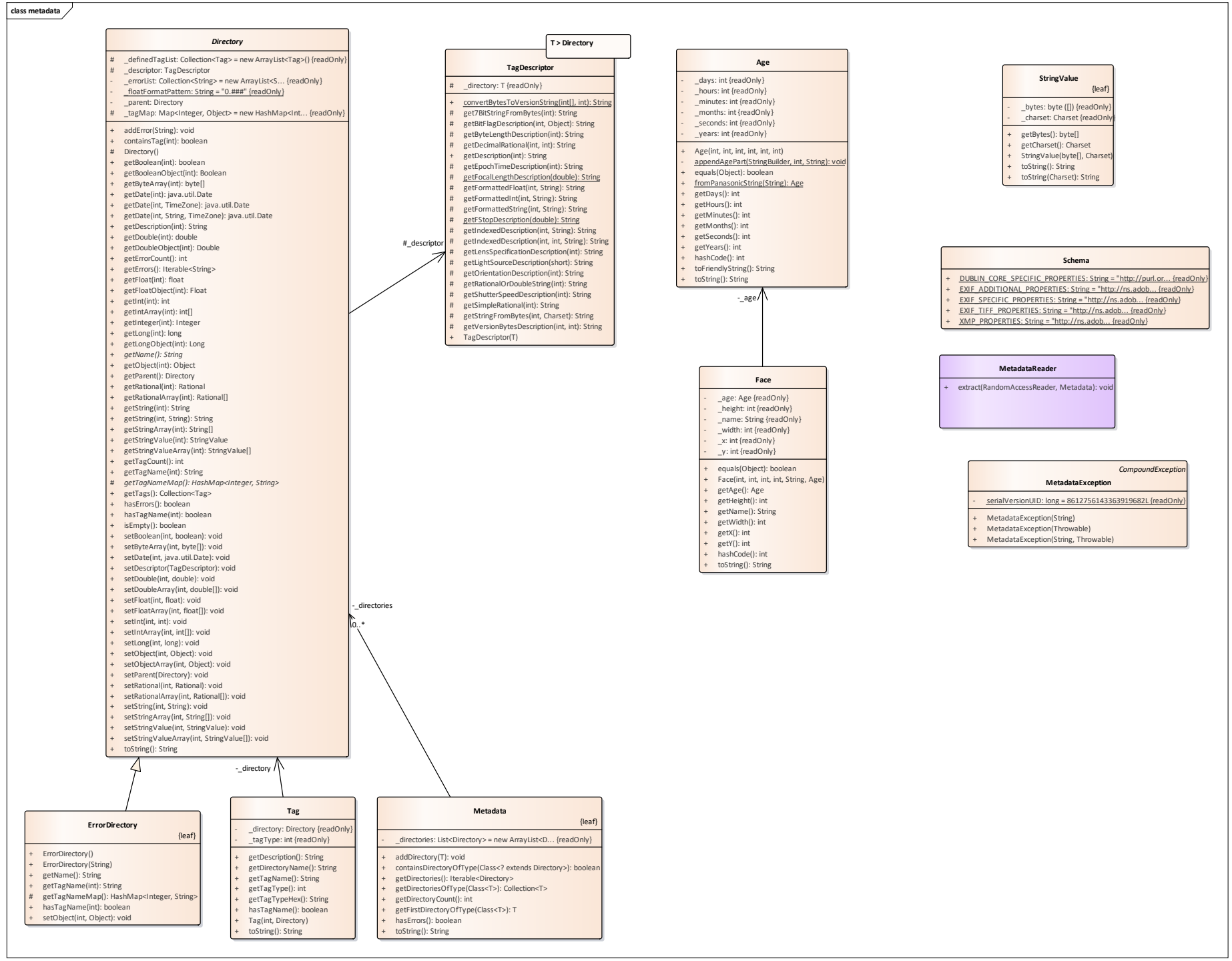
metadata-extractor

Reengineering mit Enterprise Architect

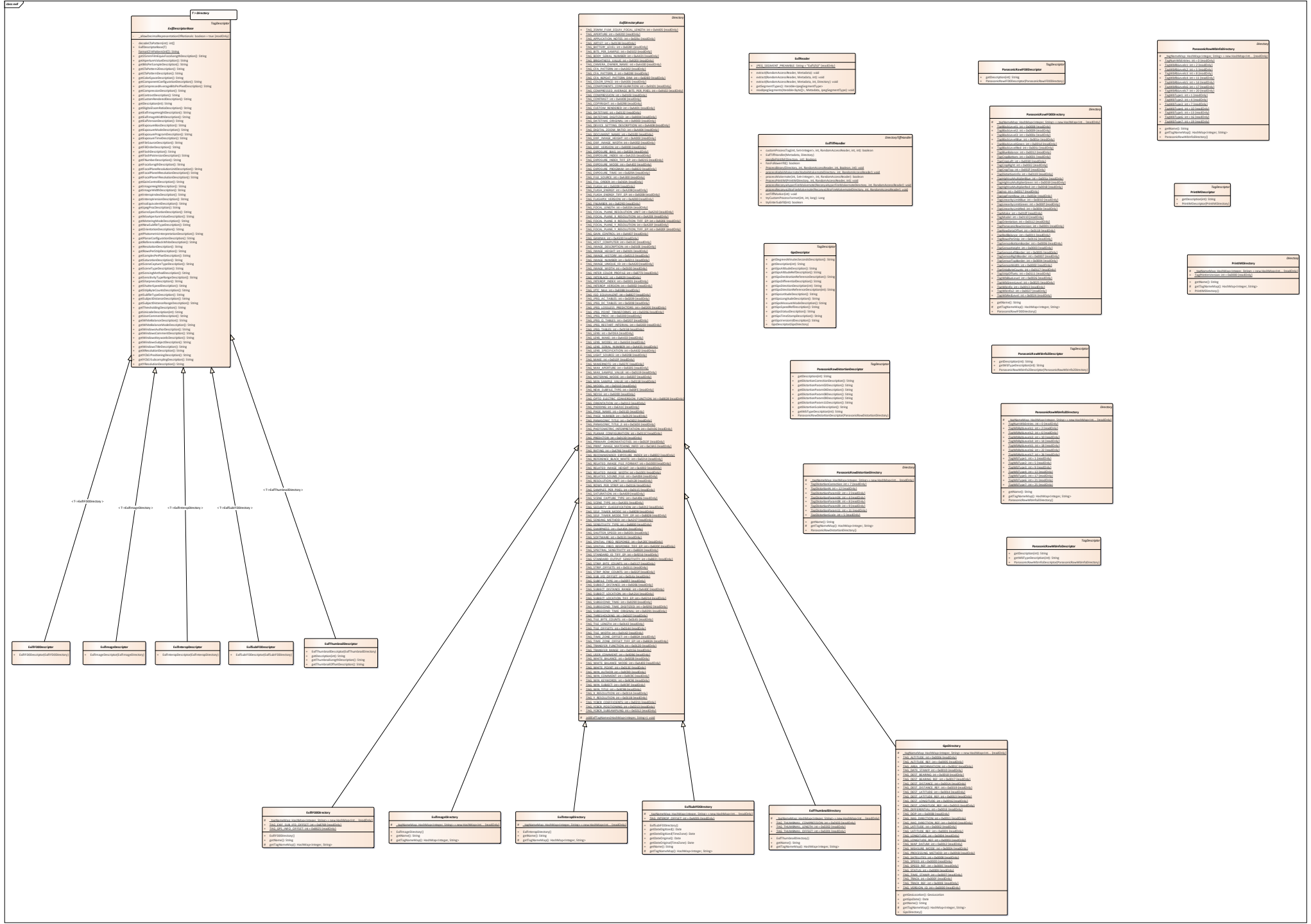
21.22.2017

Dr. Ingo Kreuz

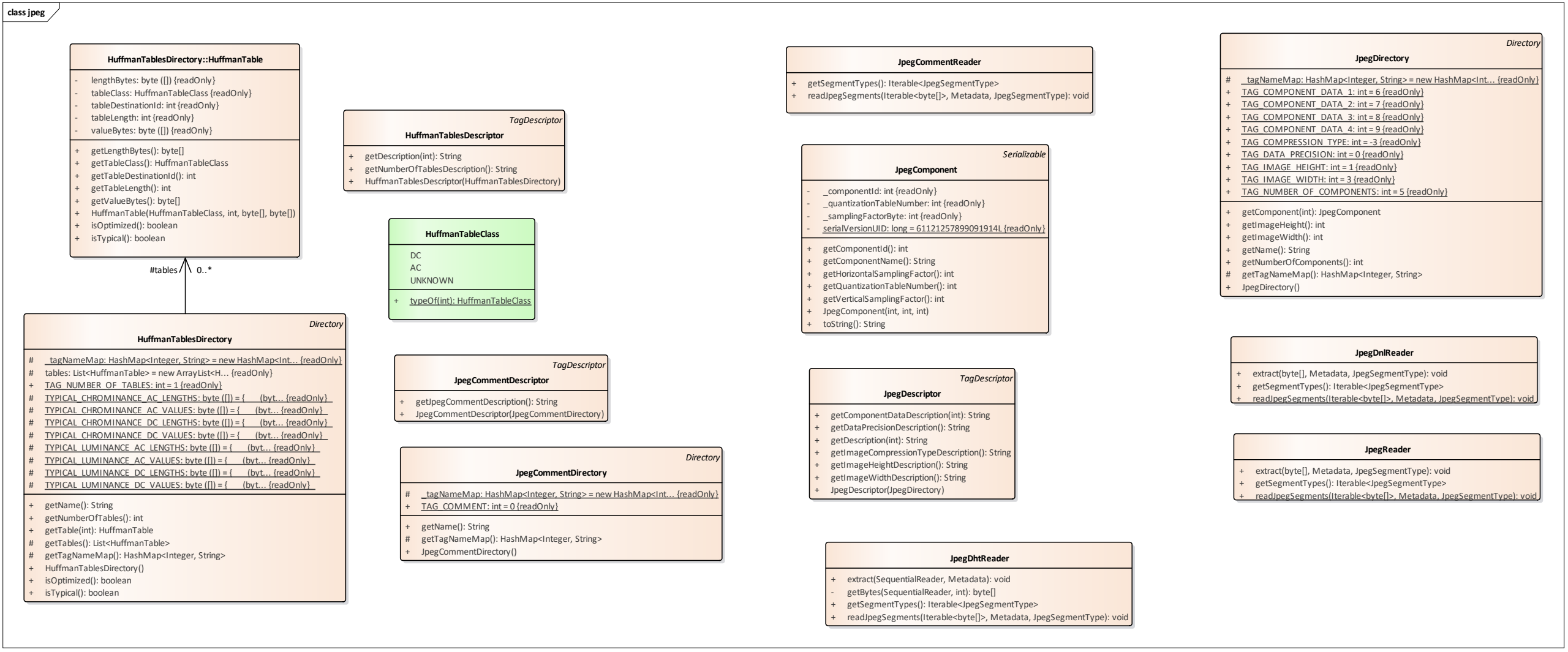
metadata


class metadata
Directory
_definedTagList: Collection<Tag> = new ArrayList<Tag>() {readOnly}
_descriptor: TagDescriptor
- _errorList: Collection<String> = new ArrayList<S... {readOnly}
- floatFormatPattern: String = "0.###" {readOnly}
- _parent: Directory
_tagMap: Map<Integer, Object> = new HashMap<Int... {readOnly}
+ addError(String): void
+ containsTag(int): boolean
# Directory()
+ getBoolean(int): boolean
+ getBooleanObject(int): Boolean
+ getByteArray(int): byte[]
+ getDate(int): java.util.Date
+ getDate(int, TimeZone): java.util.Date
+ getDate(int, String, TimeZone): java.util.Date
+ getDescription(int): String
+ getDouble(int): double
+ getDoubleObject(int): Double
+ getErrorCount(): int
+ getErrors(): Iterable<String>
+ getFloat(int): float
+ getFloatObject(int): Float
+ getInt(int): int
+ getIntArray(int): int[]
+ getInteger(int): Integer
+ getLong(int): long
+ getLongObject(int): Long
+ getName(): String
+ getObject(int): Object
+ getParent(): Directory
+ getRational(int): Rational
+ getRationalArray(int): Rational[]
+ getString(int): String
