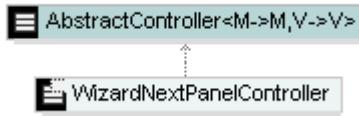
## Class AbstractController<M->M,V->V>

diagram	
hierarchy	
owner	<b>reportmanagement</b>
template	<b>AbstractController</b>
templ.par.su bstitution	formal actual <b>M MV V</b>
specific	<b>ReportSaveController</b>

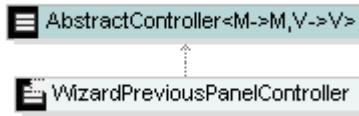
## Class AbstractController<M->M,V->V>

diagram	
hierarchy	
owner	<a href="#">reportmanagement</a>
template	<a href="#">AbstractController</a>
templ.par.su bstitution	formal actual <b>M MV V</b>
specific	<a href="#">WizardCloseController</a>

## Class AbstractController<M->M,V->V>

diagram	
hierarchy	
owner	<a href="#">reportmanagement</a>
template	<a href="#">AbstractController</a>
templ.par.su bstitution	formal actual <b>M MV V</b>
specific	<a href="#">WizardNextPanelController</a>

## Class AbstractController<M->M,V->V>

diagram	
hierarchy	
owner	<a href="#">reportmanagement</a>
template	<a href="#">AbstractController</a>
templ.par.su bstitution	formal actual <b>M MV V</b>
specific	<a href="#">WizardPreviousPanelController</a>

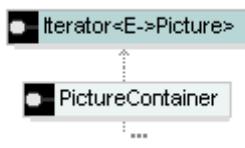
## Class AbstractReportCompilation<M->AbstractReportModel>

diagram	
owner	<a href="#">reportmanagement</a>
template	<a href="#">AbstractReportCompilation</a>
templ.par.su bstitution	formal actual M <a href="#">AbstractReportModel</a>

## Class JReportWizard<M->?,V->?>

diagram	
owner	<a href="#">reportmanagement</a>
template	<a href="#">JReportWizard</a>
templ.par.su bstitution	formal actual M ?V ?

## Interface Iterator<E->Picture>

diagram	
hierarchy	 <pre>classDiagram PictureContainer &lt; -- Iterator&lt;E-&gt;Picture</pre>
owner	<a href="#">picturemanagement</a>
template	<b>Iterator</b>
templ.par.su bstitution	formal actual <b>E Picture</b>
specific	<a href="#">PictureContainer</a>

## Interface `List<E->?`

diagram	
owner	<a href="#">picturemanagement</a>
template	<b>List</b>
templ.par.su bstitution	formal actual <b>E</b> ?
typedEleme nts	Class <a href="#">Directory</a> Operation <a href="#">getItems</a> Class <a href="#">Picture</a> Operation <a href="#">getItems</a> Interface <a href="#">PictureContainer</a> Operation <a href="#">getItems</a>

## Interface List<E>-Picture

diagram	 A UML interface diagram showing a box labeled '<<interface>>' containing the text 'List<E>-Picture'.
owner	<a href="#">picturemanagement</a>
template	<a href="#">List</a>
templ.par.substitution	formal actual <b>E Picture</b>
typedElements	Class <a href="#">Directory</a> Property <a href="#">pictures</a>

## Interface `List<E>PictureContainer`

diagram	 A UML interface diagram showing a rounded rectangle labeled <<interface>> at the top, followed by the name <code>List&lt;E&gt;PictureContainer</code> in bold.
owner	<a href="#">picturemanagement</a>
template	<b>List</b>
templ.par.substitution	formal actual <b>E PictureContainer</b>
typedElements	Class <a href="#">PictureSet</a> Property <a href="#">child</a> s Operation <a href="#">getChild</a> s <a href="#">getItems</a> <a href="#">setChild</a> s

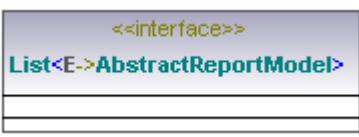
## Class JAbstractDiagram<M->?>

diagram	
owner	<a href="#">diagrams</a>
template	<a href="#">JAbstractDiagram</a>
templ.par.su bstitution	formal actual <b>M ?</b>

## Interface List<E>->ProjectEntry

diagram	 A UML interface diagram showing a box labeled '<<interface>>' containing the text 'List<E>->ProjectEntry'.
owner	<a href="#">projectmanagement</a>
template	<a href="#">List</a>
templ.par.su bstitution	formal actual <b>E ProjectEntry</b>
typedEleme nts	Class <a href="#">ProjectManagementModel</a> Property <a href="#">projectList</a> Operation <a href="#">getProjectList</a> <a href="#">ProjectManagementModel</a> <a href="#">setProjectlist</a>

**Interface List<E>AbstractReportModel>**

diagram	 A UML class diagram showing an interface named "List<E>AbstractReportModel". The interface has a single method, indicated by a small empty box below it.
owner	<a href="#">projectview</a>
template	<b>List</b>
templ.par.su bstitution	formal actual <b>E AbstractReportModel</b>
typedEleme nts	Class <a href="#">ProjectModel</a> Property <a href="#">reportList</a> Operation <a href="#">getInstance</a> <a href="#">ProjectModel</a>

## Interface List<E>-PictureSet>

diagram	
owner	<a href="#">projectview</a>
template	<b>List</b>
templ.par.su bstitution	formal actual <b>E PictureSet</b>
typedEleme nts	Class <a href="#">ProjectModel</a> Property <a href="#">pictureSetList</a> Operation <a href="#">getInstance</a> <a href="#">ProjectModel</a>

## Class ArrayList<E>->PictureContainer

diagram	
owner	<a href="#">reportmanagement</a>
template	<a href="#">ArrayList</a>
templ.par.su bstitution	formal actual <a href="#">E PictureContainer</a>
typedEleme nts	Class <a href="#">AbstractDoubleAxesModel</a> Operation <a href="#">AbstractDoubleAxesModel</a> Class <a href="#">AbstractReportModel</a> Property <a href="#">pictureContainer</a> Operation <a href="#">AbstractReportModel getPictureContainer setPictureContainer</a> Class <a href="#">AbstractSingleAxisModel</a> Operation <a href="#">AbstractSingleAxisModel</a> Class <a href="#">AbstractTripleAxesModel</a> Operation <a href="#">AbstractTripleAxesModel</a> Class <a href="#">BoxplotModel</a> Operation <a href="#">BoxplotModel</a> Class <a href="#">Cluster3DModel</a> Operation <a href="#">Cluster3DModel</a> Class <a href="#">Histogram2DModel</a> Operation <a href="#">Histogram2DModel</a> Class <a href="#">Histogram3DModel</a> Operation <a href="#">Histogram3DModel</a> Class <a href="#">TableModel</a> Operation <a href="#">TableModel</a> Class <a href="#">TextModel</a> Operation <a href="#">TextModel</a>

## Class ArrayList<E->JExifTag>

diagram	
owner	<a href="#">jexifviewer</a>
template	<b>ArrayList</b>
templ.par.su bstitution	formal actual <b>E JExifTag</b>
typedEleme nts	Class <a href="#">JIfd</a> Property <a href="#">m_tagArray</a>

## Class ArrayList<E>->String>

diagram	
owner	<a href="#">files</a>
template	<b>ArrayList</b>
templ.par.su bstitution	formal actual <b>E String</b>
typedEleme nts	Class <a href="#">JMyFileFilter</a> Property <a href="#">m_extList</a>

## Interface `List<E>ProjectEntry`

diagram	 A UML class diagram showing an interface named "List<E>ProjectEntry". The interface has one generalization arrow pointing to a concrete class below it.
owner	<a href="#">utils</a>
template	<a href="#">List</a>
templ.par.substitution	formal actual <a href="#">E ProjectEntry</a>
typedElements	Class <a href="#">FileHandler</a> Operation <a href="#">scanProjectDirectory</a>

## Class JAbstract2DDiagram<M->M>

diagram	
hierarchy	
owner	<a href="#">diagrams</a>
template	<a href="#">JAbstract2DDiagram</a>
templ.par.su bstitution	formal actual <b>M M</b>
specific	<a href="#">JBoxplot</a>

## Class JAbstract2DDiagram<M->M>

diagram	
hierarchy	
owner	<a href="#">diagrams</a>
template	<a href="#">JAbstract2DDiagram</a>
templ.par.su bstitution	formal actual <b>M M</b>
specific	<a href="#">JHistogram2D</a>

## Class JAbstract3DDiagram<M->M>

diagram	
hierarchy	
owner	<a href="#">diagrams</a>
template	<a href="#">JAbstract3DDiagram</a>
templ.par.su bstitution	formal actual <b>M M</b>
specific	<a href="#">JCcluster3D</a>

## Class JCluster3D<M->?>

diagram	
owner	<a href="#">diagrams</a>
template	<a href="#">JCluster3D</a>
templ.par.su bstitution	formal actual <b>M ?</b>

## Class JAbstract3DDiagram<M->M>

diagram	
hierarchy	
owner	<a href="#">diagrams</a>
template	<a href="#">JAbstract3DDiagram</a>
templ.par.su bstitution	formal actual <b>M M</b>
specific	<a href="#">JHistogram3D</a>

## Class JAbstract3DView<M->M>

diagram	
hierarchy	 <pre>graph TD; JAbstract3DView --&gt; JAbstract2DDiagram</pre>
owner	<a href="#">diagrams</a>
template	<a href="#">JAbstract3DView</a>
templ.par.su bstitution	formal actual <b>M M</b>
specific	<a href="#">JAbstract2DDiagram</a>

## Class JAbstract3DView<M->M>

diagram	
hierarchy	 <pre>graph TD; JAbstract3DView["JAbstract3DView&lt;M-&gt;M&gt;"] --&gt; JAbstract3DDiagram["JAbstract3DDiagram"]</pre>
owner	<a href="#">diagrams</a>
template	<a href="#">JAbstract3DView</a>
templ.par.su bstitution	formal actual <b>M M</b>
specific	<a href="#">JAbstract3DDiagram</a>

## Class JAbstractDiagram<M->M>

diagram	
hierarchy	
owner	<a href="#">diagrams</a>
template	<a href="#">JAbstractDiagram</a>
templ.par.su bstitution	formal actual <b>M M</b>
specific	<a href="#">JAbstract3DView</a>

## Class JAbstractDiagram<M->M>

diagram	
hierarchy	
owner	<a href="#">diagrams</a>
template	<a href="#">JAbstractDiagram</a>
templ.par.su bstitution	formal actual <b>M M</b>
specific	<a href="#">JTableDiagram</a>

## Class JAbstractDiagram<M->M>

diagram	
hierarchy	
owner	<a href="#">diagrams</a>
template	<a href="#">JAbstractDiagram</a>
templ.par.su bstitution	formal actual <b>M M</b>
specific	<a href="#">JTextDiagram</a>

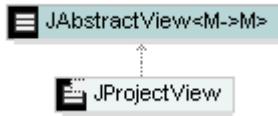
## Class JAbstractView<M->M>

diagram	
hierarchy	
owner	<a href="#">diagrams</a>
template	<a href="#">JAbstractView</a>
templ.par.su bstitution	formal actual <b>M M</b>
specific	<a href="#">JAbstractDiagram</a>

## Class JAbstractView<M->M>

diagram	
hierarchy	 <pre>graph TD; JAbstractView["JAbstractView&lt;M-&gt;M&gt;"] --&gt; JProjectManagement["JProjectManagement"]</pre>
owner	<a href="#">projectmanagement</a>
template	<a href="#">JAbstractView</a>
templ.par.su bstitution	formal actual <b>M M</b>
specific	<a href="#">JProjectManagement</a>

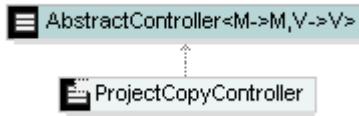
## Class JAbstractView<M->M>

diagram	
hierarchy	 <pre>graph TD; JAbstractView["JAbstractView&lt;M-&gt;M&gt;"] --&gt; JProjectView["JProjectView"]</pre>
owner	<a href="#">projectview</a>
template	<a href="#">JAbstractView</a>
templ.par.su bstitution	formal actual <b>M M</b>
specific	<a href="#">JProjectView</a>

## Class AbstractReportCompilation<M->?>

diagram	
owner	<a href="#">reportmanagement</a>
template	<a href="#">AbstractReportCompilation</a>
templ.par.substitution	formal actual M ?
typedElements	Class <a href="#">JAbstractReportUtil</a> Operation <a href="#">setReportType</a> Class <a href="#">JReportConfig</a> Operation <a href="#">setReportType</a> Class <a href="#">JReportWizard</a> Operation <a href="#">setReportType</a> Enumeration <a href="#">ReportHelper</a> Operation <a href="#">createReportCompilation</a>

## Class AbstractController<M->M,V->V>

diagram	
hierarchy	 <pre>graph TD; AbstractController["AbstractController&lt;M-&gt;M,V-&gt;V&gt;"] --&gt; ProjectCopyController["ProjectCopyController"]</pre>
owner	<a href="#">projectmanagement</a>
template	<a href="#">AbstractController</a>
templ.par.su bstitution	formal actual <b>M MV V</b>
specific	<a href="#">ProjectCopyController</a>

## Class AbstractReportCompilation<M->AbstractReportModel>

diagram	
owner	<a href="#">reportmanagement</a>
template	<a href="#">AbstractReportCompilation</a>
templ.par.substitution	formal actual M <a href="#">AbstractReportModel</a>
typedElements	Class <a href="#">JReportConfig</a> Property <a href="#">reportCompilation</a> Enumeration <a href="#">ReportHelper</a> Operation <a href="#">createReportCompilation</a>

**Class AbstractReportCompilation<M->BoxplotModel>**

diagram	
owner	<a href="#">reportmanagement</a>
template	<a href="#">AbstractReportCompilation</a>
templ.par.substitution	formal actual <b>M BoxplotModel</b>
typedElements	Enumeration <a href="#">ReportHelper</a> Operation <a href="#">createReportCompilation</a>

## Class AbstractReportCompilation<M->Cluster3DModel>

diagram	
owner	<a href="#">reportmanagement</a>
template	<a href="#">AbstractReportCompilation</a>
templ.par.su bstitution	formal actual M <a href="#">Cluster3DModel</a>
typedEleme nts	Enumeration <a href="#">ReportHelper</a> Operation <a href="#">createReportCompilation</a>

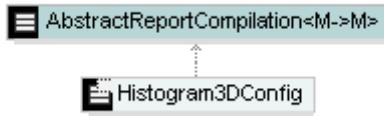
## Class AbstractReportCompilation<M->Histogram2DModel>

diagram	
owner	<a href="#">reportmanagement</a>
template	<a href="#">AbstractReportCompilation</a>
templ.par.substitution	formal actual M <a href="#">Histogram2DModel</a>
typedElements	Enumeration <a href="#">ReportHelper</a> Operation <a href="#">createReportCompilation</a>

## Class AbstractReportCompilation<M->Histogram3DModel>

diagram	
owner	<a href="#">reportmanagement</a>
template	<a href="#">AbstractReportCompilation</a>
templ.par.substitution	formal actual M <a href="#">Histogram3DModel</a>
typedElements	Enumeration <a href="#">ReportHelper</a> Operation <a href="#">createReportCompilation</a>

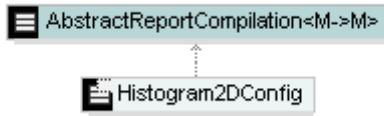
## Class AbstractReportCompilation<M->M>

diagram	
hierarchy	
owner	<a href="#">reportmanagement</a>
template	<a href="#">AbstractReportCompilation</a>
templ.par.su bstitution	formal actual <b>M M</b>
specific	<a href="#">Histogram3DConfig</a>

## Class AbstractReportCompilation<M->M>

diagram	
hierarchy	
owner	<a href="#">reportmanagement</a>
template	<a href="#">AbstractReportCompilation</a>
templ.par.su bstitution	formal actual <b>M M</b>
specific	<a href="#">Cluster3DConfig</a>

## Class AbstractReportCompilation<M->M>

diagram	
hierarchy	 <pre>graph TD; A[AbstractReportCompilation&lt;M-&gt;M&gt;] --&gt; B[Histogram2DConfig]</pre>
owner	<a href="#">reportmanagement</a>
template	<a href="#">AbstractReportCompilation</a>
templ.par.su bstitution	formal actual <b>M M</b>
specific	<a href="#">Histogram2DConfig</a>

## Class AbstractReportCompilation<M->M>

diagram	
owner	<a href="#">reportmanagement</a>
template	<a href="#">AbstractReportCompilation</a>
templ.par.su bstitution	formal actual <b>M M</b>

## Class AbstractReportCompilation<M->M>

diagram	
owner	<a href="#">reportmanagement</a>
template	<a href="#">AbstractReportCompilation</a>
templ.par.su bstitution	formal actual <b>M M</b>

## Enumeration Perspectives

diagram	 <p>The diagram shows an enumeration named "Perspectives" with four members: XYPLANE, XZPLANE, YZPLANE, and PERSPECTIVE.</p>
owner	<a href="#">diagrams</a>
ownedMember	<b>PERSPECTIVE XYPLANE XZPLANE YZPLANE</b>
target of relation	ComponentRealization <a href="#">Implementierung</a>
typedElements	Class <a href="#">JAbstract3DView</a> Operation <a href="#">setCameraPerspective</a>
documentation	This enumeration lists all predefined camera perspectives. @author David Kaufman

## EnumerationLiteral Perspectives::PERSPECTIVE

owner	<a href="#">Perspectives</a>
-------	------------------------------

## EnumerationLiteral Perspectives::XYPLANE

owner	<a href="#">Perspectives</a>
-------	------------------------------

## EnumerationLiteral Perspectives::XZPLANE

owner	<a href="#">Perspectives</a>
-------	------------------------------

## EnumerationLiteral Perspectives::YZPLANE

owner	<a href="#">Perspectives</a>
-------	------------------------------

## Class Picture

diagram	<pre> <b>Picture</b>  <b>Attributes:</b>     pictureName:String     path:String     allExifParameter:Object[*][*]=new Object[10][2]     isActive:boolean  <b>Operations:</b>     &lt;&lt;constructor&gt;&gt; Picture(in fileName:String, in path:String, in isActive:boolean)     &lt;&lt;constructor&gt;&gt; Picture()     getItems():List&lt;E-&gt;?&gt;     hasNext():boolean     next():Picture     remove():void     getBigThumbnail():Image     getSmallThumbnail():Image     getPath():String     getName():String     setName(in name:String):void     getExifParameter(in exifParameter:ExifParameter):Object     hasExifKeyword(in keyword:String):boolean     hasMinOneKeywordOf(in keywords:String[*]):boolean     hasAllKeywords(in keywords:String[*]):boolean     isActive():boolean     setActive(in isActive:boolean):void     getAllExifParameter():Object[*][*]     setAllExifParameter(in allExifParameter:Object[*][*]):void </pre>
hierarchy	<pre> classDiagram     class PictureContainer {         &lt;&lt;Iterator&lt;E-&gt;&gt;&gt;     }     class Picture     PictureContainer &lt; -- Picture </pre>
owner	<a href="#">picturemanagement</a>
ownedMember	<a href="#">allExifParameter</a> <a href="#">getAllExifParameter</a> <a href="#">getBigThumbnail</a> <a href="#">getExifParameter</a> <a href="#">getItems</a> <a href="#">getName</a> <a href="#">getPath</a> <a href="#">getSmallThumbnail</a> <a href="#">hasAllKeywords</a> <a href="#">hasExifKeyword</a> <a href="#">hasMinOneKeywordOf</a> <a href="#">hasNext</a> <a href="#">isActive</a> <a href="#">isActive</a> <a href="#">next</a> <a href="#">path</a> <a href="#">Picture</a> <a href="#">Picture</a> <a href="#">pictureName</a> <a href="#">remove</a> <a href="#">setActive</a> <a href="#">setAllExifParameter</a> <a href="#">setName</a>
implemented interfaces	<a href="#">PictureContainer</a>
source of relation	InterfaceRealization <a href="#">PictureContainer</a>
target of relation	ComponentRealization <a href="#">Implementierung</a>
typedElements	Class <a href="#">Directory</a> Operation <a href="#">next</a> Class <a href="#">Frequency3DPoint</a> Property <a href="#">pictures</a> Operation <a href="#">addPicture</a> <a href="#">getPictures</a> <a href="#">setPictures</a> Class <a href="#">JAbstract3DView</a> Operation <a href="#">setCurrentDescription</a> <a href="#">setCurrentPicture</a> Class <a href="#">Picture</a> Operation <a href="#">next</a> Class <a href="#">PictureParameter</a> Property <a href="#">picture</a> Operation <a href="#">getPicture</a> <a href="#">PictureParameter</a> <a href="#">setPicture</a> Class <a href="#">PictureSet</a> Operation <a href="#">next</a> Class <a href="#">ProjectModel</a> Operation <a href="#">getAllPictures</a> <a href="#">getPicturesOfADirectory</a> <a href="#">getPicturesOfAPictureSet</a>

**Property Picture::allExifParameter**

owner	<a href="#">Picture</a>
-------	-------------------------

**Operation Picture::getAllExifParameter**

owner	<a href="#">Picture</a>
parameter	name direction type multiplicity default <b>return return Object</b> *,*
ownedMember	<b>return</b>

**Operation Picture::getBigThumbnail**

owner	<a href="#">Picture</a>
parameter	name direction type multiplicity default <b>return return Image</b>
ownedMember	<b>return</b>

**Operation Picture::getExifParameter**

owner	<a href="#">Picture</a>
parameter	name direction type multiplicity default <b>exifParameter</b> in <a href="#">ExifParameter</a> <b>return return Object</b>
ownedMember	<b>exifParameter return</b>

**Operation Picture::getItems**

owner	<a href="#">Picture</a>
parameter	name direction type multiplicity default <b>return return List&lt;E-&gt;?&gt;</b>
ownedMember	<b>return</b>

**Operation Picture::getName**

owner	<a href="#">Picture</a>
parameter	name direction type multiplicity default <b>return return String</b>
ownedMember	<b>return</b>