+ getString(int, String): String
+ getStringArray(int): String[]
+ getStringValue(int): StringValue
+ getStringValueArray(int): StringValue[]
+ getTagCount(): int
+ getTagName(int): String
# getTagNameMap(): HashMap<Integer, String>
+ getTags(): Collection<Tag>
+ hasErrors(): boolean
+ hasTagName(int): boolean
+ isEmpty(): boolean
+ setBoolean(int, boolean): void
+ setByteArray(int, byte[]): void
+ setDate(int, java.util.Date): void
+ setDescriptor(TagDescriptor): void
+ setDouble(int, double): void
+ setDoubleArray(int, double[]): void
+ setFloat(int, float): void
+ setFloatArray(int, float[]): void
+ setInt(int, int): void
+ setIntArray(int, int[]): void
+ setLong(int, long): void
+ setObject(int, Object): void
+ setObjectArray(int, Object): void
+ setParent(Directory): void
+ setRational(int, Rational): void
+ setRationalArray(int, Rational[]): void
+ setString(int, String): void
+ setStringArray(int, String[]): void
+ setStringValue(int, StringValue): void
+ setStringValueArray(int, StringValue[]): void
+ toString(): String
T > Directory
TagDescriptor
_directory: T {readOnly}
+ convertBytesToVersionString(int[], int): String
# get7BitStringFromBytes(int): String
# getBitFlagDescription(int, Object): String
# getByteLengthDescription(int): String
# getDecimalRational(int, int): String
+ getDescription(int): String