**Operation Picture::getPath**

owner	<a href="#">Picture</a>
parameter	name direction type multiplicity default <b>return return String</b>
ownedMember	<b>return</b>

**Operation Picture::getSmallThumbnail**

owner	<a href="#">Picture</a>
parameter	name direction type multiplicity default <b>return return Image</b>

ownedMember	<b>return</b>
-------------	---------------

#### Operation Picture::hasAllKeywords

owner	<a href="#">Picture</a>
parameter	name direction type multiplicity default <b>keywords in String * return return boolean</b>
ownedMember	<b>keywords return</b>

#### Operation Picture::hasExifKeyword

owner	<a href="#">Picture</a>
parameter	name direction type multiplicity default <b>keyword in String return return boolean</b>
ownedMember	<b>keyword return</b>

#### Operation Picture::hasMinOneKeywordOf

owner	<a href="#">Picture</a>
parameter	name direction type multiplicity default <b>keywords in String * return return boolean</b>
ownedMember	<b>keywords return</b>

#### Operation Picture::hasNext

owner	<a href="#">Picture</a>
parameter	name direction type multiplicity default <b>return return boolean</b>
ownedMember	<b>return</b>

#### Operation Picture::isActive

owner	<a href="#">Picture</a>
parameter	name direction type multiplicity default <b>return return boolean</b>
ownedMember	<b>return</b>

#### Property Picture::isActive

owner	<a href="#">Picture</a>
-------	-------------------------

#### Operation Picture::next

owner	<a href="#">Picture</a>
parameter	name direction type multiplicity default <b>return return Picture</b>
ownedMember	<b>return</b>

### Property Picture::path

owner	<a href="#">Picture</a>
-------	-------------------------

### Operation Picture::Picture

owner	<a href="#">Picture</a>
parameter	name direction type multiplicity default <b>fileName in String path in String isActive in boolean</b>
ownedMember	<b>fileName isActive path</b>

### Operation Picture::Picture

owner	<a href="#">Picture</a>
-------	-------------------------

### Property Picture::pictureName

owner	<a href="#">Picture</a>
-------	-------------------------

### Operation Picture::remove

owner	<a href="#">Picture</a>
parameter	name direction type multiplicity default <b>return return void</b>
ownedMember	<b>return</b>

### Operation Picture::setActive

owner	<a href="#">Picture</a>
parameter	name direction type multiplicity default <b>isActive in boolean return return void</b>
ownedMember	<b>isActive return</b>

### Operation Picture::setAllExifParameter

owner	<a href="#">Picture</a>
parameter	name direction type multiplicity default <b>allExifParameter in Object *,* return return void</b>
ownedMember	<b>allExifParameter return</b>

### Operation Picture::setName

owner	<a href="#">Picture</a>
parameter	name direction type multiplicity default <b>name in String return return void</b>
ownedMember	<b>name return</b>

## Class Picture

diagram	<pre> classDiagram     class Picture {         getItems():ArrayList&lt;E-&gt;?&gt;         hasNext():boolean         next():Object         remove():void     }   </pre>
hierarchy	<pre> classDiagram     class PictureContainer {         &lt;&lt;Picture&gt;&gt;     }     class Picture {         &lt;&lt;Iterator&gt;&gt;     }     PictureContainer &lt; -- Picture   </pre>
owner	<a href="#">Kompositum</a>
ownedMember	<a href="#">getItems</a> <a href="#">hasNext</a> <a href="#">next</a> <a href="#">remove</a>
implemented interfaces	<a href="#">PictureContainer</a>
source of relation	InterfaceRealization <a href="#">PictureContainer</a>
target of relation	ComponentRealization <a href="#">Implementierung</a>

## Operation Picture::getItems

owner	<a href="#">Picture</a>
parameter	name direction type multiplicity default <b>return return ArrayList&lt;E-&gt;?&gt;</b>
ownedMember	<b>return</b>

## Operation Picture::hasNext

owner	<a href="#">Picture</a>
parameter	name direction type multiplicity default <b>return return boolean</b>
ownedMember	<b>return</b>

## Operation Picture::next

owner	<a href="#">Picture</a>
parameter	name direction type multiplicity default <b>return return Object</b>
ownedMember	<b>return</b>

## Operation Picture::remove

owner	<a href="#">Picture</a>
parameter	name direction type multiplicity default <b>return return void</b>

ownedMember	<b>return</b>
-------------	---------------

## Interface PictureContainer

diagram	<pre> &lt;&lt;interface&gt;&gt; PictureContainer  getItems():List&lt;E-&gt;? getName():String setName(in name:String):void </pre>
hierarchy	<pre> classDiagram     PictureContainer &lt; -- Iterator&lt;E&gt;Picture     PictureContainer &lt; -- Directory     PictureContainer &lt; -- Picture     PictureContainer &lt; -- PictureSet </pre>
owner	<a href="#">picturemanagement</a>
ownedMember	<a href="#">getItems</a> <a href="#">getName</a> <a href="#">setName</a>
general	<a href="#">Iterator&lt;E&gt;Picture</a>
target of relation	InterfaceRealization <a href="#">PictureSet</a> <a href="#">Picture</a> <a href="#">Directory</a> ComponentRealization <a href="#">Implementierung</a>
typedElements	Class <a href="#">Bar</a> Property <a href="#">pictureContainer</a> Operation <a href="#">Bar getPictureContainer setPictureContainer</a> Class <a href="#">Boxplot</a> Operation <a href="#">Boxplot</a> Class <a href="#">PictureSet</a> Operation <a href="#">add remove</a> Class <a href="#">ProjectModel</a> Operation <a href="#">addContentToPictureSet removeContentFromPictureSet</a> Class <a href="#">TableModel</a> Operation <a href="#">getPictures</a>

## Operation PictureContainer::getItems

owner	<a href="#">PictureContainer</a>
parameter	name direction type multiplicity default <b>return return List&lt;E-&gt;?&gt;</b>
ownedMember	<b>return</b>

## Operation PictureContainer::getName

owner	<a href="#">PictureContainer</a>
parameter	name direction type multiplicity default <b>return return String</b>
ownedMember	<b>return</b>

## Operation PictureContainer::setName

owner	<a href="#">PictureContainer</a>
parameter	name direction type multiplicity default <b>name in String return return void</b>
ownedMember	<b>name return</b>



## Interface PictureContainer

diagram	<pre> classDiagram     class PictureContainer {         &lt;&lt;interface&gt;&gt;         getItems():ArrayList&lt;E-&gt;?&gt;     }   </pre>
hierarchy	<pre> classDiagram     class Iterator     class PictureContainer     class Directory     class Picture     class PictureSet     Iterator &lt; -- PictureContainer     PictureContainer &lt; -- Directory     PictureContainer &lt; -- Picture     PictureContainer &lt; -- PictureSet   </pre>
owner	<a href="#">Kompositum</a>
ownedMember	<a href="#">getItems</a>
general	<a href="#">Iterator</a>
target of relation	InterfaceRealization <a href="#">Directory</a> <a href="#">Picture</a> <a href="#">PictureSet</a> ComponentRealization <a href="#">Implementierung</a>
typedElements	Class <a href="#">PictureSet</a> Operation <a href="#">add</a> <a href="#">remove</a>

## Operation PictureContainer::getItems

owner	<a href="#">PictureContainer</a>
parameter	name direction type multiplicity default <b>return</b> <b>return</b> <a href="#">ArrayList&lt;E-&gt;?&gt;</a>
ownedMember	<b>return</b>

## Class PictureListCheckBoxController

diagram	<pre> classDiagram     class PictureListCheckBoxController {         &lt;&lt;constructor&gt;&gt; PictureListCheckBoxController(in model:M, in view:V)         &lt;&lt;annotations&gt;&gt; actionPerformed(e:ActionEvent):void     }     class AbstractController {         &lt;&lt;constructor&gt;&gt; AbstractController(in M, in V)     }     PictureListCheckBoxController &lt; -- AbstractController     PictureListCheckBoxController "1" --&gt; "1" PictureListCheckBoxController   </pre>
hierarchy	<pre> classDiagram     class AbstractController {         &lt;&lt;constructor&gt;&gt; AbstractController(in M, in V)     }     class PictureListCheckBoxController {         &lt;&lt;constructor&gt;&gt; PictureListCheckBoxController(in model:M, in view:V)         &lt;&lt;annotations&gt;&gt; actionPerformed(e:ActionEvent):void     }     PictureListCheckBoxController &lt; -- AbstractController   </pre>
owner	<a href="#">projectview</a>
template parameters	name kind constrainingClassifier default M Class <a href="#">ProjectModel</a> V Class <a href="#">JProjectView&lt;M&gt;</a>
ownedMember	<a href="#">actionPerformed</a> <a href="#">PictureListCheckBoxController</a>
general	<a href="#">AbstractController&lt;M&gt;,&lt;V&gt;</a>
target of relation	ComponentRealization <a href="#">Implementierung</a>
documentation	Represents the Actions which are done by clicking on the checkbox of one of the pictures in the picturelist. Acts in harmony with a JProjectView.

## Operation PictureListCheckBoxController::actionPerformed

owner	<a href="#">PictureListCheckBoxController</a>
parameter	name direction type multiplicity default e in ActionEvent return return void
ownedMember	e return

## Operation PictureListCheckBoxController::PictureListCheckBoxController

owner	<a href="#">PictureListCheckBoxController</a>
parameter	name direction type multiplicity default model in M view in V
ownedMember	model view

## Class PictureListClickOnController

diagram	<pre> classDiagram     class PictureListClickOnController {         &lt;&lt;constructor&gt;&gt; PictureListClickOnController(in model:M, in view:V)         &lt;&lt;annotations&gt;&gt; actionPerformed(in e:ActionEvent):void     }     class AbstractController {         &lt;&lt;constructor&gt;&gt; AbstractController&lt;M&gt; M, V&gt; V     }     PictureListClickOnController &lt; -- AbstractController     </pre>
hierarchy	<pre> classDiagram     class AbstractController {         &lt;&lt;constructor&gt;&gt; AbstractController&lt;M&gt; M, V&gt; V     }     class PictureListClickOnController {         &lt;&lt;constructor&gt;&gt; PictureListClickOnController(in model:M, in view:V)         &lt;&lt;annotations&gt;&gt; actionPerformed(in e:ActionEvent):void     }     PictureListClickOnController &lt; -- AbstractController     </pre>
owner	<b>projectview</b>
template parameters	name kind constrainingClassifier default <b>M Class ProjectModel</b> <b>V Class JProjectView&lt;M-&gt;M&gt;</b>
ownedMember	<a href="#">actionPerformed PictureListClickOnController</a>
general	<a href="#">AbstractController&lt;M-&gt;M,V-&gt;V&gt;</a>
target of relation	ComponentRealization <a href="#">Implementierung</a>
documentation	Represents the Actions which are done by clicking on the picturelist. Acts in harmony with a JProjectview.

## Operation PictureListClickOnController::actionPerformed

owner	<a href="#">PictureListClickOnController</a>
parameter	name direction type multiplicity default <b>e in ActionEvent return return void</b>
ownedMember	<b>e return</b>

## Operation PictureListClickOnController::PictureListClickOnController

owner	<a href="#">PictureListClickOnController</a>
parameter	name direction type multiplicity default <b>model in M view in V</b>
ownedMember	<b>model view</b>

## Class PictureParameter

diagram	<pre> classDiagram     class PictureParameter {         exifParameter : ExifParameter         picture : Picture     }     &lt;&lt;constructor&gt;&gt; PictureParameter(in exifParameter: ExifParameter, in picture: Picture)     getExifParameter() : ExifParameter     setExifParameter(in exifParameter: ExifParameter) : void     getPicture() : Picture     setPicture(in picture: Picture) : void   </pre>
owner	<a href="#">reportmanagement</a>
ownedMember	<a href="#">exifParameter</a> <a href="#">getExifParameter</a> <a href="#">getPicture</a> <a href="#">picture</a> <a href="#">PictureParameter</a> <a href="#">setExifParameter</a> <a href="#">setPicture</a>
target of relation	ComponentRealization <a href="#">Implementierung</a>
typedElements	Class <a href="#">AbstractReportModel</a> Property <a href="#">missingExifParameters</a> Operation <a href="#">getPicturesWithMissingExifParameter</a>
documentation	Datacontainer fÃ¼r a pair of exifParameter and picture. @author Kevin Zuber

## Property PictureParameter::exifParameter

owner	<a href="#">PictureParameter</a>
-------	----------------------------------

## Operation PictureParameter::getExifParameter

owner	<a href="#">PictureParameter</a>
parameter	name direction type multiplicity default <b>return return ExifParameter</b>
ownedMember	<b>return</b>

## Operation PictureParameter::getPicture

owner	<a href="#">PictureParameter</a>
parameter	name direction type multiplicity default <b>return return Picture</b>
ownedMember	<b>return</b>

## Property PictureParameter::picture

owner	<a href="#">PictureParameter</a>
-------	----------------------------------

## Operation PictureParameter::PictureParameter

owner	<a href="#">PictureParameter</a>
parameter	name direction type multiplicity default <b>exifParameter in ExifParameter picture in Picture</b>
ownedMember	<b>exifParameter picture</b>

	ber	
--	-----	--

#### Operation **PictureParameter::setExifParameter**

owner	<a href="#">PictureParameter</a>
parameter	name direction type multiplicity default <b>exifParameter</b> in <a href="#">ExifParameter</a> return return void
ownedMember	<b>exifParameter return</b>

#### Operation **PictureParameter::setPicture**

owner	<a href="#">PictureParameter</a>
parameter	name direction type multiplicity default <b>picture</b> in <a href="#">Picture</a> return return void
ownedMember	<b>picture return</b>

## Class PictureSet

diagram	<pre> <b>PictureSet</b>  <b>childs:List&lt;E-&gt;PictureContainer&gt;=new ArrayList&lt;PictureContainer&gt;()</b> <b>name:String</b> <b>iD:int</b>  <b>&lt;&lt;constructor&gt;&gt; PictureSet(<b>in</b> pictureSetName:<b>String</b>, <b>in</b> freePictureSetID:<b>int</b>)</b> <b>&lt;&lt;constructor&gt;&gt; PictureSet(<b>in</b> pictureSetToCopy:<b>PictureSet</b>, <b>in</b> pictureSetName:<b>String</b>, <b>in</b> freePictureSetID:<b>int</b>)</b> <b>add(<b>in</b> container:<b>PictureContainer</b>):void</b> <b>remove(<b>in</b> container:<b>PictureContainer</b>):void</b> <b>getItems():List&lt;E-&gt;PictureContainer&gt;</b> <b>hasNext():boolean</b> <b>next():Picture</b> <b>remove():void</b> <b>getchilds():List&lt;E-&gt;PictureContainer&gt;</b> <b>setchilds(<b>in</b> childs:List&lt;E-&gt;PictureContainer&gt;):void</b> <b>getid():int</b> <b>setid(<b>in</b> id:int):void</b> <b>getname():String</b> <b>setname(<b>in</b> name:String):void</b> </pre>
hierarchy	<pre> classDiagram     class PictureSet {         &lt;&lt;PictureContainer&gt;&gt;     }     class PictureContainer {         &lt;&lt;Iterator&lt;E-&gt;Picture&gt;&gt;     } </pre>
owner	<a href="#">picturemanagement</a>
ownedMember	<a href="#">add</a> <a href="#">child</a> <a href="#">getChilds</a> <a href="#">getId</a> <a href="#">getItems</a> <a href="#">hasNext</a> <a href="#">iD</a> <a href="#">name</a> <a href="#">next</a> <a href="#">PictureSet</a> <a href="#">PictureSet</a> <a href="#">remove</a> <a href="#">remove</a> <a href="#">setChilds</a> <a href="#">setId</a> <a href="#">setName</a>
implemented interfaces	<a href="#">PictureContainer</a>
source of relation	InterfaceRealization <a href="#">PictureContainer</a>
target of relation	ComponentRealization <a href="#">Implementierung</a>
typedElements	Class <a href="#">JPictureSetExif</a> Property <a href="#">pictureContainer</a> Operation <a href="#">getPictureContainer</a> <a href="#">setPictureContainer</a> Class <a href="#">PictureSet</a> Operation <a href="#">PictureSet</a> Class <a href="#">ProjectModel</a> Operation <a href="#">addContentToPictureSet</a> <a href="#">addPictureSet</a> <a href="#">getDirectoriesOfAPictureSet</a> <a href="#">getPictureSetsOfAPictureSet</a> <a href="#">getPicturesOfAPictureSet</a> <a href="#">removeContentFromPictureSet</a> <a href="#">removePictureSet</a>

## Operation PictureSet::add

owner	<a href="#">PictureSet</a>
parameter	name direction type multiplicity default <b>container</b> in <a href="#">PictureContainer</a> return return void
ownedMember	<b>container</b> return

**Property PictureSet::childs**

owner	<a href="#">PictureSet</a>
-------	----------------------------

**Operation PictureSet::getChilds**

owner	<a href="#">PictureSet</a>
parameter	name direction type multiplicity default <b>return return List&lt;E-&gt;PictureContainer&gt;</b>
ownedMember	<b>return</b>

**Operation PictureSet::getId**

owner	<a href="#">PictureSet</a>
parameter	name direction type multiplicity default <b>return return int</b>
ownedMember	<b>return</b>

**Operation PictureSet::getItems**

owner	<a href="#">PictureSet</a>
parameter	name direction type multiplicity default <b>return return List&lt;E-&gt;PictureContainer&gt;</b>
ownedMember	<b>return</b>

**Operation PictureSet::getName**

owner	<a href="#">PictureSet</a>
parameter	name direction type multiplicity default <b>return return String</b>
ownedMember	<b>return</b>

**Operation PictureSet::hasNext**

owner	<a href="#">PictureSet</a>
parameter	name direction type multiplicity default <b>return return boolean</b>
ownedMember	<b>return</b>

**Property PictureSet::iD**

owner	<a href="#">PictureSet</a>
-------	----------------------------

**Property PictureSet::name**

owner	<a href="#">PictureSet</a>
-------	----------------------------

**Operation PictureSet::next**

owner	<a href="#">PictureSet</a>
parameter	name direction type multiplicity default <b>return return Picture</b>

ownedMember	<b>return</b>
-------------	---------------

#### Operation PictureSet::PictureSet

owner	<b>PictureSet</b>
parameter	name direction type multiplicity default <b>pictureSetName</b> in String <b>freePictureSetID</b> in int
ownedMember	<b>freePictureSetID</b> <b>pictureSetName</b>

#### Operation PictureSet::PictureSet

owner	<b>PictureSet</b>
parameter	name direction type multiplicity default <b>pictureSetToCopy</b> in <a href="#">PictureSet</a> <b>pictureSetName</b> in String <b>freePictureSetID</b> in int
ownedMember	<b>freePictureSetID</b> <b>pictureSetName</b> <b>pictureSetToCopy</b>

#### Operation PictureSet::remove

owner	<b>PictureSet</b>
parameter	name direction type multiplicity default <b>container</b> in <a href="#">PictureContainer</a> <b>return</b> return void
ownedMember	<b>container</b> <b>return</b>

#### Operation PictureSet::remove

owner	<b>PictureSet</b>
parameter	name direction type multiplicity default <b>return</b> <b>return</b> void
ownedMember	<b>return</b>

#### Operation PictureSet::setChilds

owner	<b>PictureSet</b>
parameter	name direction type multiplicity default <b>childs</b> in <a href="#">List&lt;E-&gt;PictureContainer&gt;</a> <b>return</b> return void
ownedMember	<b>childs</b> <b>return</b>

#### Operation PictureSet::setId

owner	<b>PictureSet</b>
parameter	name direction type multiplicity default <b>id</b> in int <b>return</b> return void
ownedMember	<b>id</b> <b>return</b>

#### Operation PictureSet::setName

owner	<b>PictureSet</b>
parameter	name direction type multiplicity default <b>name</b> in String <b>return</b> return void
ownedMember	<b>name</b> <b>return</b>



## Class PictureSet

diagram	<pre> <b>PictureSet</b>   child:List&lt;E-&gt;PictureContainer&gt;=new ArrayList&lt;PictureContainer&gt;()    □ add(in container:PictureContainer):void   □ remove(in container:PictureContainer):void   □ getItems():ArrayList&lt;E-&gt;?&gt;   □ hasNext():boolean   □ next():Object   □ remove():void </pre>
hierarchy	<pre> classDiagram     PictureSet &lt; -- PictureContainer     PictureContainer &lt; -- Iterator </pre>
owner	<a href="#">Kompositum</a>
ownedMember	<a href="#">add childs getItems hasNext next remove remove</a>
implemented interfaces	<a href="#">PictureContainer</a>
source of relation	InterfaceRealization <a href="#">PictureContainer</a>
target of relation	ComponentRealization <a href="#">Implementierung</a>

### Operation PictureSet::add

owner	<a href="#">PictureSet</a>
parameter	name direction type multiplicity default <b>container</b> in <a href="#">PictureContainer</a> return return void
ownedMember	<b>container</b> return

### Property PictureSet::childs

owner	<a href="#">PictureSet</a>
-------	----------------------------

### Operation PictureSet::getItems

owner	<a href="#">PictureSet</a>
parameter	name direction type multiplicity default <b>return</b> return <a href="#">ArrayList&lt;E-&gt;?&gt;</a>
ownedMember	<b>return</b>

### Operation PictureSet::hasNext

owner	<a href="#">PictureSet</a>
-------	----------------------------

parameter	name direction type multiplicity default <b>return return boolean</b>
ownedMember	<b>return</b>

#### Operation **PictureSet::next**

owner	<b>PictureSet</b>
parameter	name direction type multiplicity default <b>return return Object</b>
ownedMember	<b>return</b>

#### Operation **PictureSet::remove**

owner	<b>PictureSet</b>
parameter	name direction type multiplicity default <b>container in PictureContainer return return void</b>
ownedMember	<b>container return</b>

#### Operation **PictureSet::remove**

owner	<b>PictureSet</b>
parameter	name direction type multiplicity default <b>return return void</b>
ownedMember	<b>return</b>

## Class PictureSetContentListAddController

diagram	<p><code>M&gt;ProjectModel,V&gt;JProjectView&lt;M-&gt;M&gt;</code></p> <p><b>PictureSetContentListAddController</b></p> <p><code>&lt;&lt;constructor&gt;&gt; PictureSetContentListAddController(in model:M, in view:V)</code></p> <p><code>&lt;&lt;annotations&gt;&gt; actionPerformed(in e:ActionEvent):void</code></p>	<b>actionPerformed() (Operation)</b> <code>&lt;&lt;annotations&gt;&gt; modifiers = Override</code>
hierarchy	<pre> classDiagram     class AbstractController {         &lt;&lt;constructor&gt;&gt; AbstractController(in M, in V)     }     class PictureSetContentListAddController {         &lt;&lt;constructor&gt;&gt; PictureSetContentListAddController(in model:M, in view:V)         &lt;&lt;operations&gt;&gt; actionPerformed(in e:ActionEvent):void     }     AbstractController &lt; -- PictureSetContentListAddController   </pre>	
owner	<b>projectview</b>	
template parameters	name kind constrainingClassifier default <b>M Class ProjectModel</b> <b>V Class JProjectView&lt;M-&gt;M&gt;</b>	
ownedMember	<b>actionPerformed PictureSetContentListAddController</b>	
general	<b>AbstractController&lt;M-&gt;M,V-&gt;V&gt;</b>	ComponentRealization <b>Implementierung</b>
target of relation		
documentation	Represents the Actions which are done by editing the projectnamedescription. Acts in harmony with a JProjectView.	

## Operation PictureSetContentListAddController::actionPerformed

owner	<b>PictureSetContentListAddController</b>
parameter	name direction type multiplicity default <b>e in ActionEvent return return void</b>
ownedMember	<b>e return</b>

## Operation PictureSetContentListAddController::PictureSetContentListAddController

owner	<b>PictureSetContentListAddController</b>
parameter	name direction type multiplicity default <b>model in M view in V</b>
ownedMember	<b>model view</b>

## Class PictureSetContentListClickOnController

diagram	<p>M&gt;ProjectModel,V&gt;JProjectView&lt;M-&gt;M&gt;</p> <p><b>PictureSetContentListClickOnController</b></p> <p>actionPerformed() (Operation)      &lt;&lt;annotations&gt;&gt; modifiers = Override</p> <p>&lt;&lt;constructor&gt;&gt; PictureSetContentListClickOnController(in model:M, in view:V)</p> <p>&lt;&lt;annotations&gt;&gt; actionPerformed(in e:ActionEvent):void</p>
hierarchy	<pre> classDiagram     class AbstractController {         &lt;&lt;AbstractController&gt;&gt;     }     class PictureSetContentListClickOnController {         &lt;&lt;PictureSetContentListClickOnController&gt;&gt;     }     AbstractController &lt; -- PictureSetContentListClickOnController   </pre>
owner	<b>projectview</b>
template parameters	name kind constrainingClassifier default M Class <a href="#">ProjectModel</a> V Class <a href="#">JProjectView&lt;M-&gt;M&gt;</a>
ownedMember	<a href="#">actionPerformed</a> <a href="#">PictureSetContentListClickOnController</a>
general	<a href="#">AbstractController&lt;M-&gt;M,V-&gt;V&gt;</a>
target of relation	ComponentRealization <a href="#">Implementierung</a>
documentation	Represents the Actions which are done by clicking on the picturesetcontentlist. Acts in harmony with a JProjectview.

## Operation PictureSetContentListClickOnController::actionPerformed

owner	<a href="#">PictureSetContentListClickOnController</a>
parameter	name direction type multiplicity default e in ActionEvent return return void
ownedMember	e return

## Operation PictureSetContentListClickOnController::PictureSetContentListClickOnController

owner	<a href="#">PictureSetContentListClickOnController</a>
parameter	name direction type multiplicity default model in M view in V
ownedMember	model view

## Class PictureSetContentListDeleteController

diagram	<pre> classDiagram     class PictureSetContentListDeleteController {         &lt;&lt;constructor&gt;&gt; PictureSetContentListDeleteController(in model:M, in view:V)         &lt;&lt;annotations&gt;&gt; actionPerformed(in e:ActionEvent):void     }     PictureSetContentListDeleteController &lt; -- AbstractController&lt;M-&gt;M,V-&gt;V&gt;     </pre>
hierarchy	<pre> classDiagram     class PictureSetContentListDeleteController {         &lt;&lt;constructor&gt;&gt; PictureSetContentListDeleteController(in model:M, in view:V)         &lt;&lt;annotations&gt;&gt; actionPerformed(in e:ActionEvent):void     }     PictureSetContentListDeleteController &lt; -- AbstractController&lt;M-&gt;M,V-&gt;V&gt;     </pre>
owner	<a href="#">projectview</a>
template parameters	name kind constrainingClassifier default M Class <a href="#">ProjectModel</a> V Class <a href="#">JProjectView&lt;M-&gt;M&gt;</a>
ownedMember	<a href="#">actionPerformed PictureSetContentListDeleteController</a>
general	<a href="#">AbstractController&lt;M-&gt;M,V-&gt;V&gt;</a>
target of relation	ComponentRealization <a href="#">Implementierung</a>
documentation	Represents the Actions which are done by clicking on delete nearby the picturesetcontent. Acts in harmony with a JProjectView.

## Operation PictureSetContentListDeleteController::actionPerformed

owner	<a href="#">PictureSetContentListDeleteController</a>
parameter	name direction type multiplicity default e in ActionEvent return return void
ownedMember	e return

## Operation PictureSetContentListDeleteController::PictureSetContentListDeleteController

owner	<a href="#">PictureSetContentListDeleteController</a>
parameter	name direction type multiplicity default model in M view in V
ownedMember	model view

## Class PictureSetContentListDropController

diagram	<p><code>M&gt;ProjectModel, V&gt;JProjectView&lt;M-&gt;M&gt;</code></p> <p><b>PictureSetContentListDropController</b></p> <ul style="list-style-type: none"> <li><code>&lt;&lt;constructor&gt;&gt; PictureSetContentListDropController(in model:M, in view:V)</code></li> <li><code>&lt;&lt;annotations&gt;&gt; actionPerformed(in e:ActionEvent):void</code></li> <li><code>canBeInserted():boolean</code></li> </ul>	<b>actionPerformed() (Operation)</b> <code>&lt;&lt;annotations&gt;&gt; modifiers = Override</code>
hierarchy	<pre> classDiagram     class AbstractController {         &lt;&lt;AbstractController&gt;&gt; M-&gt;M, V-&gt;V     }     class PictureSetContentListDropController {         &lt;&lt;PictureSetContentListDropController&gt;&gt;     }     AbstractController &lt; -- PictureSetContentListDropController   </pre>	
owner	<a href="#">projectview</a>	
template parameters	name kind constrainingClassifier default <b>M Class ProjectModel</b> <b>V Class JProjectView&lt;M-&gt;M&gt;</b>	
ownedMember	<a href="#">actionPerformed</a> <a href="#">canBeInserted</a> <a href="#">PictureSetContentListDropController</a>	
general	<a href="#">AbstractController&lt;M-&gt;M, V-&gt;V&gt;</a>	ComponentRealization <a href="#">Implementierung</a>
target of relation		
documentation	<p>Represents the Actions which are done by dropping pictureset into the content on the ok button when you want to delete a report.</p> <p>Acts in harmony with a JProjectView.</p>	

## Operation PictureSetContentListDropController::actionPerformed

owner	<a href="#">PictureSetContentListDropController</a>
parameter	name direction type multiplicity default <b>e in ActionEvent return return void</b>
ownedMember	<b>e return</b>

## Operation PictureSetContentListDropController::canBeInserted

owner	<a href="#">PictureSetContentListDropController</a>
parameter	name direction type multiplicity default <b>return return boolean</b>
ownedMember	<b>return</b>

## Operation PictureSetContentListDropController::PictureSetContentListDropController

owner	<a href="#">PictureSetContentListDropController</a>
parameter	name direction type multiplicity default <b>model in M view in V</b>
ownedMember	<b>model view</b>



## Class PictureSetContentListRefreshController

diagram	<pre> classDiagram     class PictureSetContentListRefreshController {         &lt;&lt;constructor&gt;&gt; PictureSetContentListRefreshController(in model:M, in view:V)         &lt;&lt;annotations&gt;&gt; actionPerformed(in e:ActionEvent) void     }     class AbstractController&lt;M&gt;{M}     class JProjectView&lt;M&gt;{M}     class PictureSetContentListRefreshController &lt; -- AbstractController&lt;M&gt;     M&gt;ProjectModel &lt; -- JProjectView&lt;M&gt;     JProjectView&lt;M&gt; --&gt; PictureSetContentListRefreshController   </pre>
hierarchy	<pre> classDiagram     class AbstractController&lt;M&gt;{M}     class PictureSetContentListRefreshController &lt; -- AbstractController&lt;M&gt;   </pre>
owner	<a href="#">projectview</a>
template parameters	name kind constrainingClassifier default <b>M Class ProjectModel V Class JProjectView&lt;M-&gt;M&gt;</b>
ownedMember	<a href="#">actionPerformed PictureSetContentListRefreshController</a>
general	<a href="#">AbstractController&lt;M&gt;{M,V&gt;V}</a>
target of relation	ComponentRealization <a href="#">Implementierung</a>
documentation	Represents the Actions which are done by clicking on refresh. Acts in harmony with JProjectView.

## Operation PictureSetContentListRefreshController::actionPerformed

owner	<a href="#">PictureSetContentListRefreshController</a>
parameter	name direction type multiplicity default <b>e in ActionEvent return return void</b>
ownedMember	<b>e return</b>

## Operation PictureSetContentListRefreshController::PictureSetContentListRefreshController

owner	<a href="#">PictureSetContentListRefreshController</a>
parameter	name direction type multiplicity default <b>model in M view in V</b>
ownedMember	<b>model view</b>

## Class PictureSetListClickOnController

diagram	<pre> classDiagram     class PictureSetListClickOnController {         &lt;&lt;constructor&gt;&gt; PictureSetListClickOnController(in model:M, in view:V)         &lt;&lt;annotations&gt;&gt; actionPerformed(in e:ActionEvent):void     }     class AbstractController&lt;M-&gt;M, V-&gt;V&gt;     PictureSetListClickOnController "2" -- "1" AbstractController   </pre>	<b>actionPerformed() (Operation)</b> <i>&lt;&lt;annotations&gt;&gt;</i> modifiers = Override
hierarchy	<pre> classDiagram     class AbstractController&lt;M-&gt;M, V-&gt;V&gt;     class PictureSetListClickOnController     PictureSetListClickOnController &lt; -- AbstractController   </pre>	
owner	<a href="#">projectview</a>	
template parameters	name kind constrainingClassifier default <b>M Class ProjectModel V Class JProjectView&lt;M-&gt;M&gt;</b>	
ownedMember	<a href="#">actionPerformed PictureSetListClickOnController</a>	
general	<a href="#">AbstractController&lt;M-&gt;M,V-&gt;V&gt;</a>	ComponentRealization <a href="#">Implementierung</a>
target of relation		
documentation	Represents the Actions which are done by clicking on the picturesetlist. Acts in harmony with a JProjectview.	

## Operation PictureSetListClickOnController::actionPerformed

owner	<a href="#">PictureSetListClickOnController</a>
parameter	name direction type multiplicity default <b>e in ActionEvent return return void</b>
ownedMember	<b>e return</b>

## Operation PictureSetListClickOnController::PictureSetListClickOnController

owner	<a href="#">PictureSetListClickOnController</a>
parameter	name direction type multiplicity default <b>model in M view in V</b>
ownedMember	<b>model view</b>

## Class PictureSetListCopyController

diagram	<pre> classDiagram     PictureSetListCopyController &lt; -- AbstractController&lt;M-&gt;M,V-&gt;V&gt;     PictureSetListCopyController &lt; --&gt; actionPerformed : (Operation)     PictureSetListCopyController &lt; --&gt; constructor : (Constructor)     </pre> <p><code>&lt;&lt;constructor&gt;&gt; PictureSetListCopyController(in model:M, in view:V)</code></p> <p><code>&lt;&lt;annotations&gt;&gt; actionPerformed(in e:ActionEvent):void</code></p>
hierarchy	<pre> classDiagram     class AbstractController&lt;M-&gt;M,V-&gt;V&gt;     class PictureSetListCopyController     AbstractController &lt; -- PictureSetListCopyController     </pre>
owner	<a href="#">projectview</a>
template parameters	name kind constrainingClassifier default M Class <a href="#">ProjectModel</a> V Class <a href="#">JProjectView&lt;M-&gt;M&gt;</a>
ownedMember	<a href="#">actionPerformed PictureSetListCopyController</a>
general	<a href="#">AbstractController&lt;M-&gt;M,V-&gt;V&gt;</a>
target of relation	ComponentRealization <a href="#">Implementierung</a>
documentation	<p>Represents the actions which are done by clicking on copy picture set.</p> <p>Acts in harmony with a JProjectView.</p>

## Operation PictureSetListCopyController::actionPerformed

owner	<a href="#">PictureSetListCopyController</a>
parameter	name direction type multiplicity default e in ActionEvent return return void
ownedMember	e return

## Operation PictureSetListCopyController::PictureSetListCopyController

owner	<a href="#">PictureSetListCopyController</a>
parameter	name direction type multiplicity default model in M view in V
ownedMember	model view

## Class PictureSetListCreateController

diagram	<p>M&gt;ProjectModel,V&gt;JProjectView&lt;M-&gt;M&gt;</p> <p><b>PictureSetListCreateController</b></p> <p>&lt;&lt;constructor&gt;&gt; PictureSetListCreateController(in model:M, in view:V)</p> <p>&lt;&lt;annotations&gt;&gt; actionPerformed(e:ActionEvent):void</p>	<b>actionPerformed() (Operation)</b> <<annotations>> modifiers = Override
hierarchy	<pre> classDiagram     class AbstractController {         &lt;&lt;AbstractController&gt;&gt; M&gt;M, V&gt;V     }     class PictureSetListCreateController {         &lt;&lt;PictureSetListCreateController&gt;&gt;     }     AbstractController &lt; -- PictureSetListCreateController   </pre>	
owner	<a href="#">projectview</a>	
template parameters	name kind constrainingClassifier default M Class <a href="#">ProjectModel</a> V Class <a href="#">JProjectView&lt;M-&gt;M&gt;</a>	
ownedMember	<a href="#">actionPerformed</a> <a href="#">PictureSetListCreateController</a>	
general	<a href="#">AbstractController&lt;M&gt;M, V&gt;V</a>	ComponentRealization <a href="#">Implementierung</a>
target of relation		
documentation	Represents the Actions which are done by clicking on create pictureset. Acts in harmony with a JProjectView.	

## Operation PictureSetListCreateController::actionPerformed

owner	<a href="#">PictureSetListCreateController</a>
parameter	name direction type multiplicity default e in ActionEvent return return void
ownedMember	e return

## Operation PictureSetListCreateController::PictureSetListCreateController

owner	<a href="#">PictureSetListCreateController</a>
parameter	name direction type multiplicity default model in M view in V
ownedMember	model view

## Class PictureSetListDeleteController

diagram	<p><code>M&gt;ProjectModel,V&gt;JProjectView&lt;M-&gt;M&gt;</code></p> <p><b>PictureSetListDeleteController</b></p> <ul style="list-style-type: none"> <li><code>&lt;&lt;constructor&gt;&gt; PictureSetListDeleteController(in model:M, in view:V)</code></li> <li><code>&lt;&lt;annotations&gt;&gt; actionPerformed(in e:ActionEvent):void</code></li> <li><code>generateToDeleteText(in pictureSets:Object[]*, in toDelete:int[]*):String</code></li> </ul>
hierarchy	<pre> classDiagram     class AbstractController {         &lt;&lt;Annotations&gt;&gt; M-&gt;M, V-&gt;V     }     class PictureSetListDeleteController {         &lt;&lt;Annotations&gt;&gt; M-&gt;M, V-&gt;V     }     AbstractController &lt; -- PictureSetListDeleteController   </pre>
owner	<a href="#">projectview</a>
template parameters	name kind constrainingClassifier default <b>M Class ProjectModel V Class JProjectView&lt;M-&gt;M&gt;</b>
ownedMember	<a href="#">actionPerformed generateToDeleteText PictureSetListDeleteController</a>
general	<a href="#">AbstractController&lt;M-&gt;M,V-&gt;V&gt;</a>
target of relation	ComponentRealization <a href="#">Implementierung</a>
documentation	Represents the Actions which are done by clicking on delete pictureset. Acts in harmony with a JProjectView.

## Operation PictureSetListDeleteController::actionPerformed

owner	<a href="#">PictureSetListDeleteController</a>
parameter	name direction type multiplicity default <b>e in ActionEvent return return void</b>
ownedMember	<b>e return</b>

## Operation PictureSetListDeleteController::generateToDeleteText

owner	<a href="#">PictureSetListDeleteController</a>
parameter	name direction type multiplicity default <b>pictureSets in Object * toDelete in int * return return String</b>
ownedMember	<b>pictureSets return toDelete</b>

## Operation PictureSetListDeleteController::PictureSetListDeleteController

owner	<a href="#">PictureSetListDeleteController</a>
parameter	name direction type multiplicity default <b>model in M view in V</b>
ownedMember	<b>model view</b>

## Class PictureSetListDragController

diagram	<p>M&gt;ProjectModel,V&gt;JProjectView&lt;M-&gt;M&gt;</p> <p><b>PictureSetListDragController</b></p> <p>&lt;&lt;constructor&gt;&gt; PictureSetListDragController(<b>in</b> model:<b>M</b>, <b>in</b> view:<b>V</b>)</p> <p>&lt;&lt;annotations&gt;&gt; actionPerformed(<b>in</b> e:<b>ActionEvent</b>):<b>void</b></p>	<b>actionPerformed() (Operation)</b> <<annotations>> modifiers = Override
hierarchy	<pre> classDiagram     class AbstractController {         &lt;&lt;AbstractController&lt;M&gt;&gt;     }     class PictureSetListDragController {         &lt;&lt;PictureSetListDragController&gt;&gt;     }     AbstractController &lt; -- PictureSetListDragController   </pre>	
owner	<b>projectview</b>	
template parameters	name kind constrainingClassifier default <b>M</b> Class <b>ProjectModel</b> <b>V</b> Class <b>JProjectView&lt;M-&gt;M&gt;</b>	
ownedMember	<b>actionPerformed PictureSetListDragController</b>	
general	<b>AbstractController&lt;M-&gt;M,V-&gt;V&gt;</b>	ComponentRealization <b>Implementierung</b>
target of relation		
documentation	Represents the Actions which are done by editing the projectnamedescription. Acts in harmony with a JProjectView.	

## Operation PictureSetListDragController::actionPerformed

owner	<b>PictureSetListDragController</b>
parameter	name direction type multiplicity default <b>e</b> in <b>ActionEvent</b> return return void
ownedMember	<b>e</b> return

## Operation PictureSetListDragController::PictureSetListDragController

owner	<b>PictureSetListDragController</b>
parameter	name direction type multiplicity default <b>model</b> in <b>M</b> <b>view</b> in <b>V</b>
ownedMember	<b>model view</b>

## Class PictureThumbnailAdapter

diagram	
owner	<a href="#">ThumbnailAdapter</a>
ownedMember	<a href="#">PictureThumbnailAdapter</a>
target of relation	ComponentRealization <a href="#">Implementierung</a>

## Operation PictureThumbnailAdapter::PictureThumbnailAdapter

owner	<a href="#">PictureThumbnailAdapter</a>
-------	---

## Class PointPlotterWindow

diagram	<pre> classDiagram     class PointPlotterWindow {         &lt;&lt;final&gt;&gt; serialVersionUID:long=1L         xlabel:JLabel         ylabel:JLabel         zlabel:JLabel         istatusbar:JLabel     }     PointPlotterWindow &lt;&lt;constructor&gt;&gt; PointPlotterWindow()   </pre>
hierarchy	<pre> classDiagram     class PointPlotterWindow     class Component     class Container     class Accessible     class Window     class MenuContainer     class Frame     class WindowConstants     class Accessible     class RootPaneContainer     class JFrame     PointPlotterWindow --&gt; JFrame     JFrame --&gt; Frame     Frame --&gt; Window     Window --&gt; Container     Container --&gt; Component     Component &lt; -- Accessible     Container &lt; -- Accessible     Window &lt; -- MenuContainer     Window &lt; -- Accessible     Window &lt; -- RootPaneContainer   </pre>
owner	<a href="#">plotter3D</a>
ownedMember	<a href="#">jstatusbar</a> <a href="#">PointPlotterWindow</a> <a href="#">serialVersionUID</a> <a href="#">xLabel</a> <a href="#">yLabel</a> <a href="#">zLabel</a>
general	<a href="#">JFrame</a>
target of relation	ComponentRealization <a href="#">Implementierung</a>
typedElements	Class <a href="#">Pointplotter</a> Property <a href="#">frame</a>

## Property PointPlotterWindow::jstatusbar

owner	<a href="#">PointPlotterWindow</a>
-------	------------------------------------

## Operation PointPlotterWindow::PointPlotterWindow

owner	<a href="#">PointPlotterWindow</a>
-------	------------------------------------

## Property PointPlotterWindow::serialVersionUID

owner	<a href="#">PointPlotterWindow</a>
documentation	

## Property PointPlotterWindow::xLabel

owner	<a href="#">PointPlotterWindow</a>
-------	------------------------------------

Property **PointPlotterWindow::yLabel**

owner	<a href="#">PointPlotterWindow</a>
-------	------------------------------------

Property **PointPlotterWindow::zLabel**

owner	<a href="#">PointPlotterWindow</a>
-------	------------------------------------