# getEpochTimeDescription(int): String
# getFocalLengthDescription(double): String
# getFormattedFloat(int, String): String
# getFormattedInt(int, String): String
# getFormattedString(int, String): String
# getFStopDescription(double): String
# getIndexedDescription(int, String): String
# getIndexedDescription(int, int, String): String
# getLensSpecificationDescription(int): String
# getLightSourceDescription(short): String
# getOrientationDescription(int): String
# getRationalOrDoubleString(int): String
# getShutterSpeedDescription(int): String
# getSimpleRational(int): String
# getStringFromBytes(int, Charset): String
# getVersionBytesDescription(int, int): String
+ TagDescriptor(T)
#_descriptor
Age
- _days: int {readOnly}
- _hours: int {readOnly}
- _minutes: int {readOnly}
- _months: int {readOnly}
- _seconds: int {readOnly}
- _years: int {readOnly}
+ Age(int, int, int, int, int, int)
- appendAgePart(StringBuilder, int, String): void
+ equals(Object): boolean
+ fromPanasonicString(String): Age
+ getDays(): int
+ getHours(): int
+ getMinutes(): int
+ getMonths(): int
+ getSeconds(): int
+ getYears(): int
+ hashCode(): int
+ toFriendlyString(): String
+ toString(): String
-_age
Face
- _age: Age {readOnly}
- _height: int {readOnly}
- _name: String {readOnly}
- _width: int {readOnly}
- _x: int {readOnly}
- _y: int {readOnly}
+ equals(Object): boolean
+ Face(int, int, int, int, String, Age)
+ getAge(): Age
+ getHeight(): int
+ getName(): String
+ getWidth(): int
+ getX(): int
+ getY(): int
+ hashCode(): int
+ toString(): String
StringValue
{leaf}
- _bytes: byte ([]) {readOnly}
- _charset: Charset {readOnly}
+ getBytes(): byte[]
+ getCharset(): Charset
+ StringValue(byte[], Charset)
+ toString(): String
+ toString(Charset): String
Schema
+ DUBLIN_CORE_SPECIFIC_PROPERTIES: String = "http://purl.or... {readOnly}
+ EXIF_ADDITIONAL_PROPERTIES: String = "http://ns.adob... {readOnly}
+ EXIF_SPECIFIC_PROPERTIES: String = "http://ns.adob... {readOnly}
+ EXIF_TIFF_PROPERTIES: String = "http://ns.adob... {readOnly}
+ XMP_PROPERTIES: String = "http://ns.adob... {readOnly}
MetadataReader
+ extract(RandomAccessReader, Metadata): void
CompoundException
MetadataException
- serialVersionUID: long = 8612756143363919682L {readOnly}
+ MetadataException(String)
+ MetadataException(Throwable)
+ MetadataException(String, Throwable)
-_directories
0..*
-_directory
ErrorDirectory
{leaf}
+ ErrorDirectory()
+ ErrorDirectory(String)
+ getName(): String
+ getTagName(int): String
# getTagNameMap(): HashMap<Integer, String>
+ hasTagName(int): boolean