## Class Pointplotter

diagram	<pre> <b>Pointplotter</b>      &lt;&lt;final&gt;&gt; serialVersionUID:long=1L     &lt;&lt;final&gt;&gt; AXISSIZE:double=10.0     &lt;&lt;final&gt;&gt; BOXDENSITYFACTOR:double=0.25     &lt;&lt;final&gt;&gt; GRIDDENSITYFACTOR:double=1     &lt;&lt;final&gt;&gt; DATAAMOUNT:int=100     &lt;&lt;final&gt;&gt; GEODETAIL:int=10     &lt;&lt;final&gt;&gt; OBJROOT:BranchGroup=new BranchGroup()     canvas3D:Canvas3D=null     simpleU:SimpleUniverse=null     pickCanvas:PickCanvas     frame:PointPlotterWindow      basicMaterial(in x:float, in y:float, in z:float):Appearance     createAxis():void     createGrid():void     createBackground(in color:Color3f):void     createSingleText(in pos:Vector3d, in text:String, in size:double):void     createLabels():void     makeInteractive():void     changeCameraPosition(in simpleU:SimpleUniverse, in x:float, in y:float, in z:float):void     insertData():void     addLights():void     addSingleBar(in z:double):void     addAllBars():void     createSceneGraph(in canvas3d:Canvas3D):BranchGroup     &lt;&lt;constructor&gt;&gt; Pointplotter()     mouseClicked(in e:MouseEvent):void     main(in args:String[]):void      myShape3D   </pre>
hierarchy	<pre> classDiagram     class Pointplotter {         &lt;&lt;MouseListener&gt;&gt;         &lt;&lt;MouseMotionListener&gt;&gt;         &lt;&lt;MouseWheelListener&gt;&gt;     }     class MouseAdapter {         &lt;&lt;MouseListener&gt;&gt;         &lt;&lt;MouseMotionListener&gt;&gt;         &lt;&lt;MouseWheelListener&gt;&gt;     }     class EventListener     class MouseListener     class MouseMotionListener     class MouseWheelListener     Pointplotter --&gt; MouseAdapter     MouseAdapter --&gt; EventListener     MouseAdapter --&gt; MouseListener     MouseAdapter --&gt; MouseMotionListener     MouseAdapter --&gt; MouseWheelListener   </pre>
owner	<a href="#">plotter3D</a>
ownedMember	<a href="#">addAllBars</a> <a href="#">addLights</a> <a href="#">addSingleBar</a> <a href="#">AXISSIZE</a> <a href="#">basicMaterial</a> <a href="#">BOXDENSITYFACTOR</a> <a href="#">canvas3D</a> <a href="#">changeCameraPosition</a> <a href="#">createAxis</a> <a href="#">createBackground</a> <a href="#">createGrid</a> <a href="#">createLabels</a> <a href="#">createSceneGraph</a> <a href="#">createSingleText</a> <a href="#">DATAAMOUNT</a> <a href="#">frame</a> <a href="#">GEODETAIL</a> <a href="#">GRIDDENSITYFACTOR</a> <a href="#">insertData</a> <a href="#">main</a> <a href="#">makeInteractive</a> <a href="#">mouseClicked</a> <a href="#">myShape3D</a> <a href="#">OBJROOT</a> <a href="#">pickCanvas</a> <a href="#">Pointplotter</a> <a href="#">serialVersionUID</a> <a href="#">simpleU</a>

general	<b>MouseAdapter</b>
target of relation	ComponentRealization <a href="#">Implementierung</a>

#### Operation **Pointplotter::addAllBars**

owner	<b>Pointplotter</b>
parameter	name direction type multiplicity default <b>return return void</b>
ownedMember	<b>return</b>

#### Operation **Pointplotter::addLights**

owner	<b>Pointplotter</b>
parameter	name direction type multiplicity default <b>return return void</b>
ownedMember	<b>return</b>

#### Operation **Pointplotter::addSingleBar**

owner	<b>Pointplotter</b>
parameter	name direction type multiplicity default <b>z in double return return void</b>
ownedMember	<b>return z</b>

#### Property **Pointplotter::AXISSIZE**

owner	<b>Pointplotter</b>
-------	---------------------

#### Operation **Pointplotter::basicMaterial**

owner	<b>Pointplotter</b>
parameter	name direction type multiplicity default <b>x in float y in float z in float return return Appearance</b>
ownedMember	<b>return x y z</b>

#### Property **Pointplotter::BOXDENSITYFACTOR**

owner	<b>Pointplotter</b>
-------	---------------------

#### Property **Pointplotter::canvas3D**

owner	<b>Pointplotter</b>
-------	---------------------

#### Operation **Pointplotter::changeCameraPosition**

owner	<b>Pointplotter</b>
parameter	name direction type multiplicity default <b>simpleU in SimpleUniverse x in float y in float z in float return return void</b>
ownedMember	<b>return simpleU x y z</b>

#### Operation **Pointplotter::createAxis**

owner	<a href="#">Pointplotter</a>
parameter	name direction type multiplicity default <b>return return void</b>
ownedMember	<b>return</b>

#### Operation **Pointplotter::createBackground**

owner	<a href="#">Pointplotter</a>
parameter	name direction type multiplicity default <b>color in Color3f return return void</b>
ownedMember	<b>color return</b>

#### Operation **Pointplotter::createGrid**

owner	<a href="#">Pointplotter</a>
parameter	name direction type multiplicity default <b>return return void</b>
ownedMember	<b>return</b>

#### Operation **Pointplotter::createLabels**

owner	<a href="#">Pointplotter</a>
parameter	name direction type multiplicity default <b>return return void</b>
ownedMember	<b>return</b>

#### Operation **Pointplotter::createSceneGraph**

owner	<a href="#">Pointplotter</a>
parameter	name direction type multiplicity default <b>canvas3d in Canvas3D return return BranchGroup</b>
ownedMember	<b>canvas3d return</b>

#### Operation **Pointplotter::createSingleText**

owner	<a href="#">Pointplotter</a>
parameter	name direction type multiplicity default <b>pos in Vector3d text in String size in double return return void</b>
ownedMember	<b>pos return size text</b>

#### Property **Pointplotter::DATAAMOUNT**

owner	<a href="#">Pointplotter</a>
-------	------------------------------

#### Property **Pointplotter::frame**

owner	<a href="#">Pointplotter</a>
-------	------------------------------

#### Property Pointplotter::GEODETAIL

owner	<a href="#">Pointplotter</a>
-------	------------------------------

#### Property Pointplotter::GRIDDENSITYFACTOR

owner	<a href="#">Pointplotter</a>
-------	------------------------------

#### Operation Pointplotter::insertData

owner	<a href="#">Pointplotter</a>
parameter	name direction type multiplicity default <b>return return void</b>
ownedMember	<b>return</b>

#### Operation Pointplotter::main

owner	<a href="#">Pointplotter</a>
parameter	name direction type multiplicity default <b>args in String * return return void</b>
ownedMember	<b>args return</b>

#### Operation Pointplotter::makeInteractive

owner	<a href="#">Pointplotter</a>
parameter	name direction type multiplicity default <b>return return void</b>
ownedMember	<b>return</b>

#### Operation Pointplotter::mouseClicked

owner	<a href="#">Pointplotter</a>
parameter	name direction type multiplicity default <b>e in MouseEvent return return void</b>
ownedMember	<b>e return</b>

#### Class Pointplotter::myShape3D

diagram	
hierarchy	
owner	<a href="#">Pointplotter</a>

ownedMember	<a href="#">id myShape3D</a>
general	<a href="#">Shape3D</a>

#### Property **Pointplotter::myShape3D::id**

owner	<a href="#">myShape3D</a>
-------	---------------------------

#### Operation **Pointplotter::myShape3D::myShape3D**

owner	<a href="#">myShape3D</a>
parameter	name direction type multiplicity default <b>x in float y in float z in float</b>
ownedMember	<b>x y z</b>

#### Property **Pointplotter::OBJROOT**

owner	<a href="#">Pointplotter</a>
-------	------------------------------

#### Property **Pointplotter::pickCanvas**

owner	<a href="#">Pointplotter</a>
-------	------------------------------

#### Operation **Pointplotter::Pointplotter**

owner	<a href="#">Pointplotter</a>
-------	------------------------------

#### Property **Pointplotter::serialVersionUID**

owner	<a href="#">Pointplotter</a>
-------	------------------------------

#### Property **Pointplotter::simpleU**

owner	<a href="#">Pointplotter</a>
-------	------------------------------

## Class Programm

diagram	
owner	<a href="#">knipsX</a>
ownedMember	<a href="#">main</a>
target of relation	ComponentRealization <a href="#">Implementierung</a>
documentation	This class is the entry to our program.

## Operation Programm::main

owner	<a href="#">Programm</a>
parameter	name direction type multiplicity default <b>args in String *</b> <b>return return void</b>
ownedMember	<b>args return</b>
documentation	<p>Starts knipsX.</p> <p>This function shows the first knipsX window.</p> <p><b>@param args</b> stores parameters which are set up by a user during program start.</p>

## Class ProjectCopyController

diagram	<p>The diagram shows the class <b>ProjectCopyController</b> with its dependencies. It has a dependency on <b>M Class ProjectManagementModel</b> and <b>V Class JProjectManagement&lt;M-&gt;?</b>. The class itself is shaded grey. A callout box highlights the operation <b>actionPerformed()</b> with the annotation <code>&lt;&lt;annotations&gt;&gt; modifiers = Override</code>.</p> <pre> classDiagram     class ProjectCopyController {         &lt;&lt;constructor&gt;&gt; ProjectCopyController(in model:M, in view:V)         &lt;&lt;annotations&gt;&gt; actionPerformed(in e:ActionEvent):void     }     M "1..&gt;" ProjectCopyController     V "1..&gt;" ProjectCopyController     ProjectCopyController --&gt; AbstractController     </pre>
hierarchy	<p>A class hierarchy diagram showing <b>AbstractController&lt;M-&gt;M,V-&gt;V&gt;</b> as the superclass and <b>ProjectCopyController</b> as the subclass. An arrow points from the superclass to the subclass.</p> <pre> classDiagram     class AbstractController&lt;M-&gt;M,V-&gt;V&gt;     class ProjectCopyController     AbstractController --&gt; ProjectCopyController     </pre>
owner	<b>projectmanagement</b>
template parameters	name kind constrainingClassifier default <b>M Class ProjectManagementModel</b> <b>V Class JProjectManagement&lt;M-&gt;?</b>
ownedMember	<b>actionPerformed ProjectCopyController</b>
general	<b>AbstractController&lt;M-&gt;M,V-&gt;V&gt;</b>
target of relation	ComponentRealization <b>Implementierung</b>
documentation	<p>Represents the actions which are done by pushing the project copy button.</p> <p>Acts in harmony with JProjectManagement.</p>

## Operation ProjectCopyController::actionPerformed

owner	<b>ProjectCopyController</b>
parameter	name direction type multiplicity default <b>e in ActionEvent return return void</b>
ownedMember	<b>e return</b>

## Operation ProjectCopyController::ProjectCopyController

owner	<b>ProjectCopyController</b>
parameter	name direction type multiplicity default <b>model in M view in V</b>
ownedMember	<b>model view</b>

## Class ProjectCreateController

diagram	<pre> M&gt;ProjectManagementModel,V&gt;JProjectManagement&lt;M-&gt;M&gt; ProjectCreateController &lt;&lt;constructor&gt;&gt; ProjectCreateController(in model:M, in view:V) &lt;&lt;annotations&gt;&gt; actionPerformed(in e:ActionEvent):void   </pre>	<b>actionPerformed() (Operation)</b> <b>&lt;&lt;annotations&gt;&gt;</b> modifiers = Override
hierarchy	<pre> AbstractController&lt;M-&gt;M,V-&gt;V&gt;   </pre>	
owner	<a href="#">projectmanagement</a>	
template parameters		name kind constrainingClassifier default M Class <a href="#">ProjectManagementModel</a> V Class <a href="#">JProjectManagement&lt;M-&gt;M&gt;</a>
ownedMember	<a href="#">actionPerformed ProjectCreateController</a>	
general	<a href="#">AbstractController&lt;M-&gt;M,V-&gt;V&gt;</a>	ComponentRealization <a href="#">Implementierung</a>
target of relation		
documentation	<p>Represents the actions which are done by pushing the project create button.</p> <p>Acts in harmony with JProjectManagement.</p>	

## Operation ProjectCreateController::actionPerformed

owner	<a href="#">ProjectCreateController</a>
parameter	name direction type multiplicity default e in ActionEvent return return void
ownedMember	e return

## Operation ProjectCreateController::ProjectCreateController

owner	<a href="#">ProjectCreateController</a>
parameter	name direction type multiplicity default model in M view in V
ownedMember	model view

## Class ProjectDeleteController

diagram	<p>The diagram shows the class <b>ProjectDeleteController</b> with its superclass <b>AbstractController&lt;M-&gt;M,V-&gt;V&gt;</b>. It has three operations: <code>&lt;&lt;constructor&gt;&gt; ProjectDeleteController(in model:M, in view:V)</code>, <code>&lt;&lt;annotations&gt;&gt; actionPerformed(in e:ActionEvent):void</code>, and <code>&lt;&lt;annotation&gt;&gt; generateToDeleteText(in toDelete:int[]):String</code>. The <code>actionPerformed()</code> operation is annotated with <code>&lt;&lt;annotations&gt;&gt; modifiers = Override</code>.</p>
hierarchy	<pre> classDiagram     class AbstractController {         &lt;&lt;constructor&gt;&gt; AbstractController&lt;M-&gt;M,V-&gt;V&gt;     }     class ProjectDeleteController {         &lt;&lt;constructor&gt;&gt; ProjectDeleteController(in model:M, in view:V)         &lt;&lt;annotations&gt;&gt; actionPerformed(in e:ActionEvent):void         &lt;&lt;annotation&gt;&gt; generateToDeleteText(in toDelete:int[]):String     }     AbstractController &lt; -- ProjectDeleteController   </pre>
owner	<b>projectmanagement</b>
template parameters	name kind constrainingClassifier default M Class <b>ProjectManagementModel</b> V Class <b>JProjectManagement&lt;M-&gt;M&gt;</b>
ownedMember	<b>actionPerformed</b> <b>generateToDeleteText</b> <b>ProjectDeleteController</b>
general	<b>AbstractController&lt;M-&gt;M,V-&gt;V&gt;</b>
target of relation	ComponentRealization <b>Implementierung</b>
documentation	<p>Represents the actions which are done by pushing the project delete button.</p> <p>Acts in harmony with JProjectManagement.</p>

### Operation ProjectDeleteController::actionPerformed

owner	<b>ProjectDeleteController</b>
parameter	name direction type multiplicity default <b>e</b> in <b>ActionEvent</b> return return void
ownedMember	<b>e</b> return

### Operation ProjectDeleteController::generateToDeleteText

owner	<b>ProjectDeleteController</b>
parameter	name direction type multiplicity default <b>toDelete</b> in <b>int *</b> return return String
ownedMember	<b>return toDelete</b>

### Operation ProjectDeleteController::ProjectDeleteController

owner	<b>ProjectDeleteController</b>
parameter	name direction type multiplicity default <b>model</b> in <b>M</b> <b>view</b> in <b>V</b>
ownedMember	<b>model</b> <b>view</b>



## Class ProjectEditDescriptionController

diagram	<p>The diagram shows the class <b>ProjectEditDescriptionController</b> with its constructor and an operation <b>actionPerformed()</b>. The class is associated with <b>ProjectModel</b> and <b>JProjectView</b>. The operation <b>actionPerformed()</b> is annotated with <code>&lt;&lt;annotations&gt;&gt;</code> and has modifiers set to <code>Override</code>.</p>
hierarchy	<pre> classDiagram     class AbstractController {         &lt;&lt;constructor&gt;&gt; AbstractController&lt;M&gt; M, &lt;&lt;constructor&gt;&gt; AbstractController&lt;V&gt; V     }     class ProjectEditDescriptionController {         &lt;&lt;constructor&gt;&gt; ProjectEditDescriptionController&lt;in model:M, in view:V&gt;         &lt;&lt;operations&gt;&gt; actionPerformed&lt;in e:ActionEvent&gt; void     }     AbstractController &lt; -- ProjectEditDescriptionController   </pre>
owner	<a href="#">projectview</a>
template parameters	name kind constrainingClassifier default <b>M</b> Class <a href="#">ProjectModel</a> <b>V</b> Class <a href="#">JProjectView&lt;M&gt;M</a>
ownedMember	<a href="#">actionPerformed</a> <a href="#">ProjectEditDescriptionController</a>
general	<a href="#">AbstractController&lt;M&gt;M, V&gt;V</a>
target of relation	ComponentRealization <a href="#">Implementierung</a>
documentation	Represents the Actions which are done by editing the projectdescription. Acts in harmony with a JProjectView.

## Operation ProjectEditDescriptionController::actionPerformed

owner	<a href="#">ProjectEditDescriptionController</a>
parameter	name direction type multiplicity default <b>e</b> in <b>ActionEvent</b> return return void
ownedMember	<b>e</b> return

## Operation ProjectEditDescriptionController::ProjectEditDescriptionController

owner	<a href="#">ProjectEditDescriptionController</a>
parameter	name direction type multiplicity default <b>model</b> in <b>M</b> <b>view</b> in <b>V</b>
ownedMember	<b>model</b> <b>view</b>

## Class ProjectEditNameController

diagram	<p>ProjectEditNameController</p> <p><code>&lt;&lt;constructor&gt;&gt; ProjectEditNameController(in model:M, in view:V)</code></p> <p><code>&lt;&lt;annotations&gt;&gt; actionPerformed(in e:ActionEvent):void</code></p>	<b>actionPerformed() (Operation)</b> <code>&lt;&lt;annotations&gt;&gt; modifiers = Override</code>
hierarchy	<pre> classDiagram     class AbstractController {         &lt;&lt;M-&gt;M, V-&gt;V&gt;&gt;     }     class ProjectEditNameController {         &lt;&lt;M-&gt;M, V-&gt;V&gt;&gt;     }     AbstractController &lt; -- ProjectEditNameController   </pre>	
owner	<a href="#">projectview</a>	
template parameters	name kind constrainingClassifier default M Class <a href="#">ProjectModel</a> V Class <a href="#">JProjectView&lt;M-&gt;M&gt;</a>	
ownedMember	<a href="#">actionPerformed ProjectEditNameController</a>	
general	<a href="#">AbstractController&lt;M-&gt;M,V-&gt;V&gt;</a>	ComponentRealization <a href="#">Implementierung</a>
target of relation		
documentation	<p>Represents the actions which are done by editing the project name.</p> <p>Acts in harmony with a JProjectView.</p>	

## Operation ProjectEditNameController::actionPerformed

owner	<a href="#">ProjectEditNameController</a>
parameter	name direction type multiplicity default e in ActionEvent return return void
ownedMember	e return

## Operation ProjectEditNameController::ProjectEditNameController

owner	<a href="#">ProjectEditNameController</a>
parameter	name direction type multiplicity default model in M view in V
ownedMember	model view

## Class ProjectEntry

diagram	<pre> <b>ProjectEntry</b>  id:int projectName:String projectDescription:String creationDate:GregorianCalendar path:String  &lt;&lt;constructor&gt;&gt; ProjectEntry(in projectName:String) &lt;&lt;constructor&gt;&gt; ProjectEntry(in id:int, in projectName:String, in projectDescription:String, in creationDate:GregorianCalendar, in path:String) &lt;&lt;constructor&gt;&gt; ProjectEntry(in projectEntry:ProjectEntry, in projectName:String) &lt;&lt;constructor&gt;&gt; ProjectEntry(in projectEntry:ProjectEntry)  generateFreeProjectId():int getPathForProject():String getId():int getProjectName():String setProjectName(in projectName:String):void getProjectDescription():String setProjectDescription(in projectDescription:String):void getCreationDate():GregorianCalendar getPath():String calendarToString():String </pre>
hierarchy	<pre> classDiagram     class Observable     class AbstractModel {         &lt;&lt;Abstract Model&gt;&gt;     }     class ProjectEntry {         &lt;&lt;Project Entry&gt;&gt;     }     class ProjectModel {         &lt;&lt;Project Model&gt;&gt;     }      Observable &lt; -- AbstractModel     AbstractModel &lt; -- ProjectEntry     ProjectEntry &lt; -- ProjectModel </pre>
owner	<a href="#">projectview</a>
ownedMember	<a href="#">calendarToString</a> <a href="#">creationDate</a> <a href="#">generateFreeProjectId</a> <a href="#">getCreationDate</a> <a href="#">getId</a> <a href="#">getPath</a> <a href="#">getPathForProject</a> <a href="#">getProjectDescription</a> <a href="#">getProjectName</a> <a href="#">id</a> <a href="#">path</a> <a href="#">projectDescription</a> <a href="#">ProjectEntry</a> <a href="#">ProjectEntry</a> <a href="#">ProjectEntry</a> <a href="#">projectName</a> <a href="#">setProjectDescription</a> <a href="#">setProjectName</a>
general	<a href="#">AbstractModel</a>
specific	<a href="#">ProjectModel</a>
target of relation	ComponentRealization <a href="#">Implementierung</a>
typedElements	Class <a href="#">FileHandler</a> Operation <a href="#">copyProject</a> <a href="#">createNewProjectFile</a> <a href="#">deleteProjectFile</a> <a href="#">manipulateFileNameAndID</a> Class <a href="#">ProjectEntry</a> Operation <a href="#">ProjectEntry</a> Class <a href="#">ProjectManagementModel</a> Operation <a href="#">copyProject</a> <a href="#">removeProject</a> Class <a href="#">ProjectModel</a> Operation <a href="#">getInstance</a> <a href="#">ProjectModel</a>
documentation	Represents a project entry (only meta data).

## Operation ProjectEntry::calendarToString

owner	<a href="#">ProjectEntry</a>
parameter	name direction type multiplicity default <b>return return String</b>
ownedMember	<b>return</b>
documentation	

on	Get the creation date of the project as readable String.  @return the creation date of the project as readable String.
----	--

#### Property **ProjectEntry::creationDate**

owner	<a href="#">ProjectEntry</a>
-------	------------------------------

#### Operation **ProjectEntry::generateFreeProjectId**

owner	<a href="#">ProjectEntry</a>
parameter	name direction type multiplicity default <b>return return int</b>
ownedMember	<b>return</b>

#### Operation **ProjectEntry::getCreationDate**

owner	<a href="#">ProjectEntry</a>
parameter	name direction type multiplicity default <b>return return GregorianCalendar</b>
ownedMember	<b>return</b>
documentation	Get the creation date of the project.  @return the creation date of the project.

#### Operation **ProjectEntry::getId**

owner	<a href="#">ProjectEntry</a>
parameter	name direction type multiplicity default <b>return return int</b>
ownedMember	<b>return</b>
documentation	Get the id of the project.  @return the id of the project.

#### Operation **ProjectEntry::getPath**

owner	<a href="#">ProjectEntry</a>
parameter	name direction type multiplicity default <b>return return String</b>
ownedMember	<b>return</b>
documentation	Get the path to the project configuration file.  @return the path to the project configuration file.

#### Operation **ProjectEntry::getPathForProject**

owner	<a href="#">ProjectEntry</a>
-------	------------------------------

parameter	name direction type multiplicity default <b>return return String</b>
ownedMember	<b>return</b>

#### Operation **ProjectEntry::getProjectDescription**

owner	<b>ProjectEntry</b>
parameter	name direction type multiplicity default <b>return return String</b>
ownedMember	<b>return</b>
documentation	<p>Get the description of the project.</p> <p>@return the description of the project.</p>

#### Operation **ProjectEntry::getProjectName**

owner	<b>ProjectEntry</b>
parameter	name direction type multiplicity default <b>return return String</b>
ownedMember	<b>return</b>
documentation	<p>Get the name of the project.</p> <p>@return the name of the project.</p>

#### Property **ProjectEntry::id**

owner	<b>ProjectEntry</b>
-------	---------------------

#### Property **ProjectEntry::path**

owner	<b>ProjectEntry</b>
-------	---------------------

#### Property **ProjectEntry::projectDescription**

owner	<b>ProjectEntry</b>
-------	---------------------

#### Operation **ProjectEntry::ProjectEntry**

owner	<b>ProjectEntry</b>
parameter	name direction type multiplicity default <b>projectName in String</b>
ownedMember	<b>projectName</b>
documentation	<p>Creates a new project entry from scratch.</p> <p>@param projectName the name of the project.</p>

### Operation ProjectEntry::ProjectEntry

owner	<a href="#">ProjectEntry</a>
parameter	name direction type multiplicity default <b>id</b> in int <b>projectName</b> in String <b>projectDescription</b> in String <b>creationDate</b> in GregorianCalendar <b>path</b> in String
ownedMember	<b>creationDate</b> <b>id</b> <b>path</b> <b>projectDescription</b> <b>projectName</b>
documentation	<p>Creates a new project entry with given details.</p> <p> <b>@param id</b>          the id of the project.  <b>@param projectName</b>          the name of the project.  <b>@param projectDescription</b>          the description of the project.  <b>@param creationDate</b>          the creation date of the project  <b>@param path</b>          the path to the project configuration file.       </p>

### Operation ProjectEntry::ProjectEntry

owner	<a href="#">ProjectEntry</a>
parameter	name direction type multiplicity default <b>projectEntry</b> in <a href="#">ProjectEntry</a> <b>projectName</b> in String
ownedMember	<b>projectEntry</b> <b>projectName</b>
documentation	<p>Creates a new project as copy from another project. This constructor call forces a recreation of the project id.</p> <p> <b>@param projectEntry</b>          the other project to copy from.  <b>@param projectName</b>          the name of the new project.       </p>

### Operation ProjectEntry::ProjectEntry

owner	<a href="#">ProjectEntry</a>
parameter	name direction type multiplicity default <b>projectEntry</b> in <a href="#">ProjectEntry</a>
ownedMember	<b>projectEntry</b>
documentation	<p>Creates a new project as copy from another project.</p> <p> <b>@param projectEntry</b>          the other project to copy from.       </p>

### Property ProjectEntry::projectName

owner	<a href="#">ProjectEntry</a>
-------	------------------------------

### Operation ProjectEntry::setProjectDescription

owner	<a href="#">ProjectEntry</a>
parameter	name direction type multiplicity default <b>projectDescription</b> in String <b>return</b> return void
ownedMember	<b>projectDescription</b> <b>return</b>

ber	
documentati on	<p>Set the description of the project.</p> <p>@return the description of the project.</p>

### Operation **ProjectEntry::setProjectName**

owner	<a href="#">ProjectEntry</a>
parameter	name direction type multiplicity default <b>projectName</b> in String return return void
ownedMem ber	<b>projectName</b> return
documentati on	<p>Set the name of the project.</p> <p>@return the name of the project.</p>

## Class ProjectManagementModel

diagram	<pre> <b>ProjectManagementModel</b>  &lt;&lt;final&gt;&gt; ACTIVE:int=1 &lt;&lt;final&gt;&gt; INACTIVE:int=0 state:int=ACTIVE projectList:List&lt;E-&gt;ProjectEntry&gt;  &lt;&lt;constructor&gt;&gt; ProjectManagementModel(in linkedList:List&lt;E-&gt;ProjectEntry&gt;) getProjectList():List&lt;E-&gt;ProjectEntry&gt; setProjectlist(in projectlist:List&lt;E-&gt;ProjectEntry&gt;):void setState(in state:int):void getState():int addNewProject(in projectName:String):void removeProject(in project:ProjectEntry):void copyProject(in toCopy:ProjectEntry, in projectName:String):void generateFreeProjectID():int generatePathforID(in id:int):String </pre>
hierarchy	<pre> classDiagram     class Observable     class AbstractModel {         &lt;&lt;Abstract Model&gt;&gt;     }     class ProjectManagementModel {         &lt;&lt;Project Management Model&gt;&gt;     }     Observable &lt; -- AbstractModel     AbstractModel &lt; -- ProjectManagementModel </pre>
owner	<a href="#">projectmanagement</a>
ownedMember	<a href="#">ACTIVE</a> <a href="#">addNewProject</a> <a href="#">copyProject</a> <a href="#">generateFreeProjectID</a> <a href="#">generatePathforID</a> <a href="#">getProjectList</a> <a href="#">getState</a> <a href="#">INACTIVE</a> <a href="#">projectList</a> <a href="#">ProjectManagementModel</a> <a href="#">removeProject</a> <a href="#">setProjectlist</a> <a href="#">setState</a> <a href="#">state</a>
general	<a href="#">AbstractModel</a>
target of relation	ComponentRealization <a href="#">Implementierung</a>
documentation	Manages all the projects the user has.

## Property ProjectManagementModel::ACTIVE

owner	<a href="#">ProjectManagementModel</a>
-------	--

## Operation ProjectManagementModel::addNewProject

owner	<a href="#">ProjectManagementModel</a>
parameter	name direction type multiplicity default <b>projectName</b> in String return void
ownedMember	<b>projectName</b> return
documentation	Add a new project to the model.  <b>@param</b> <b>projectName</b> the name of the project.

--	--

#### Operation **ProjectManagementModel::copyProject**

owner	<b>ProjectManagementModel</b>
parameter	name direction type multiplicity default <b>toCopy</b> in <a href="#">ProjectEntry</a> <b>projectName</b> in String <b>return return void</b>
ownedMember	<b>projectName return toCopy</b>
documentation	<p>Create a new model based on an old model.</p> <p>@param <b>toCopy</b> the old model.</p> <p>@param <b>projectName</b> the name of the new project.</p>

#### Operation **ProjectManagementModel::generateFreeProjectID**

owner	<b>ProjectManagementModel</b>
parameter	name direction type multiplicity default <b>return return int</b>
ownedMember	<b>return</b>

#### Operation **ProjectManagementModel::generatePathforID**

owner	<b>ProjectManagementModel</b>
parameter	name direction type multiplicity default <b>id in int return return String</b>
ownedMember	<b>id return</b>

#### Operation **ProjectManagementModel::getProjectList**

owner	<b>ProjectManagementModel</b>
parameter	name direction type multiplicity default <b>return return List&lt;E-&gt;ProjectEntry&gt;</b>
ownedMember	<b>return</b>

#### Operation **ProjectManagementModel::getState**

owner	<b>ProjectManagementModel</b>
parameter	name direction type multiplicity default <b>return return int</b>
ownedMember	<b>return</b>

#### Property **ProjectManagementModel::INACTIVE**

owner	<b>ProjectManagementModel</b>
-------	-------------------------------

#### Property **ProjectManagementModel::projectList**

owner	<b>ProjectManagementModel</b>
-------	-------------------------------

#### Operation ProjectManagementModel::ProjectManagementModel

owner	<b>ProjectManagementModel</b>
parameter	name direction type multiplicity default <b>linkedList</b> in <a href="#">List&lt;E&gt;ProjectEntry</a>
ownedMember	<b>linkedList</b>
documentation	<p>Creates a project management model based on projects.</p> <p>@param linkedList the projects.</p>

#### Operation ProjectManagementModel::removeProject

owner	<b>ProjectManagementModel</b>
parameter	name direction type multiplicity default <b>project</b> in <a href="#">ProjectEntry</a> return return void
ownedMember	<b>project return</b>
documentation	<p>Remove a project from the model.</p> <p>@param project the project.</p>

#### Operation ProjectManagementModel::setProjectlist

owner	<b>ProjectManagementModel</b>
parameter	name direction type multiplicity default <b>projectlist</b> in <a href="#">List&lt;E&gt;ProjectEntry</a> return return void
ownedMember	<b>projectlist return</b>

#### Operation ProjectManagementModel::setState

owner	<b>ProjectManagementModel</b>
parameter	name direction type multiplicity default <b>state</b> in int return return void
ownedMember	<b>return state</b>

#### Property ProjectManagementModel::state

owner	<b>ProjectManagementModel</b>
-------	-------------------------------

## Class ProjectModel

diagram	<pre> <b>ProjectModel</b>  ■ pictureSetList:List&lt;E-&gt;PictureSet&gt; ■ reportList:List&lt;E-&gt;AbstractReportModel&gt; ■ exifParameter:Object[*][*] ■ lastModel:ProjectModel=null  ● getINSTANCE(in projectEntry:ProjectEntry, in pictureSets:List&lt;E-&gt;PictureSet&gt;, in reports:List&lt;E-&gt;AbstractReportModel&gt;):ProjectModel ● &lt;&lt;constructor&gt;&gt; ProjectModel(in projectEntry:ProjectEntry, in pictureSets:List&lt;E-&gt;PictureSet&gt;, in reports:List&lt;E-&gt;AbstractReportModel&gt;) ● addPictureSet(in pictureSet:PictureSet):boolean ● removePictureSet(in pictureSet:PictureSet):boolean ● addContentToPictureSet(in pictureSet:PictureSet, in pictureContainer:PictureContainer):boolean ● removeContentFromPictureSet(in pictureSet:PictureSet, in pictureContainer:PictureContainer):boolean ● addReport(in report:AbstractReportModel):boolean ● removeReport(in report:AbstractReportModel):boolean ● destroy():void ● getExifParameter():Object[*][*] ● getPictureSets():Object[*] ● getPictureSetsOfAPictureSet(in pictureSet:PictureSet):PictureSet[*] ● getDirectoriesOfAPictureSet(in pictureSet:PictureSet):Directory[*] ● getPicturesOfAPictureSet(in pictureSet:PictureSet):Picture[*] ● getPicturesOfADirectory(in directory:Directory):Picture[*] ● getAllPictures():Picture[*] ● getReports():Object[*] ● saveProjectModel():void </pre>
hierarchy	<pre> classDiagram     class Observable     class AbstractModel {         &lt;&lt;Observable&gt;&gt;     }     class ProjectEntry {         &lt;&lt;AbstractModel&gt;&gt;     }     class ProjectModel {         &lt;&lt;ProjectEntry&gt;&gt;     }     ProjectModel --&gt; ProjectEntry     ProjectEntry --&gt; AbstractModel     AbstractModel --&gt; Observable </pre>
owner	<a href="#">projectview</a>
ownedMember	<a href="#">addContentToPictureSet</a> <a href="#">addPictureSet</a> <a href="#">addReport</a> <a href="#">destroy</a> <a href="#">exifParameter</a> <a href="#">getAllPictures</a> <a href="#">getDirectoriesOfAPictureSet</a> <a href="#">getExifParameter</a> <a href="#">getInstance</a> <a href="#">getPictureSets</a> <a href="#">getPictureSetsOfAPictureSet</a> <a href="#">getPicturesOfADirectory</a> <a href="#">getPicturesOfAPictureSet</a> <a href="#">getReports</a> <a href="#">lastModel</a> <a href="#">pictureSetList</a> <a href="#">ProjectModel</a> <a href="#">removeContentFromPictureSet</a> <a href="#">removePictureSet</a> <a href="#">removeReport</a> <a href="#">reportList</a> <a href="#">saveProjectModel</a>
general	<a href="#">ProjectEntry</a>
target of relation	ComponentRealization <a href="#">Implementierung</a>
typedElements	Class <a href="#">FileHandler</a> Operation <a href="#">scanProjectFile</a> <a href="#">writeProjectToFile</a> Class <a href="#">ProjectModel</a> Property <a href="#">lastModel</a> Operation <a href="#">getInstance</a> Enumeration <a href="#">ReportHelper</a> Property <a href="#">currentProjectModel</a>
documentation	Manages all the data for an active project.

### Operation ProjectModel::addContentToPictureSet

owner	<a href="#">ProjectModel</a>
parameter	name direction type multiplicity default <b>pictureSet</b> in <a href="#">PictureSet</a> <b>pictureContainer</b> in <a href="#">PictureContainer</a> return <b>return boolean</b>

ownedMember	<b>pictureContainer pictureSet return</b>
documentation	<p>Add a picture set content of a picture set.</p> <p>@param pictureSet the picture set where the content must be added.</p> <p>@param pictureContainer the content which must be added.</p> <p>@return true if the picture set content was added, false if not.</p>

#### Operation ProjectModel::addPictureSet

owner	<b>ProjectModel</b>
parameter	name direction type multiplicity default <b>pictureSet in PictureSet</b> return return boolean
ownedMember	<b>pictureSet return</b>
documentation	<p>Add a picture set to the current project.</p> <p>@param pictureSet the picture set to add.</p> <p>@return true if the picture set was added, false if not.</p>

#### Operation ProjectModel::addReport

owner	<b>ProjectModel</b>
parameter	name direction type multiplicity default <b>report in AbstractReportModel</b> return return boolean
ownedMember	<b>report return</b>
documentation	<p>Add a report to the current project.</p> <p>@param report the report to add.</p> <p>@return true if the report was added, false if not.</p>

#### Operation ProjectModel::destroy

owner	<b>ProjectModel</b>
parameter	name direction type multiplicity default <b>return return void</b>
ownedMember	<b>return</b>
documentation	This method destroys the model.

#### Property ProjectModel::exifParameter

owner	<b>ProjectModel</b>
-------	---------------------

### Operation ProjectModel::getAllPictures

owner	<b>ProjectModel</b>
parameter	name direction type multiplicity default <b>return return Picture *</b>
ownedMember	<b>return</b>
documentation	<p>Get all pictures which the model handle with.</p> <p>@return an amount of pictures.</p>

### Operation ProjectModel::getDirectoriesOfAPictureSet

owner	<b>ProjectModel</b>
parameter	name direction type multiplicity default <b>pictureSet in PictureSet return return Directory *</b>
ownedMember	<b>pictureSet return</b>
documentation	<p>Get all directories which a picture set handle with.</p> <p>@param pictureSet the picture set which we use as root.</p> <p>@return an amount of picture sets.</p>

### Operation ProjectModel::getExifParameter

owner	<b>ProjectModel</b>
parameter	name direction type multiplicity default <b>return return Object *;*</b>
ownedMember	<b>return</b>
documentation	<p>Get the exif parameters which knipsX can handle with.</p> <p>@return the parameters.</p>

### Operation ProjectModel::getInstance

owner	<b>ProjectModel</b>
parameter	name direction type multiplicity default <b>projectEntry in ProjectEntry pictureSets in List&lt;E-&gt;PictureSet&gt; reports in List&lt;E-&gt;AbstractReportModel&gt; return return ProjectModel</b>
ownedMember	<b>pictureSets projectEntry reports return</b>
documentation	<p>Returns a new instance of ProjectModel, if the project which is committed to the method is != the former project which was managed by the ProjectModel. If the to projects are equals, the no new instance will be returned.</p> <p>@param projectEntry the project which the ProjectModel have to manage.</p> <p>@param pictureSets an amount of picture sets for the committed project which have to be managed by the ProjectModel.</p> <p>@param reports an amount of reports for the committed project which have to be managed by the ProjectModel.</p> <p>@return an instance of ProjectModel which manages the project.</p>

### Operation ProjectModel::getPictureSets

owner	<b>ProjectModel</b>
parameter	name direction type multiplicity default <b>return return Object *</b>
ownedMember	<b>return</b>
documentation	<p>Get all picture sets which the model handle with.</p> <p>@return an amount of picture sets.</p>

### Operation ProjectModel::getPictureSetsOfAPictureSet

owner	<b>ProjectModel</b>
parameter	name direction type multiplicity default <b>pictureSet in PictureSet return return PictureSet *</b>
ownedMember	<b>pictureSet return</b>
documentation	<p>Get all picture sets which a picture set handle with.</p> <p>@param pictureSet the picture set which we use as root.</p> <p>@return an amount of picture sets.</p>

### Operation ProjectModel::getPicturesOfADirectory

owner	<b>ProjectModel</b>
parameter	name direction type multiplicity default <b>directory in Directory return return Picture *</b>
ownedMember	<b>directory return</b>
documentation	<p>Get all pictures which a directory handle with.</p> <p>@param directory the directory which we use as root.</p> <p>@return an amount of pictures.</p>

### Operation ProjectModel::getPicturesOfAPictureSet

owner	<b>ProjectModel</b>
parameter	name direction type multiplicity default <b>pictureSet in PictureSet return return Picture *</b>
ownedMember	<b>pictureSet return</b>
documentation	<p>Get all pictures which a picture set handle with.</p> <p>@param pictureSet the picture set which we use as root.</p> <p>@return an amount of pictures.</p>

#### Operation **ProjectModel::getReports**

owner	<b>ProjectModel</b>
parameter	name direction type multiplicity default <b>return return Object *</b>
ownedMember	<b>return</b>
documentation	Get all reports which the model handle with. @return an amount of picture sets.

#### Property **ProjectModel::lastModel**

owner	<b>ProjectModel</b>
-------	---------------------

#### Property **ProjectModel::pictureSetList**

owner	<b>ProjectModel</b>
-------	---------------------

#### Operation **ProjectModel::ProjectModel**

owner	<b>ProjectModel</b>
parameter	name direction type multiplicity default <b>projectEntry in ProjectEntry pictureSets in List&lt;E&gt;PictureSet reports in List&lt;E&gt;AbstractReportModel&gt;</b>
ownedMember	<b>pictureSets projectEntry reports</b>

#### Operation **ProjectModel::removeContentFromPictureSet**

owner	<b>ProjectModel</b>
parameter	name direction type multiplicity default <b>pictureSet in PictureSet pictureContainer in PictureContainer return return boolean</b>
ownedMember	<b>pictureContainer pictureSet return</b>
documentation	Remove a picture set content of a picture set. @param pictureSet the picture set where the content must be removed. @param pictureContainer the content which must be removed. @return true if the picture set content was removed, false if not.

#### Operation **ProjectModel::removePictureSet**

owner	<b>ProjectModel</b>
parameter	name direction type multiplicity default <b>pictureSet in PictureSet return return boolean</b>
ownedMember	<b>pictureSet return</b>
documentation	Remove a picture set of the current project. @param pictureSet the picture set to remove.

	@return true if the picture set was removed, false if not.
--	--

### Operation ProjectModel::removeReport

owner	<a href="#">ProjectModel</a>
parameter	name direction type multiplicity default <b>report in AbstractReportModel</b> return return boolean
ownedMember	<b>report return</b>
documentation	<p>Remove a report of the current project.</p> <p>@param report the picture set to remove.</p> <p>@return true if the report was removed, false if not.</p>

### Property ProjectModel::reportList

owner	<a href="#">ProjectModel</a>
-------	------------------------------

### Operation ProjectModel::saveProjectModel

owner	<a href="#">ProjectModel</a>
parameter	name direction type multiplicity default <b>return return void</b>
ownedMember	<b>return</b>

## Class ProjectOpenController

diagram	<p>The diagram shows the class <b>ProjectOpenController</b> with a dependency on <b>ProjectManagementModel</b> and <b>JProjectManagement</b>. It has two constructors: <code>&lt;&lt;constructor&gt;&gt; ProjectOpenController(in model:M, in view:V)</code> and <code>&lt;&lt;annotations&gt;&gt; actionPerformed(in e:ActionEvent):void</code>. An operation <b>actionPerformed()</b> is also defined.</p>
hierarchy	<p>The hierarchy diagram shows <b>ProjectOpenController</b> inheriting from <b>AbstractController&lt;M-&gt;M,V-&gt;V&gt;</b>.</p>
owner	<b>projectmanagement</b>
template parameters	name kind constrainingClassifier default M Class <b>ProjectManagementModel</b> V Class <b>JProjectManagement&lt;M-&gt;M&gt;</b>
ownedMember	<b>actionPerformed ProjectOpenController</b>
general	<b>AbstractController&lt;M-&gt;M,V-&gt;V&gt;</b>
target of relation	ComponentRealization <b>Implementierung</b>
documentation	<p>Represents the actions which are done by pushing the project open button.</p> <p>Acts in harmony with JProjectManagement.</p>

## Operation ProjectOpenController::actionPerformed

owner	<b>ProjectOpenController</b>
parameter	name direction type multiplicity default <b>e in ActionEvent return return void</b>
ownedMember	<b>e return</b>

## Operation ProjectOpenController::ProjectOpenController

owner	<b>ProjectOpenController</b>
parameter	name direction type multiplicity default <b>model in M view in V</b>
ownedMember	<b>model view</b>

## Class ProjectSaveController

diagram	<p>ProjectSaveController</p> <p><code>&lt;&lt;constructor&gt;&gt; ProjectSaveController(in model:M, in view:V)</code></p> <p><code>&lt;&lt;annotations&gt;&gt; actionPerformed(in e:ActionEvent):void</code></p>	<b>actionPerformed() (Operation)</b> <code>&lt;&lt;annotations&gt;&gt; modifiers = Override</code>
hierarchy	<pre> classDiagram     class AbstractController {         &lt;&lt;M-&gt;&gt;M, &lt;&lt;V-&gt;&gt;V     }     class ProjectSaveController {         &lt;&lt;M-&gt;&gt;M, &lt;&lt;V-&gt;&gt;V     }     ProjectSaveController &lt; -- AbstractController   </pre>	
owner	<a href="#">projectview</a>	
template parameters	name kind constrainingClassifier default M Class <a href="#">ProjectModel</a> V Class <a href="#">JProjectView&lt;M-&gt;M&gt;</a>	
ownedMember	<a href="#">actionPerformed ProjectSaveController</a>	
general	<a href="#">AbstractController&lt;M-&gt;M,V-&gt;V&gt;</a>	ComponentRealization <a href="#">Implementierung</a>
target of relation		
documentation	<p>Represents the Actions which are done by clicking save project.</p> <p>Acts in harmony with a JProjectView.</p>	

## Operation ProjectSaveController::actionPerformed

owner	<a href="#">ProjectSaveController</a>
parameter	name direction type multiplicity default <code>e in ActionEvent return return void</code>
ownedMember	<code>e return</code>

## Operation ProjectSaveController::ProjectSaveController

owner	<a href="#">ProjectSaveController</a>
parameter	name direction type multiplicity default <code>model in M view in V</code>
ownedMember	<code>model view</code>

## Class ProjectSwitchController

diagram	<p>The diagram shows the class <b>ProjectSwitchController</b> with its template parameters <b>M&gt;ProjectModel, V&gt;JProjectView&lt;M-&gt;M&gt;</b>. It has two constructor associations: <b>&lt;&lt;constructor&gt;&gt; ProjectSwitchController(in model:M, in view:V)</b> and <b>&lt;&lt;annotations&gt;&gt; actionPerformed(in e:ActionEvent):void</b>. A note indicates that <b>actionPerformed()</b> is an operation with annotations and modifiers = Override.</p>
hierarchy	<pre> classDiagram     class AbstractController {         &lt;&lt;M-&gt;M, V-&gt;V&gt;&gt;     }     class ProjectSwitchController {         &lt;&lt;AbstractController&gt;&gt;     }     ProjectSwitchController --&gt; AbstractController   </pre> <p>The hierarchy diagram shows <b>ProjectSwitchController</b> inheriting from <b>AbstractController&lt;M-&gt;M, V-&gt;V&gt;</b>.</p>
owner	<b>projectview</b>
template parameters	name kind constrainingClassifier default <b>M Class ProjectModel V Class JProjectView&lt;M-&gt;M&gt;</b>
ownedMember	<a href="#">actionPerformed ProjectSwitchController</a>
general	<a href="#">AbstractController&lt;M-&gt;M, V-&gt;V&gt;</a>
target of relation	ComponentRealization <a href="#">Implementierung</a>
documentation	<p>Represents the actions which are done by clicking switch project.</p> <p>Acts in harmony with <b>JProjectView</b>.</p>

## Operation ProjectSwitchController::actionPerformed

owner	<a href="#">ProjectSwitchController</a>
parameter	name direction type multiplicity default <b>e in ActionEvent return return void</b>
ownedMember	<b>e return</b>

## Operation ProjectSwitchController::ProjectSwitchController

owner	<a href="#">ProjectSwitchController</a>
parameter	name direction type multiplicity default <b>model in M view in V</b>
ownedMember	<b>model view</b>

## Class ReportAddExifKeywordController

diagram	<pre> classDiagram     class ReportAddExifKeywordController {         &lt;&lt;constructor&gt;&gt; ReportAddExifKeywordController(in view:V)         &lt;&lt;annotations&gt;&gt; actionPerformed(in e:ActionEvent):void     }     JPictureSetExif "M,V-&gt;JPictureSetExif" --&gt; ReportAddExifKeywordController     Note over actionPerformed: actionPerformed() (Operation)     &lt;&lt;annotations&gt;&gt; modifiers = Override   </pre>
hierarchy	<pre> classDiagram     class AbstractController&lt;M-&gt;M,V-&gt;V&gt;     class ReportAddExifKeywordController {         &lt;&lt;constructor&gt;&gt; ReportAddExifKeywordController(in view:V)         &lt;&lt;annotations&gt;&gt; actionPerformed(in e:ActionEvent):void     }     ReportAddExifKeywordController --&gt; AbstractController&lt;M-&gt;M,V-&gt;V&gt;   </pre>
owner	<a href="#">reportmanagement</a>
template parameters	name kind constrainingClassifier default <b>M Class V Class <a href="#">JPictureSetExif</a></b>
ownedMember	<a href="#">actionPerformed ReportAddExifKeywordController</a>
general	<a href="#">AbstractController&lt;M-&gt;M,V-&gt;V&gt;</a>
target of relation	ComponentRealization <a href="#">Implementierung</a>
documentation	<p>This controller is responsible for associating an EXIF keyword with the report which invoked this controller.</p> <p>@author David Kaufman</p> <p>@param &lt;M&gt; @param &lt;V&gt;</p>

## Operation ReportAddExifKeywordController::actionPerformed

owner	<a href="#">ReportAddExifKeywordController</a>
parameter	name direction type multiplicity default <b>e in ActionEvent return return void</b>
ownedMember	<b>e return</b>

## Operation ReportAddExifKeywordController::ReportAddExifKeywordController

owner	<a href="#">ReportAddExifKeywordController</a>
parameter	name direction type multiplicity default <b>view in V</b>
ownedMember	<b>view</b>

## Class ReportAddPictureSetController

diagram	<p>The diagram shows the class <b>ReportAddPictureSetController</b> with its constructor and an operation <b>actionPerformed()</b>. The class is annotated with <b>M, V &gt; JPicutreSetExif</b>. The operation <b>actionPerformed()</b> is annotated with <b>&lt;&lt;annotations&gt;&gt; modifiers = Override</b>. Constructors shown are <b>&lt;&lt;constructor&gt;&gt; ReportAddPictureSetController(in view:V)</b> and <b>&lt;&lt;annotation&gt;&gt; actionPerformed(in e:ActionEvent):void</b>.</p>
hierarchy	<pre> classDiagram     class ReportAddPictureSetController {         &lt;&lt;constructor&gt;&gt; ReportAddPictureSetController(in view:V)         &lt;&lt;annotation&gt;&gt; actionPerformed(in e:ActionEvent):void     }     class AbstractController&lt;M-&gt;M, V-&gt;V&gt;     ReportAddPictureSetController &lt; -- AbstractController   </pre>
owner	<b>reportmanagement</b>
template parameters	name kind constrainingClassifier default <b>M Class</b> <b>V Class</b> <b>JPictureSetExif</b>
ownedMember	<b>actionPerformed ReportAddPictureSetController</b>
general	<b>AbstractController&lt;M-&gt;M, V-&gt;V&gt;</b>
target of relation	ComponentRealization <b>Implementierung</b>
documentation	<p>This controller is responsible for associating one or more PictureSets with the report which invoked this controller.</p> <p>@author David Kaufman</p> <p>@param &lt;M&gt; @param &lt;V&gt;</p>

## Operation ReportAddPictureSetController::actionPerformed

owner	<b>ReportAddPictureSetController</b>
parameter	name direction type multiplicity default <b>e in ActionEvent return return void</b>
ownedMember	<b>e return</b>

## Operation ReportAddPictureSetController::ReportAddPictureSetController

owner	<b>ReportAddPictureSetController</b>
parameter	name direction type multiplicity default <b>view in V</b>
ownedMember	<b>view</b>

## Class ReportCloseController

diagram	<p>The diagram shows the class <b>ReportCloseController</b> with its constructor and an operation <b>actionPerformed()</b>. The class is associated with <b>M</b> and <b>V</b>. The operation <b>actionPerformed()</b> is annotated with <code>&lt;&lt;annotations&gt;&gt;</code> and <code>modifiers = Override</code>.</p> <pre> classDiagram     class ReportCloseController {         &lt;&lt;constructor&gt;&gt; ReportCloseController(in view:V)         &lt;&lt;annotations&gt;&gt; actionPerformed(in e:ActionEvent):void     }     ReportCloseController "1" -- "1" M, V     ReportCloseController "1" -- "1" actionPerformed() (Operation)         &lt;&lt;annotations&gt;&gt; modifiers = Override   </pre>
hierarchy	<p>The hierarchy diagram shows <b>ReportCloseController</b> inheriting from <b>AbstractController&lt;M-&gt;M,V-&gt;V&gt;</b>.</p> <pre> classDiagram     class ReportCloseController {         &lt;&lt;constructor&gt;&gt; ReportCloseController(in view:V)         &lt;&lt;annotations&gt;&gt; actionPerformed(in e:ActionEvent):void     }     ReportCloseController "1" -- "1" AbstractController&lt;M-&gt;M,V-&gt;V&gt;   </pre>
owner	<b>reportmanagement</b>
template parameters	name kind constrainingClassifier default <b>M Class V Class</b>
ownedMember	<b>actionPerformed ReportCloseController</b>
general	<b>AbstractController&lt;M-&gt;M,V-&gt;V&gt;</b>
target of relation	ComponentRealization <b>Implementierung</b>
documentation	<p>This controller manages the closure of the report configuration view.</p> <p>@author David Kaufman</p> <p>@param &lt;M&gt;  @param &lt;V&gt;</p>

## Operation ReportCloseController::actionPerformed

owner	<b>ReportCloseController</b>
parameter	name direction type multiplicity default <b>e in ActionEvent return return void</b>
ownedMember	<b>e return</b>

## Operation ReportCloseController::ReportCloseController

owner	<b>ReportCloseController</b>
parameter	name direction type multiplicity default <b>view in V</b>
ownedMember	<b>view</b>

## Class ReportCreateController

diagram	<p>ReportCreateController</p> <p><code>&lt;&lt;constructor&gt;&gt; ReportCreateController(<b>in</b> model:<b>M</b>, <b>in</b> view:<b>V</b>)</code></p> <p><code>&lt;&lt;annotations&gt;&gt; actionPerformed(<b>in</b> e:<b>ActionEvent</b>):<b>void</b></code></p>	<b>actionPerformed() (Operation)</b> <code>&lt;&lt;annotations&gt;&gt; modifiers = Override</code>
hierarchy	<pre> classDiagram     class ReportCreateController {         &lt;&lt;constructor&gt;&gt; ReportCreateController(in model:M, in view:V)         &lt;&lt;annotations&gt;&gt; actionPerformed(in e:ActionEvent):void     }     class AbstractController {         &lt;&lt;constructor&gt;&gt; AbstractController(M-&gt;M, V-&gt;V)     }     ReportCreateController &lt; -- AbstractController   </pre>	
owner	<a href="#">projectview</a>	
template parameters		name kind constrainingClassifier default <b>M</b> Class <a href="#">ProjectModel</a> <b>V</b> Class <a href="#">JProjectView&lt;M-&gt;M&gt;</a>
ownedMember	<a href="#">actionPerformed ReportCreateController</a>	
general	<a href="#">AbstractController&lt;M-&gt;M,V-&gt;V&gt;</a>	ComponentRealization <a href="#">Implementierung</a>
target of relation		
documentation	Represents the Actions which are done by clicking on create report. Acts in harmony with JProjectView.	

## Operation ReportCreateController::actionPerformed

owner	<a href="#">ReportCreateController</a>
parameter	name direction type multiplicity default <b>e</b> in <b>ActionEvent</b> return return void
ownedMember	<b>e</b> return

## Operation ReportCreateController::ReportCreateController

owner	<a href="#">ReportCreateController</a>
parameter	name direction type multiplicity default <b>model</b> in <b>M</b> <b>view</b> in <b>V</b>
ownedMember	<b>model</b> <b>view</b>

## Class ReportDeleteController

diagram	<p>The diagram shows the class <b>ReportDeleteController</b> with its template parameters <b>M&gt;ProjectModel,V&gt;JProjectView&lt;M-&gt;M&gt;</b>. It has two constructor annotations: <code>&lt;&lt;constructor&gt;&gt; ReportDeleteController(in model:M, in view:V)</code> and <code>&lt;&lt;annotations&gt;&gt; actionPerformed(in e:ActionEvent):void</code>. A note indicates that <code>actionPerformed()</code> is an override operation.</p>
hierarchy	<p>The hierarchy diagram shows <b>ReportDeleteController</b> inheriting from <b>AbstractController&lt;M-&gt;M,V-&gt;V&gt;</b>.</p>
owner	<b>projectview</b>
template parameters	name kind constrainingClassifier default <b>M Class ProjectModel V Class JProjectView&lt;M-&gt;M&gt;</b>
ownedMember	<b>actionPerformed ReportDeleteController</b>
general	<b>AbstractController&lt;M-&gt;M,V-&gt;V&gt;</b>
target of relation	ComponentRealization <b>Implementierung</b>
documentation	Represents the Actions which are done by clicking on the delete report. Acts in harmony with a JProjectview.

## Operation ReportDeleteController::actionPerformed

owner	<b>ReportDeleteController</b>
parameter	name direction type multiplicity default <b>e in ActionEvent return return void</b>
ownedMember	<b>e return</b>

## Operation ReportDeleteController::ReportDeleteController

owner	<b>ReportDeleteController</b>
parameter	name direction type multiplicity default <b>model in M view in V</b>
ownedMember	<b>model view</b>

## Class ReportEditController

diagram	<pre> classDiagram     class ReportEditController {         &lt;&lt;constructor&gt;&gt; ReportEditController(in view:V)         &lt;&lt;annotations&gt;&gt; actionPerformed(in e:ActionEvent):void     }     class AbstractController {         &lt;&lt;constructor&gt;&gt; AbstractController&lt;M-&gt;M, V-&gt;V&gt;     }     ReportEditController &lt; -- AbstractController     </pre>
hierarchy	<pre> classDiagram     class AbstractController {         &lt;&lt;constructor&gt;&gt; AbstractController&lt;M-&gt;M, V-&gt;V&gt;     }     class ReportEditController {         &lt;&lt;constructor&gt;&gt; ReportEditController(in view:V)         &lt;&lt;annotations&gt;&gt; actionPerformed(in e:ActionEvent):void     }     ReportEditController &lt; -- AbstractController     </pre>
owner	<a href="#">diagrams</a>
template parameters	name kind constrainingClassifier default <b>M Class V Class JAbstract3DView&lt;M-&gt;?&gt;</b>
ownedMember	<a href="#">actionPerformed ReportEditController</a>
general	<a href="#">AbstractController&lt;M-&gt;M, V-&gt;V&gt;</a>
target of relation	ComponentRealization <a href="#">Implementierung</a>

## Operation ReportEditController::actionPerformed

owner	<a href="#">ReportEditController</a>
parameter	name direction type multiplicity default <b>e in ActionEvent return return void</b>
ownedMember	<b>e return</b>

## Operation ReportEditController::ReportEditController

owner	<a href="#">ReportEditController</a>
parameter	name direction type multiplicity default <b>view in V</b>
ownedMember	<b>view</b>

## Enumeration ReportHelper

diagram	<pre>&lt;&lt;enumeration&gt;&gt; ReportHelper</pre> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 5px;"><code>currentReport:ReportHelper</code></td></tr> <tr> <td style="padding: 5px;"><code>defaultReport:ReportHelper=ReportHelper_Boxplot</code></td></tr> <tr> <td style="padding: 5px;"><code>currentReportUtil:JAbstractReportUtil&lt;M-&gt;AbstractReportModel,V-&gt;AbstractReportCompilation&lt;M-&gt;AbstractReportModel&gt;</code></td></tr> <tr> <td style="padding: 5px;"><code>currentModel:AbstractReportModel</code></td></tr> <tr> <td style="padding: 5px;"><code>currentProjectModel:ProjectModel</code></td></tr> </table> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 5px;"><code>createReportCompilation():AbstractReportCompilation&lt;M-&gt;?&gt;</code></td></tr> <tr> <td style="padding: 5px;"><code>createReportModel():AbstractReportModel</code></td></tr> <tr> <td style="padding: 5px;"><code>getDiagram(<i>in</i> model:AbstractReportModel):JAbstractDiagram&lt;M-&gt;AbstractReportModel&gt;</code></td></tr> <tr> <td style="padding: 5px;"><code>getNumberOfAxes():int</code></td></tr> <tr> <td style="padding: 5px;"><code>updateReport(<i>in</i> myreport:ReportHelper):void</code></td></tr> <tr> <td style="padding: 5px;"><code>addCurrentModelToReportList():void</code></td></tr> <tr> <td style="padding: 5px;"><code>cleanUp():void</code></td></tr> </table> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 5px;">Boxplot</td></tr> <tr> <td style="padding: 5px;">Histogram2D</td></tr> <tr> <td style="padding: 5px;">Histogram3D</td></tr> <tr> <td style="padding: 5px;">Cluster3D</td></tr> <tr> <td style="padding: 5px;">Table</td></tr> </table>	<code>currentReport:ReportHelper</code>	<code>defaultReport:ReportHelper=ReportHelper_Boxplot</code>	<code>currentReportUtil:JAbstractReportUtil&lt;M-&gt;AbstractReportModel,V-&gt;AbstractReportCompilation&lt;M-&gt;AbstractReportModel&gt;</code>	<code>currentModel:AbstractReportModel</code>	<code>currentProjectModel:ProjectModel</code>	<code>createReportCompilation():AbstractReportCompilation&lt;M-&gt;?&gt;</code>	<code>createReportModel():AbstractReportModel</code>	<code>getDiagram(<i>in</i> model:AbstractReportModel):JAbstractDiagram&lt;M-&gt;AbstractReportModel&gt;</code>	<code>getNumberOfAxes():int</code>	<code>updateReport(<i>in</i> myreport:ReportHelper):void</code>	<code>addCurrentModelToReportList():void</code>	<code>cleanUp():void</code>	Boxplot	Histogram2D	Histogram3D	Cluster3D	Table
<code>currentReport:ReportHelper</code>																		
<code>defaultReport:ReportHelper=ReportHelper_Boxplot</code>																		
<code>currentReportUtil:JAbstractReportUtil&lt;M-&gt;AbstractReportModel,V-&gt;AbstractReportCompilation&lt;M-&gt;AbstractReportModel&gt;</code>																		
<code>currentModel:AbstractReportModel</code>																		
<code>currentProjectModel:ProjectModel</code>																		
<code>createReportCompilation():AbstractReportCompilation&lt;M-&gt;?&gt;</code>																		
<code>createReportModel():AbstractReportModel</code>																		
<code>getDiagram(<i>in</i> model:AbstractReportModel):JAbstractDiagram&lt;M-&gt;AbstractReportModel&gt;</code>																		
<code>getNumberOfAxes():int</code>																		
<code>updateReport(<i>in</i> myreport:ReportHelper):void</code>																		
<code>addCurrentModelToReportList():void</code>																		
<code>cleanUp():void</code>																		
Boxplot																		
Histogram2D																		
Histogram3D																		
Cluster3D																		
Table																		
owner	<code>reportmanagement</code>																	
ownedMember	<code>addCurrentModelToReportList Boxplot cleanUp Cluster3D createReportCompilation createReportModel currentModel currentProjectModel currentReport currentReportUtil defaultReport getDiagram getNumberOfAxes Histogram2D Histogram3D Table updateReport</code>																	
target of relation	ComponentRealization <b>Implementierung</b>																	
typedElements	Enumeration <b>ReportHelper</b> Property <code>currentReport defaultReport</code> Operation <code>updateReport</code>																	
documentation	<p>This class is the mediator between the report configuration utility and the model it creates.</p> <p>It is responsible for saving the currently active report and also for refreshing the specified configuration utility.</p> <p>@author David Kaufman</p>																	

## Operation ReportHelper::addCurrentModelToReportList

owner	<b>ReportHelper</b>
parameter	name direction type multiplicity default <b>return return void</b>
ownedMember	<b>return</b>
documentation	Adds the currently active model in the ReportHelper class into the ProjectModel

## EnumerationLiteral ReportHelper::Boxplot

owner	<b>ReportHelper</b>
-------	---------------------

#### Operation ReportHelper::Boxplot::createReportCompilation

owner	<a href="#">Boxplot</a>
parameter	name direction type multiplicity default <b>return return AbstractReportCompilation&lt;M-&gt;BoxplotModel&gt;</b>
ownedMember	<b>return</b>

#### Operation ReportHelper::Boxplot::createReportModel

owner	<a href="#">Boxplot</a>
parameter	name direction type multiplicity default <b>return return AbstractReportModel</b>
ownedMember	<b>return</b>

#### Operation ReportHelper::Boxplot::getDiagram

owner	<a href="#">Boxplot</a>
parameter	name direction type multiplicity default <b>model in AbstractReportModel return return JAbstractDiagram&lt;M-&gt;AbstractReportModel&gt;</b>
ownedMember	<b>model return</b>

#### Operation ReportHelper::Boxplot::getNumberOfAxes

owner	<a href="#">Boxplot</a>
parameter	name direction type multiplicity default <b>return return int</b>
ownedMember	<b>return</b>

#### Operation ReportHelper::cleanUp

owner	<a href="#">ReportHelper</a>
parameter	name direction type multiplicity default <b>return return void</b>
ownedMember	<b>return</b>
documentation	This function resets all settings so that a new report can be generated properly. Basically initializes all of the variables with their respective default values

#### EnumerationLiteral ReportHelper::Cluster3D

owner	<a href="#">ReportHelper</a>
-------	------------------------------

#### Operation ReportHelper::Cluster3D::createReportCompilation

owner	<a href="#">Cluster3D</a>
parameter	name direction type multiplicity default <b>return return AbstractReportCompilation&lt;M-&gt;Cluster3DModel&gt;</b>
ownedMember	<b>return</b>

#### Operation ReportHelper::Cluster3D::createReportModel

owner	<a href="#">Cluster3D</a>
parameter	name direction type multiplicity default <b>return return AbstractReportModel</b>
ownedMember	<b>return</b>

#### Operation ReportHelper::Cluster3D::getDiagram

owner	<a href="#">Cluster3D</a>
parameter	name direction type multiplicity default <b>model in AbstractReportModel return return JAbstractDiagram&lt;M-&gt;AbstractReportModel&gt;</b>
ownedMember	<b>model return</b>

#### Operation ReportHelper::Cluster3D::getNumberOfAxes

owner	<a href="#">Cluster3D</a>
parameter	name direction type multiplicity default <b>return return int</b>
ownedMember	<b>return</b>

#### Operation ReportHelper::createReportCompilation

owner	<a href="#">ReportHelper</a>
parameter	name direction type multiplicity default <b>return return AbstractReportCompilation&lt;M-&gt;?&gt;</b>
ownedMember	<b>return</b>
documentation	Returns the report type associated with the specified report enum @return the report type associated with the specified report enum

#### Operation ReportHelper::createReportModel

owner	<a href="#">ReportHelper</a>
parameter	name direction type multiplicity default <b>return return AbstractReportModel</b>
ownedMember	<b>return</b>
documentation	Creates and returns the associated reportmodel with the specified report enum @return the associated reportmodel with the specified report enum

#### Property ReportHelper::currentModel

owner	<a href="#">ReportHelper</a>
documentation	The current model of the current report configuration run

### Property ReportHelper::currentProjectModel

owner	<a href="#">ReportHelper</a>
documentation	The current reference of the ProjectModel. This makes it fairly easy to add new reports to the project model.

### Property ReportHelper::currentReport

owner	<a href="#">ReportHelper</a>
documentation	The current report of the current report configuration run

### Property ReportHelper::currentReportUtil

owner	<a href="#">ReportHelper</a>
documentation	The current configuration utility of the current report configuration run

### Property ReportHelper::defaultReport

owner	<a href="#">ReportHelper</a>
documentation	The default report which is selected automatically when you start the Wizard

### Operation ReportHelper::getDiagram

owner	<a href="#">ReportHelper</a>
parameter	name direction type multiplicity default <b>model</b> in <a href="#">AbstractReportModel</a> return return <a href="#">JAbstractDiagram&lt;M-&gt;AbstractReportModel&gt;</a>
ownedMember	<b>model return</b>
documentation	Returns the diagram of the associated report

### Operation ReportHelper::getNumberOfAxes

owner	<a href="#">ReportHelper</a>
parameter	name direction type multiplicity default <b>return return int</b>
ownedMember	<b>return</b>
documentation	Returns the number of axes each diagram type uses @return the number of axes each diagram type uses   returns 0 if no axis is used

### EnumerationLiteral ReportHelper::Histogram2D

owner	<a href="#">ReportHelper</a>
-------	------------------------------

**Operation ReportHelper::Histogram2D::createReportCompilation**

owner	<a href="#">Histogram2D</a>
parameter	name direction type multiplicity default <b>return return AbstractReportCompilation&lt;M-&gt;Histogram2DModel&gt;</b>
ownedMember	<b>return</b>

**Operation ReportHelper::Histogram2D::createReportModel**

owner	<a href="#">Histogram2D</a>
parameter	name direction type multiplicity default <b>return return AbstractReportModel</b>
ownedMember	<b>return</b>

**Operation ReportHelper::Histogram2D::getDiagram**

owner	<a href="#">Histogram2D</a>
parameter	name direction type multiplicity default <b>model in AbstractReportModel return return JAbstractDiagram&lt;M-&gt;AbstractReportModel&gt;</b>
ownedMember	<b>model return</b>

**Operation ReportHelper::Histogram2D::getNumberOfAxes**

owner	<a href="#">Histogram2D</a>
parameter	name direction type multiplicity default <b>return return int</b>
ownedMember	<b>return</b>

**EnumerationLiteral ReportHelper::Histogram3D**

owner	<a href="#">ReportHelper</a>
-------	------------------------------

**Operation ReportHelper::Histogram3D::createReportCompilation**

owner	<a href="#">Histogram3D</a>
parameter	name direction type multiplicity default <b>return return AbstractReportCompilation&lt;M-&gt;Histogram3DModel&gt;</b>
ownedMember	<b>return</b>

**Operation ReportHelper::Histogram3D::createReportModel**

owner	<a href="#">Histogram3D</a>
parameter	name direction type multiplicity default <b>return return AbstractReportModel</b>
ownedMember	<b>return</b>

**Operation ReportHelper::Histogram3D::getDiagram**

owner	<a href="#">Histogram3D</a>
parameter	name direction type multiplicity default <b>model in AbstractReportModel return return JAbstractDiagram&lt;M-&gt;AbstractReportModel&gt;</b>

ownedMember	<b>model return</b>
-------------	---------------------

#### Operation **ReportHelper::Histogram3D::getNumberOfAxes**

owner	<a href="#">Histogram3D</a>
parameter	name direction type multiplicity default <b>return return int</b>
ownedMember	<b>return</b>

#### EnumerationLiteral **ReportHelper::Table**

owner	<a href="#">ReportHelper</a>
-------	------------------------------

#### Operation **ReportHelper::Table::createReportCompilation**

owner	<a href="#">Table</a>
parameter	name direction type multiplicity default <b>return return AbstractReportCompilation&lt;M&gt;AbstractReportModel&gt;</b>
ownedMember	<b>return</b>

#### Operation **ReportHelper::Table::createReportModel**

owner	<a href="#">Table</a>
parameter	name direction type multiplicity default <b>return return AbstractReportModel</b>
ownedMember	<b>return</b>

#### Operation **ReportHelper::Table::getDiagram**

owner	<a href="#">Table</a>
parameter	name direction type multiplicity default <b>model in AbstractReportModel return return JAbstractDiagram&lt;M-&gt;AbstractReportModel&gt;</b>
ownedMember	<b>model return</b>

#### Operation **ReportHelper::Table::getNumberOfAxes**

owner	<a href="#">Table</a>
parameter	name direction type multiplicity default <b>return return int</b>
ownedMember	<b>return</b>

#### Operation **ReportHelper::updateReport**

owner	<a href="#">ReportHelper</a>
parameter	name direction type multiplicity default <b>myreport in ReportHelper return return void</b>
ownedMember	<b>myreport return</b>
documentation	Sets the current report and updates the configuration utility Use this method if you want explicitly update the configuration utility

	@param myreport	
--	-----------------	--

## Class ReportOpenController

diagram	<p>The diagram shows the class <b>ReportOpenController</b> with its constructor and the overridden operation <b>actionPerformed()</b>. The class is constrained by the template parameters <b>M&gt;ProjectModel,V&gt;JProjectView&lt;M-&gt;M&gt;</b>.</p>
hierarchy	<p>The hierarchy diagram shows <b>ReportOpenController</b> inheriting from <b>AbstractController&lt;M-&gt;M,V-&gt;V&gt;</b>.</p>
owner	<b>projectview</b>
template parameters	name kind constrainingClassifier default <b>M Class ProjectModel V Class JProjectView&lt;M-&gt;M&gt;</b>
ownedMember	<b>actionPerformed ReportOpenController</b>
general	<b>AbstractController&lt;M-&gt;M,V-&gt;V&gt;</b>
target of relation	ComponentRealization <b>Implementierung</b>
documentation	Represents the Actions which are done by klicking on open report. Acts in harmony with a JProjectView.

## Operation ReportOpenController::actionPerformed

owner	<b>ReportOpenController</b>
parameter	name direction type multiplicity default <b>e in ActionEvent return return void</b>
ownedMember	<b>e return</b>

## Operation ReportOpenController::ReportOpenController

owner	<b>ReportOpenController</b>
parameter	name direction type multiplicity default <b>model in M view in V</b>
ownedMember	<b>model view</b>

## Class ReportPictureSetRemoveController

diagram	<p>The diagram shows the class <b>ReportPictureSetRemoveController</b> with its constructor and an operation <b>actionPerformed()</b>. The class is annotated with <b>JPictureSetExif</b>. The operation <b>actionPerformed()</b> is annotated with <b>modifiers = Override</b>.</p> <pre> classDiagram     class ReportPictureSetRemoveController {         &lt;&lt;constructor&gt;&gt; ReportPictureSetRemoveController(in view:V)         &lt;&lt;annotations&gt;&gt; actionPerformed(in e:ActionEvent):void     }     ReportPictureSetRemoveController &lt; -- AbstractController&lt;M-&gt;M,V-&gt;V&gt;     JPictureSetExif &lt; -- ReportPictureSetRemoveController   </pre>
hierarchy	<p>The hierarchy diagram shows <b>ReportPictureSetRemoveController</b> inheriting from <b>AbstractController&lt;M-&gt;M,V-&gt;V&gt;</b>.</p>
owner	<b>reportmanagement</b>
template parameters	name kind constrainingClassifier default <b>M Class V Class JPictureSetExif</b>
ownedMember	<b>actionPerformed ReportPictureSetRemoveController</b>
general	<b>AbstractController&lt;M-&gt;M,V-&gt;V&gt;</b>
target of relation	ComponentRealization <b>Implementierung</b>
documentation	<p>This controller is responsible for removing one or more PictureSets of the report which invoked this controller.</p> <p>@author David Kaufman</p> <p>@param &lt;M&gt; @param &lt;V&gt;</p>

## Operation ReportPictureSetRemoveController::actionPerformed

owner	<b>ReportPictureSetRemoveController</b>
parameter	name direction type multiplicity default <b>e in ActionEvent return return void</b>
ownedMember	<b>e return</b>

## Operation ReportPictureSetRemoveController::ReportPictureSetRemoveController

owner	<b>ReportPictureSetRemoveController</b>
parameter	name direction type multiplicity default <b>view in V</b>
ownedMember	<b>view</b>

## Class ReportRemoveExifKeywordController

diagram	<p>The diagram shows the class <b>ReportRemoveExifKeywordController</b> with its constructor and an operation <b>actionPerformed()</b>. The class is associated with a view <b>JPictureSetExif</b>.</p> <pre> classDiagram     class ReportRemoveExifKeywordController {         &lt;&lt;constructor&gt;&gt; ReportRemoveExifKeywordController(in view:V)         &lt;&lt;annotations&gt;&gt; actionPerformed(in e:ActionEvent):void     }     JPictureSetExif "M,V&gt;&gt;" --&gt; ReportRemoveExifKeywordController     ReportRemoveExifKeywordController --&gt; actionPerformed : Operation     actionPerformed &lt;&lt;annotations&gt;&gt; modifiers = Override   </pre>
hierarchy	<p>The hierarchy diagram shows <b>ReportRemoveExifKeywordController</b> inheriting from <b>AbstractController&lt;M-&gt;M,V-&gt;V&gt;</b>.</p> <pre> classDiagram     class AbstractController&lt;M-&gt;M,V-&gt;V&gt;     class ReportRemoveExifKeywordController {         &lt;&lt;constructor&gt;&gt; ReportRemoveExifKeywordController(in view:V)         &lt;&lt;annotations&gt;&gt; actionPerformed(in e:ActionEvent):void     }     ReportRemoveExifKeywordController --&gt; AbstractController   </pre>
owner	<b>reportmanagement</b>
template parameters	name kind constrainingClassifier default <b>M Class V Class JPictureSetExif</b>
ownedMember	<b>actionPerformed ReportRemoveExifKeywordController</b>
general	<b>AbstractController&lt;M-&gt;M,V-&gt;V&gt;</b>
target of relation	ComponentRealization <b>Implementierung</b>
documentation	<p>This controller is responsible for removing an EXIF keyword of the report which invoked this controller.</p> <p>@author David Kaufman</p> <p>@param &lt;M&gt; @param &lt;V&gt;</p>

## Operation ReportRemoveExifKeywordController::actionPerformed

owner	<b>ReportRemoveExifKeywordController</b>
parameter	name direction type multiplicity default <b>e in ActionEvent return return void</b>
ownedMember	<b>e return</b>

## Operation ReportRemoveExifKeywordController::ReportRemoveExifKeywordController

owner	<b>ReportRemoveExifKeywordController</b>
parameter	name direction type multiplicity default <b>view in V</b>
ownedMember	<b>view</b>

## Class ReportSaveController

diagram	<pre> classDiagram     class ReportSaveController {         &lt;&lt;M&gt;&gt;AbstractReportModel &lt;&lt;V&gt;&gt;AbstractReportCompilation &lt;&lt;M&gt;&gt;AbstractReportModel         showDiagram:boolean         &lt;&lt;constructor&gt;&gt; ReportSaveController(in model:M, in view:V, in showDiagram:boolean)         &lt;&lt;annotations&gt;&gt; actionPerformed(in e:ActionEvent):void     } </pre>
hierarchy	<pre> classDiagram     class AbstractController&lt;M&gt;M,V&gt;V&gt;     class ReportSaveController {         &lt;&lt;AbstractController&gt;&gt;     } </pre>
owner	<b>reportmanagement</b>
template parameters	name kind constrainingClassifier default M Class <a href="#">AbstractReportModel</a> V Class <a href="#">AbstractReportCompilation&lt;M-&gt;AbstractReportModel&gt;</a>
ownedMember	<a href="#">actionPerformed ReportSaveController showDiagram</a>
general	<a href="#">AbstractController&lt;M&gt;M,V&gt;V&gt;</a>
target of relation	ComponentRealization <a href="#">Implementierung</a>
documentation	<p>This controller is responsible for saving a report configuration into its corresponding model and returning that model to the ProjectModel.</p> <p>@author David Kaufman</p> <p>@param &lt;M&gt; @param &lt;V&gt;</p>

## Operation ReportSaveController::actionPerformed

owner	<a href="#">ReportSaveController</a>
parameter	name direction type multiplicity default e in ActionEvent return return void
ownedMember	e return

## Operation ReportSaveController::ReportSaveController

owner	<a href="#">ReportSaveController</a>
parameter	name direction type multiplicity default model in M view in V showDiagram in boolean
ownedMember	model showDiagram view

## Property ReportSaveController::showDiagram

owner	<a href="#">ReportSaveController</a>
-------	--------------------------------------



## Package Root

ownedMember

[Component View](#) [Java JDK 6 \(types only\)](#) [Java Profile](#) [Package1](#) Unknown External

## Class SelectableCluster3DSphere

diagram	<pre> classDiagram     class SelectableCluster3DSphere {         #frequencypoint: Frequency3DPoint     }     SelectableCluster3DSphere &lt;&lt;constructor&gt;&gt; SelectableCluster3DSphere(in frequencypoint: Frequency3DPoint)     SelectableCluster3DSphere getFrequency3DPoint(): Frequency3DPoint   </pre>
hierarchy	<pre> classDiagram     class Shape3D     class SelectableCluster3DSphere {         #frequencypoint: Frequency3DPoint     }     SelectableCluster3DSphere &lt;&lt;constructor&gt;&gt; SelectableCluster3DSphere(in frequencypoint: Frequency3DPoint)     SelectableCluster3DSphere getFrequency3DPoint(): Frequency3DPoint   </pre>
owner	<a href="#">diagrams</a>
ownedMember	<a href="#">frequencypoint</a> <a href="#">getFrequency3DPoint</a> <a href="#">SelectableCluster3DSphere</a>
general	<a href="#">Shape3D</a>
target of relation	ComponentRealization <a href="#">Implementierung</a>
documentation	<p>This class represents a selectable point in a 3D diagram. Additionally it stores a reference to a Frequency3DPoint object which can be used for evaluation purposes.</p> <p>@author David Kaufman</p> <p>@param &lt;M&gt;</p>

## Property SelectableCluster3DSphere::frequencypoint

owner	<a href="#">SelectableCluster3DSphere</a>
-------	---

## Operation SelectableCluster3DSphere::getFrequency3DPoint

owner	<a href="#">SelectableCluster3DSphere</a>
parameter	name direction type multiplicity default return return <a href="#">Frequency3DPoint</a>
ownedMember	<b>return</b>

## Operation SelectableCluster3DSphere::SelectableCluster3DSphere

owner	<a href="#">SelectableCluster3DSphere</a>
parameter	name direction type multiplicity default <b>frequencypoint</b> in <a href="#">Frequency3DPoint</a>
ownedMember	<b>frequencypoint</b>

## Class StringChecker

diagram	
owner	<a href="#">utils</a>
ownedMember	<a href="#">isStringOk</a>
target of relation	ComponentRealization <a href="#">Implementierung</a>
documentation	Checks strings from view components.

## Operation StringChecker::isStringOk

owner	<a href="#">StringChecker</a>
parameter	name direction type multiplicity default <b>toCheck</b> in String return return boolean
ownedMember	<b>return toCheck</b>
documentation	Checks the string which is committed for validation.  @param toCheck the String to check. @return true if it validates, false if not.

## Class TableConfig

diagram	<pre> classDiagram     class TableConfig {         &lt;&lt;final&gt;&gt; serialVersionUID:long=1L         &lt;&lt;constructor&gt;&gt; TableConfig(in model:M)     }     TableConfig "1" --&gt; "1" TableModel : M&gt;TableModel   </pre>
hierarchy	<pre> classDiagram     class TableConfig {     }     class AbstractReportCompilation {     }     TableConfig &lt; -- AbstractReportCompilation   </pre>
owner	<a href="#">reportmanagement</a>
template parameters	name kind constrainingClassifier default <b>M</b> Class <a href="#">TableModel</a>
ownedMember	<a href="#">serialVersionUID TableConfig</a>
general	<a href="#">AbstractReportCompilation&lt;M&gt;</a>
target of relation	ComponentRealization <a href="#">Implementierung</a>
documentation	<p>This class represents the Table configuration with all its necessary panels.</p> <p>@author David Kaufman</p> <p>@param &lt;M&gt;</p>

## Property TableConfig::serialVersionUID

owner	<a href="#">TableConfig</a>
-------	-----------------------------

## Operation TableConfig::TableConfig

owner	<a href="#">TableConfig</a>
parameter	name direction type multiplicity default <b>model in M</b>
ownedMember	<b>model</b>
documentation	<p>Constructor which initialized the report with all its panels</p> <p>@param tablemodel the model which is used the panels</p>

## Class TableModel

diagram	<pre> classDiagram     class TableModel {         &lt;&lt;constructor&gt;&gt; TableModel(in pictureContainer:ArrayList&lt;E&gt;PictureContainer)         &lt;&lt;constructor&gt;&gt; TableModel()         getPictures():PictureContainer[*]     }     class Observable     class AbstractModel     class AbstractReportModel     TableModel &lt; -- Observable     TableModel &lt; -- AbstractModel     TableModel &lt; -- AbstractReportModel   </pre>
hierarchy	<pre> classDiagram     class Observable     class AbstractModel     class AbstractReportModel     TableModel &lt; -- Observable     TableModel &lt; -- AbstractModel     TableModel &lt; -- AbstractReportModel   </pre>
owner	<a href="#">reportmanagement</a>
ownedMember	<a href="#">getPictures</a> <a href="#">TableModel</a> <a href="#">TableModel</a>
general	<a href="#">AbstractReportModel</a>
target of relation	ComponentRealization <a href="#">Implementierung</a>
documentation	A simple Table of the exif Data. Gives every Picture with all Data. @author Kevin Zuber

## Operation TableModel::getPictures

owner	<a href="#">TableModel</a>
parameter	name direction type multiplicity default <b>return</b> <b>return</b> <a href="#">PictureContainer</a> *
ownedMember	<b>return</b>

## Operation TableModel::TableModel

owner	<a href="#">TableModel</a>
parameter	name direction type multiplicity default <b>pictureContainer</b> in <a href="#">ArrayList&lt;E&gt;PictureContainer</a>
ownedMember	<b>pictureContainer</b>

## Operation TableModel::TableModel

owner	<a href="#">TableModel</a>
-------	----------------------------



## Class TextModel

diagram	<pre> <b>TextModel</b> text:String &lt;&lt;constructor&gt;&gt; TextModel(in pictureContainer:ArrayList&lt;E-&gt;PictureContainer&gt;, in text:String) &lt;&lt;constructor&gt;&gt; TextModel() setText(in text:String):void getText():String </pre>
hierarchy	<pre> classDiagram     class Observable     class AbstractModel     class AbstractReportModel     class TextModel     TextModel &lt; -- AbstractReportModel     AbstractReportModel &lt; -- AbstractModel     Observable &lt; -- AbstractModel </pre>
owner	<a href="#">reportmanagement</a>
ownedMember	<a href="#">getText</a> <a href="#">setText</a> <a href="#">text</a> <a href="#">TextModel</a> <a href="#">TextModel</a>
general	<a href="#">AbstractReportModel</a>
target of relation	ComponentRealization <a href="#">Implementierung</a>
documentation	A simple text model which solely contains text @author David Kaufman

## Operation TextModel::getText

owner	<a href="#">TextModel</a>
parameter	name direction type multiplicity default <b>return return String</b>
ownedMember	<b>return</b>

## Operation TextModel::setText

owner	<a href="#">TextModel</a>
parameter	name direction type multiplicity default <b>text in String return return void</b>
ownedMember	<b>return text</b>

## Property TextModel::text

owner	<a href="#">TextModel</a>
-------	---------------------------

Operation **TextModel::TextModel**

owner	<a href="#">TextModel</a>
parameter	name direction type multiplicity default <b>pictureContainer</b> in <a href="#">ArrayList&lt;E&gt;PictureContainer</a> <b>text</b> in <a href="#">String</a>
ownedMember	<b>pictureContainer</b> <b>text</b>

Operation **TextModel::TextModel**

owner	<a href="#">TextModel</a>
-------	---------------------------

## Class View3DClickController

diagram	<p>View3DClickController</p> <p><b>actionPerformed() (Operation)</b>  <b>&lt;&lt;annotations&gt;&gt;</b> modifiers = Override</p> <p><b>&lt;&lt;constructor&gt;&gt;</b> View3DClickController(<b>in</b> view:<b>V</b>)  <b>&lt;&lt;constructor&gt;&gt;</b> actionPerformed(<b>in</b> e:<b>ActionEvent</b>):void</p>
hierarchy	<pre> classDiagram     class AbstractController {         &lt;&lt;M-&gt;M, V-&gt;V&gt;&gt;     }     class View3DClickController {         &lt;&lt;M-&gt;M, V-&gt;V&gt;&gt;     }     View3DClickController &lt; -- AbstractController   </pre>
owner	<a href="#">diagrams</a>
template parameters	name kind constrainingClassifier default <b>M Class V Class JCluster3D&lt;M-&gt;?</b>
ownedMember	<a href="#">actionPerformed View3DClickController</a>
general	<a href="#">AbstractController&lt;M-&gt;M, V-&gt;V&gt;</a>
target of relation	ComponentRealization <a href="#">Implementierung</a>
documentation	<p>This controller manages the mouse input that occur in a JCluster3D diagram. It listens for input and upon that decides if a SelectableCluster3D Sphere was clicked and outputs EXIF parameters and a thumbnail of picture of one of the selected spheres into the view.</p> <p>@author David Kaufman</p>

## Operation View3DClickController::actionPerformed

owner	<a href="#">View3DClickController</a>
parameter	name direction type multiplicity default <b>e in ActionEvent return return void</b>
ownedMember	<b>e return</b>

## Operation View3DClickController::View3DClickController

owner	<a href="#">View3DClickController</a>
parameter	name direction type multiplicity default <b>view in V</b>
ownedMember	<b>view</b>

## Class View3DCycleController

diagram	<pre> classDiagram     class View3DCycleController {         &lt;&lt;JAbstract3DView&lt;M-&gt;?&gt;&gt;         &lt;&lt;constructor&gt;&gt; View3DCycleController(in view:V)         &lt;&lt;annotations&gt;&gt; actionPerformed(in e:ActionEvent):void     }     class AbstractController {         &lt;&lt;M-&gt;M,V-&gt;V&gt;&gt;     }     View3DCycleController &lt; -- AbstractController   </pre>
hierarchy	<pre> classDiagram     class AbstractController {         &lt;&lt;M-&gt;M,V-&gt;V&gt;&gt;     }     class View3DCycleController {         &lt;&lt;JAbstract3DView&lt;M-&gt;?&gt;&gt;     }     View3DCycleController &lt; -- AbstractController   </pre>
owner	<a href="#">diagrams</a>
template parameters	name kind constrainingClassifier default <b>M Class V Class JAbstract3DView&lt;M-&gt;?&gt;</b>
ownedMember	<a href="#">actionPerformed View3DCycleController</a>
general	<a href="#">AbstractController&lt;M-&gt;M,V-&gt;V&gt;</a>
target of relation	ComponentRealization <a href="#">Implementierung</a>
documentation	<p>This controller makes it possible to cycle through the different 3D perspectives associated with the CyclePerspective enumeration</p> <p>@author David Kaufman</p>

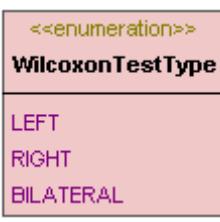
## Operation View3DCycleController::actionPerformed

owner	<a href="#">View3DCycleController</a>
parameter	name direction type multiplicity default <b>e in ActionEvent return return void</b>
ownedMember	<b>e return</b>

## Operation View3DCycleController::View3DCycleController

owner	<a href="#">View3DCycleController</a>
parameter	name direction type multiplicity default <b>view in V</b>
ownedMember	<b>view</b>

### Enumeration **WilcoxonTestType**

diagram	
owner	<b>reportmanagement</b>
ownedMember	<b>BILATERAL</b> <b>LEFT</b> <b>RIGHT</b>
target of relation	ComponentRealization <b>Implementierung</b>
typedElements	Class <b>BoxplotModel</b> Property <b>wilcoxonTestType</b> Operation <b>getWilcoxonTestType</b> <b>setWilcoxonTestType</b> Class <b>JWilcoxon</b> Operation <b>getTestType</b>

### EnumerationLiteral **WilcoxonTestType::BILATERAL**

owner	<b>WilcoxonTestType</b>
-------	-------------------------

### EnumerationLiteral **WilcoxonTestType::LEFT**

owner	<b>WilcoxonTestType</b>
-------	-------------------------

### EnumerationLiteral **WilcoxonTestType::RIGHT**

owner	<b>WilcoxonTestType</b>
-------	-------------------------

## Class WizardCloseController

diagram	<pre> classDiagram     class WizardCloseController {         &lt;&lt;constructor&gt;&gt; WizardCloseController(in view:V)         &lt;&lt;constructor&gt;&gt; actionPerformed(in e:ActionEvent):void         actionPerformed() (Operation)             &lt;&lt;annotations&gt;&gt; modifiers = Override     }     WizardCloseController --&gt; M     WizardCloseController --&gt; V </pre>
hierarchy	<pre> classDiagram     class WizardCloseController {         &lt;&lt;AbstractController&lt;M-&gt;M,V-&gt;V&gt;&gt;     } </pre>
owner	<a href="#">reportmanagement</a>
template parameters	name kind constrainingClassifier default <b>M Class V Class</b>
ownedMember	<a href="#">actionPerformed WizardCloseController</a>
general	<a href="#">AbstractController&lt;M-&gt;M,V-&gt;V&gt;</a>
target of relation	ComponentRealization <a href="#">Implementierung</a>
documentation	<p>This controller manages the closure of the report wizard view.</p> <p>@author David Kaufman</p> <p>@param &lt;M&gt; @param &lt;V&gt;</p>

## Operation WizardCloseController::actionPerformed

owner	<a href="#">WizardCloseController</a>
parameter	name direction type multiplicity default <b>e in ActionEvent return return void</b>
ownedMember	<b>e return</b>

## Operation WizardCloseController::WizardCloseController

owner	<a href="#">WizardCloseController</a>
parameter	name direction type multiplicity default <b>view in V</b>
ownedMember	<b>view</b>

## Class WizardNextPanelController

diagram	<pre> 'M-&gt;AbstractReportModel,V-&gt;JReportWizard&lt;M-&gt;?,V-&gt;?&gt; WizardNextPanelController &lt;&lt;constructor&gt;&gt; WizardNextPanelController(in view:V) &lt;&lt;annotations&gt;&gt; actionPerformed(in e:ActionEvent):void   </pre>	<b>actionPerformed() (Operation)</b> <<annotations>> modifiers = Override
hierarchy	<pre> AbstractController&lt;M-&gt;M,V-&gt;V&gt; WizardNextPanelController   </pre>	
owner	<b>reportmanagement</b>	
template parameters	name kind constrainingClassifier default M Class <a href="#">AbstractReportModel</a> V Class <a href="#">JReportWizard&lt;M-&gt;?,V-&gt;?&gt;</a>	
ownedMember	<a href="#">actionPerformed WizardNextPanelController</a>	
general	<a href="#">AbstractController&lt;M-&gt;M,V-&gt;V&gt;</a>	ComponentRealization <a href="#">Implementierung</a>
target of relation		
documentation	<p>This controller is responsible for managing the wizard navigation to the following pane.</p> <p>@author David Kaufman</p> <p>@param &lt;M&gt;</p> <p>@param &lt;V&gt;</p>	

## Operation WizardNextPanelController::actionPerformed

owner	<a href="#">WizardNextPanelController</a>
parameter	name direction type multiplicity default <b>e</b> in ActionEvent return return void
ownedMember	<b>e</b> return

## Operation WizardNextPanelController::WizardNextPanelController

owner	<a href="#">WizardNextPanelController</a>
parameter	name direction type multiplicity default <b>view</b> in V
ownedMember	<b>view</b>

## Class WizardPreviousPanelController

diagram	<pre> !M&gt;AbstractReportModel,V&gt;JReportWizard&lt;M-&gt;?,V-&gt;?&gt; WizardPreviousPanelController actionPerformed() (Operation) &lt;&lt;annotations&gt;&gt; modifiers = Override &lt;&lt;constructor&gt;&gt; WizardPreviousPanelController(in view:V) &lt;&lt;annotations&gt;&gt; actionPerformed(in e:ActionEvent):void </pre>
hierarchy	<pre> classDiagram     class WizardPreviousPanelController {         &lt;&lt;AbstractController&lt;M&gt;&gt;         &lt;&lt;JReportWizard&lt;M-&gt;?,V-&gt;?&gt;&gt;     } </pre>
owner	<a href="#">reportmanagement</a>
template parameters	name kind constrainingClassifier default M Class <a href="#">AbstractReportModel</a> V Class <a href="#">JReportWizard&lt;M-&gt;?,V-&gt;?&gt;</a>
ownedMember	<a href="#">actionPerformed WizardPreviousPanelController</a>
general	<a href="#">AbstractController&lt;M-&gt;M,V-&gt;V&gt;</a>
target of relation	ComponentRealization <a href="#">Implementierung</a>
documentation	<p>This controller is responsible for managing the wizard navigation to the preceding pane.</p> <p>@author David Kaufman</p> <p>@param &lt;M&gt;</p> <p>@param &lt;V&gt;</p>

## Operation WizardPreviousPanelController::actionPerformed

owner	<a href="#">WizardPreviousPanelController</a>
parameter	name direction type multiplicity default e in ActionEvent return return void
ownedMember	e return

## Operation WizardPreviousPanelController::WizardPreviousPanelController

owner	<a href="#">WizardPreviousPanelController</a>
parameter	name direction type multiplicity default view in V
ownedMember	view

## Package XML

owner	<a href="#">studien</a>
ownedMember	<a href="#">Object2XML XML2Object</a>

## Class XML2Object

diagram	
owner	<a href="#">XML</a>
ownedMember	<a href="#">main</a>
target of relation	ComponentRealization <a href="#">Implementierung</a>
documentation	This shows how to use JAXB to unmarshal an xml file Then display the information from the content tree

## Operation XML2Object::main

owner	<a href="#">XML2Object</a>
parameter	name direction type multiplicity default <b>args</b> <b>in</b> <b>String</b> * <b>return</b> <b>return void</b>
ownedMember	<b>args return</b>

## Package controller

owner	<a href="#">knipsX</a>
ownedMember	<a href="#">AbstractController</a> <a href="#">diagrams</a> <a href="#">projectmanagement</a> <a href="#">projectview</a> <a href="#">reportmanagement</a>

## Package **dataformathelper**

owner	<a href="#"><u>shared</u></a>
ownedMember	<a href="#"><u>JDataFormatHelper</u></a>

## Package **dataformathelper**

owner	<a href="#"><u>shared</u></a>
ownedMember	<a href="#"><u>JDataFormatHelper</u></a>

## Package diagrams

owner	<a href="#">controller</a>
ownedMember	<a href="#">AbstractController&lt;M-&gt;M,V-&gt;V&gt;</a> <a href="#">AbstractController&lt;M-&gt;M,V-&gt;V&gt;</a> <a href="#">AbstractController&lt;M-&gt;M,V-&gt;V&gt;</a> <a href="#">AbstractController&lt;M-&gt;M,V-&gt;V&gt;</a> <a href="#">AbstractController&lt;M-&gt;M,V-&gt;V&gt;</a> <a href="#">AbstractController&lt;M-&gt;M,V-&gt;V&gt;</a> <a href="#">DiagramCloseController</a> <a href="#">DiagramExportAsJPGController</a> <a href="#">DiagramResetViewController</a> <a href="#">JAbstract3DView&lt;M-&gt;?&gt;</a> <a href="#">JAbstractDiagram&lt;M-&gt;?&gt;</a> <a href="#">JCluster3D&lt;M-&gt;?&gt;</a> <a href="#">ReportEditController</a> <a href="#">View3DClickController</a> <a href="#">View3DCycleController</a>

## Package diagrams

owner	<a href="#">view</a>
ownedMember	<a href="#">JAbstract2DDiagram</a> <a href="#">JAbstract2DDiagram&lt;M-&gt;M&gt;</a> <a href="#">JAbstract2DDiagram&lt;M-&gt;M&gt;</a> <a href="#">JAbstract3DDiagram</a> <a href="#">JAbstract3DDiagram&lt;M-&gt;M&gt;</a> <a href="#">JAbstract3DDiagram&lt;M-&gt;M&gt;</a> <a href="#">JAbstract3DView</a> <a href="#">JAbstract3DView&lt;M-&gt;M&gt;</a> <a href="#">JAbstract3DView&lt;M-&gt;M&gt;</a> <a href="#">JAbstractDiagram</a> <a href="#">JAbstractDiagram&lt;M-&gt;M&gt;</a> <a href="#">JAbstractDiagram&lt;M-&gt;M&gt;</a> <a href="#">JAbstractDiagram&lt;M-&gt;M&gt;</a> <a href="#">JAbstractView&lt;M-&gt;M&gt;</a> <a href="#">JBoxplot</a> <a href="#">JCluster3D</a> <a href="#">JDiagramButtons2D</a> <a href="#">JDiagramButtons3D</a> <a href="#">JDiagramButtonsPlain</a> <a href="#">JHistogram2D</a> <a href="#">JHistogram3D</a> <a href="#">JTableDiagram</a> <a href="#">JTextDiagram</a> <a href="#">Perspectives</a> <a href="#">SelectableCluster3DSphere</a>

## Package **exifAdapter**

owner	<a href="#">utils</a>
ownedMember	<a href="#">jexifviewer</a>

## Package exifAdapter

owner	<a href="#">studien</a>
ownedMember	<a href="#">ArrayList&lt;E-&gt;ExifAdapter&gt; jexifviewer Main</a>

## Package fileChooser

owner	<a href="#"><u>studien</u></a>
ownedMember	<a href="#"><u>DemoJFileChooser</u></a>

## Package fileChoser

owner	<a href="#"><u>studien</u></a>
ownedMember	<a href="#"><u>DemoJFileChooser</u></a>

## Package files

owner	<a href="#"><u>shared</u></a>
ownedMember	<a href="#"><u>ArrayList&lt;E-&gt;String&gt;</u></a> <a href="#"><u>JFileHelper</u></a> <a href="#"><u>JMyFileFilter</u></a> <a href="#"><u>JPathHelper</u></a> <a href="#"><u>JRelPathHelper</u></a>

## Package files

owner	<a href="#"><u>shared</u></a>
ownedMember	<a href="#"><u>ArrayList&lt;E-&gt;String&gt;</u></a> <a href="#"><u>JFileHelper</u></a> <a href="#"><u>JMyFileFilter</u></a> <a href="#"><u>JPathHelper</u></a> <a href="#"><u>JRelPathHelper</u></a>

## Package **jexifviewer**

owner	<a href="#">exifAdapter</a>
ownedMember	<a href="#">ArrayList&lt;E-&gt;JExifTag&gt;</a> <a href="#">ExifAdapter</a> <a href="#">JExif</a> <a href="#">JExifTag</a> <a href="#">JIfd</a> <a href="#">JIfdData</a> <a href="#">JJPEGHelper</a> <a href="#">JTiffHeader</a> <a href="#">shared</a>

## Package **jexifviewer**

owner	<a href="#">exifAdapter</a>
ownedMember	<a href="#">ArrayList&lt;E-&gt;JExifTag&gt;</a> <a href="#">ExifAdapter</a> <a href="#">JExif</a> <a href="#">JExifTag</a> <a href="#">JIfd</a> <a href="#">JIfdData</a> <a href="#">JJPEGHelper</a> <a href="#">JTiffHeader</a> <a href="#">shared</a>

## Package knipsX

owner	<a href="#">org</a>
ownedMember	<a href="#">controller</a> <a href="#">model</a> <a href="#">Programm</a> <a href="#">utils</a> <a href="#">view</a>

## Package model

owner	<a href="#">knipsX</a>
ownedMember	<a href="#">AbstractModel</a> <a href="#">picturemanagement</a> <a href="#">projectmanagement</a> <a href="#">projectview</a> <a href="#">reportmanagement</a>

## Package org

owner	<a href="#">Package1</a>
ownedMember	<a href="#">knipsX studien</a>

## Package picturemanagement

owner	<a href="#">model</a>
ownedMember	<a href="#">Directory Iterator&lt;E-&gt;Picture&gt;</a> <a href="#">List&lt;E-&gt;?&gt;</a> <a href="#">List&lt;E-&gt;Picture&gt;</a> <a href="#">List&lt;E-&gt;PictureContainer&gt;</a> <a href="#">Picture</a> <a href="#">PictureContainer</a> <a href="#">PictureSet</a>

## Package **plotter3D**

owner	<a href="#"><u>studien</u></a>
ownedMember	<a href="#"><b>Pointplotter PointPlotterWindow</b></a>

UML documentation generated by [\*\*UModel\*\*](#) UML Editor <http://www.altova.com/umodel>

12/17/09 13:54:45

## Package projectmanagement

owner	<a href="#">controller</a>
ownedMember	<a href="#">AbstractController&lt;M-&gt;M,V-&gt;V&gt;</a> <a href="#">AbstractController&lt;M-&gt;M,V-&gt;V&gt;</a> <a href="#">AbstractController&lt;M-&gt;M,V-&gt;V&gt;</a> <a href="#">AbstractController&lt;M-&gt;M,V-&gt;V&gt;</a> <a href="#">JProjectManagement&lt;M-&gt;?&gt;</a> <a href="#">JProjectManagement&lt;M-&gt;M&gt;</a> <a href="#">JProjectManagement&lt;M-&gt;M&gt;</a> <a href="#">JProjectManagement&lt;M-&gt;M&gt;</a> <a href="#">ProjectCopyController</a> <a href="#">ProjectCreateController</a> <a href="#">ProjectDeleteController</a> <a href="#">ProjectOpenController</a>

## Package projectmanagement

owner	<a href="#">model</a>
ownedMember	<a href="#">List&lt;E-&gt;ProjectEntry&gt; ProjectManagementModel</a>

## Package projectmanagement

owner	<a href="#">view</a>
ownedMember	<a href="#">JAbstractView&lt;M-&gt;M&gt;</a> <a href="#">JProjectManagement</a> <a href="#">MyProjectListCellRenderer</a>

## Package projectview

## Package projectview

owner	<a href="#">model</a>
ownedMember	<a href="#">List&lt;E-&gt;AbstractReportModel&gt;</a> <a href="#">List&lt;E-&gt;PictureSet&gt;</a> <a href="#">ProjectEntry</a> <a href="#">ProjectModel</a>

## Package projectview

owner	<a href="#">view</a>
ownedMember	<a href="#">JAbstractView&lt;M-&gt;M&gt;</a> <a href="#">JProjectView</a> <a href="#">MyPictureListCellRenderer</a> <a href="#">MyPictureSetContentListCellRenderer</a> <a href="#">MyPictureSetListCellRenderer</a> <a href="#">MyReportListCellRenderer</a> <a href="#">MyTableCellRenderer</a>

## Package reportmanagement

owner	<a href="#">controller</a>
ownedMember	<a href="#">AbstractController&lt;M-&gt;M,V-&gt;V&gt;</a> <a href="#">AbstractController&lt;M-&gt;M,V-&gt;V&gt;</a> <a href="#">AbstractController&lt;M-&gt;M,V-&gt;V&gt;</a> <a href="#">AbstractController&lt;M-&gt;M,V-&gt;V&gt;</a> <a href="#">AbstractController&lt;M-&gt;M,V-&gt;V&gt;</a> <a href="#">AbstractController&lt;M-&gt;M,V-&gt;V&gt;</a> <a href="#">AbstractController&lt;M-&gt;M,V-&gt;V&gt;</a> <a href="#">AbstractController&lt;M-&gt;M,V-&gt;V&gt;</a> <a href="#">AbstractController&lt;M-&gt;M,V-&gt;V&gt;</a> <a href="#">AbstractReportCompilation&lt;M-&gt;AbstractReportModel&gt;</a> <a href="#">DiagramTypeSelectController</a> <a href="#">JReportWizard&lt;M-&gt;?,V-&gt;?&gt;</a> <a href="#">ReportAddExifKeywordController</a> <a href="#">ReportAddPictureSetController</a> <a href="#">ReportCloseController</a> <a href="#">ReportPictureSetRemoveController</a> <a href="#">ReportRemoveExifKeywordController</a> <a href="#">ReportSaveController</a> <a href="#">WizardCloseController</a> <a href="#">WizardNextPanelController</a> <a href="#">WizardPreviousPanelController</a>

## Package reportmanagement

owner	<a href="#">model</a>
ownedMember	<a href="#">AbstractDoubleAxesModel</a> <a href="#">AbstractReportModel</a> <a href="#">AbstractSingleAxisModel</a> <a href="#">AbstractTripleAxesModel</a> <a href="#">ArrayList&lt;E-&gt;PictureContainer&gt;</a> <a href="#">Axis</a> <a href="#">Bar</a> <a href="#">Boxplot</a> <a href="#">BoxplotModel</a> <a href="#">Category</a> <a href="#">Cluster3DModel</a> <a href="#">Frequency3DPoint</a> <a href="#">Histogram2DModel</a> <a href="#">Histogram3DModel</a> <a href="#">PictureParameter</a> <a href="#">TableModel</a> <a href="#">TextModel</a> <a href="#">WilcoxonTestType</a>

## Package reportmanagement

UML documentation generated by **UModel** UML Editor <http://www.altova.com/umodel>

12/17/09 13:54:45

**Package shared**

owner	<a href="#">jexifviewer</a>
ownedMember	<a href="#">dataformathelper files</a>

**Package shared**

owner	<a href="#">jexifviewer</a>
ownedMember	<a href="#">dataformathelper files</a>

## Package studien

owner	<a href="#">org</a>
ownedMember	<a href="#">exifAdapter</a> <a href="#">fileChooser</a> <a href="#">fileChoser</a> <a href="#">Kompositum</a> <a href="#">plotter3D</a> <a href="#">thumbnail</a> <a href="#">tooltip</a> <a href="#">XML</a>

## Package thumbnail

owner	<a href="#">studien</a>
ownedMember	<a href="#">DemoThumbnailQuality</a> <a href="#">DemoThumbnailSpeed</a>

## Package **thumbnailAdapeter**

owner	<a href="#">utils</a>
ownedMember	<a href="#">PictureThumbnailAdapter</a>

## Package tooltip

owner	<a href="#">studien</a>
ownedMember	<a href="#">DemoTooltip</a>

## Package utils

owner	<a href="#">knipsX</a>
ownedMember	<a href="#">exifAdapter</a> <a href="#">ExifParameter</a> <a href="#">FileChooser</a> <a href="#">FileHandler</a> <a href="#">JFileSaver</a> <a href="#">List&lt;E-&gt;</a> <a href="#">ProjectEntry</a> <a href="#">StringChecker</a> <a href="#">ThumbnailAdapeter</a>

## Package view

owner	<a href="#">knipsX</a>
ownedMember	<a href="#">diagrams</a> <a href="#">JAbstractView</a> <a href="#">projectmanagement</a> <a href="#">projectview</a> <a href="#">reportmanagement</a>