+ setObject(int, Object): void
Tag
- _directory: Directory {readOnly}
- _tagType: int {readOnly}
+ getDescription(): String
+ getDirectoryName(): String
+ getTagName(): String
+ getTagType(): int
+ getTagTypeHex(): String
+ hasTagName(): boolean
+ Tag(int, Directory)
+ toString(): String
Metadata
{leaf}
- _directories: List<Directory> = new ArrayList<D... {readOnly}
+ addDirectory(T): void
+ containsDirectoryOfType(Class<? extends Directory>): boolean
+ getDirectories(): Iterable<Directory>
+ getDirectoriesOfType(Class<T>): Collection<T>
+ getDirectoryCount(): int
+ getFirstDirectoryOfType(Class<T>): T
+ hasErrors(): boolean
+ toString(): String

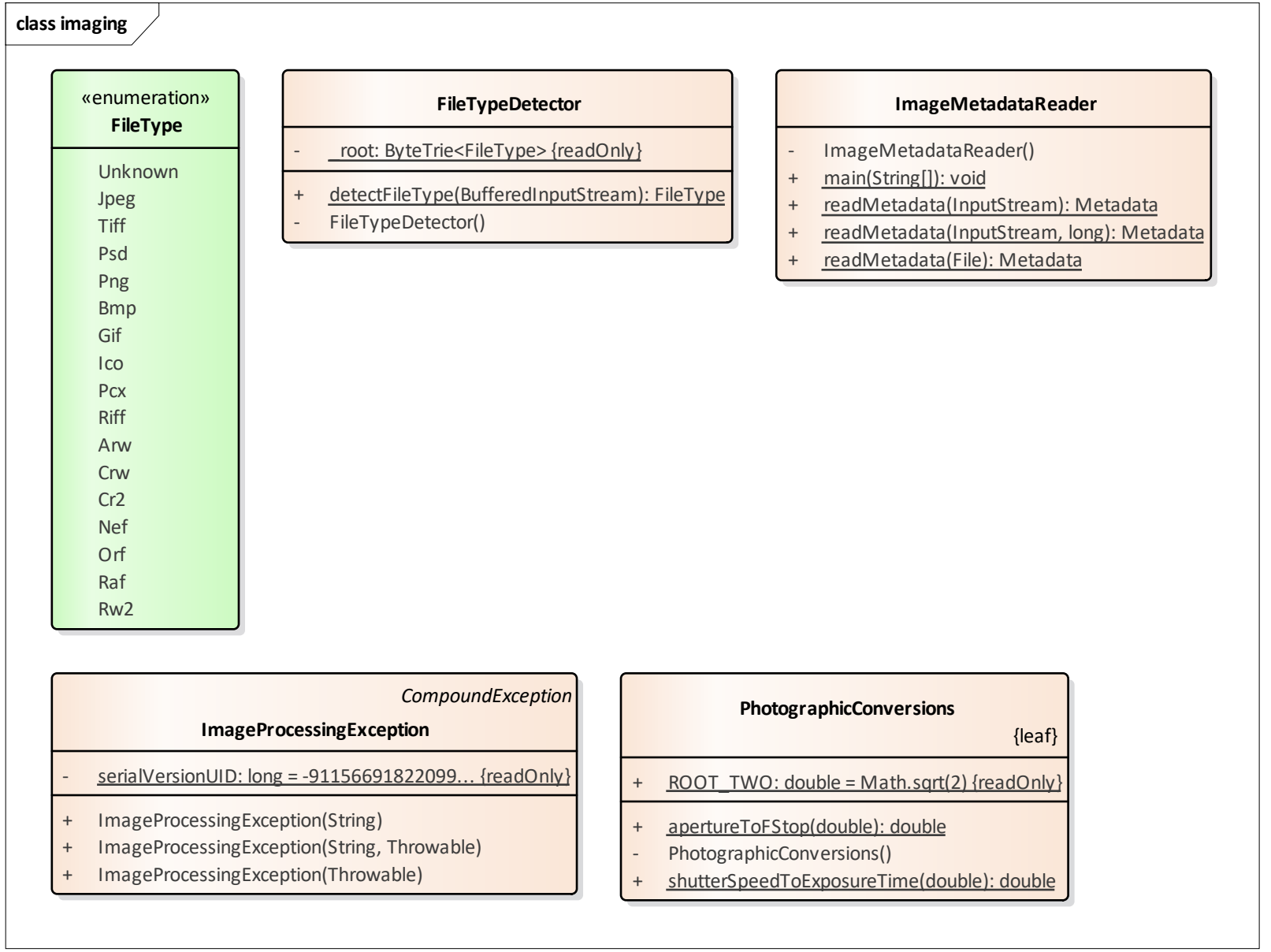
exif


class exif
ExifDescriptorBase
ExifDirectoryBase
ExifIFD0Directory
ExifIFD0Descriptor
ExifSubIFDDirectory
ExifSubIFDDescriptor
ExifInteropDirectory
ExifInteropDescriptor
ExifThumbnailDirectory
ExifThumbnailDescriptor
ExifReader
JpegDirectory
JpegDescriptor
JpegReader
PanasonicRawDistortionDirectory
PanasonicRawDistortionDescriptor
PanasonicRawIFD0Directory
PanasonicRawIFD0Descriptor
PanasonicRawWbInfo2Directory
PanasonicRawWbInfo2Descriptor
PanasonicRawWbInfoDirectory
PanasonicRawWbInfoDescriptor
PanasonicRawDirectory
PrintIMDirectory
PrintIMDescriptor

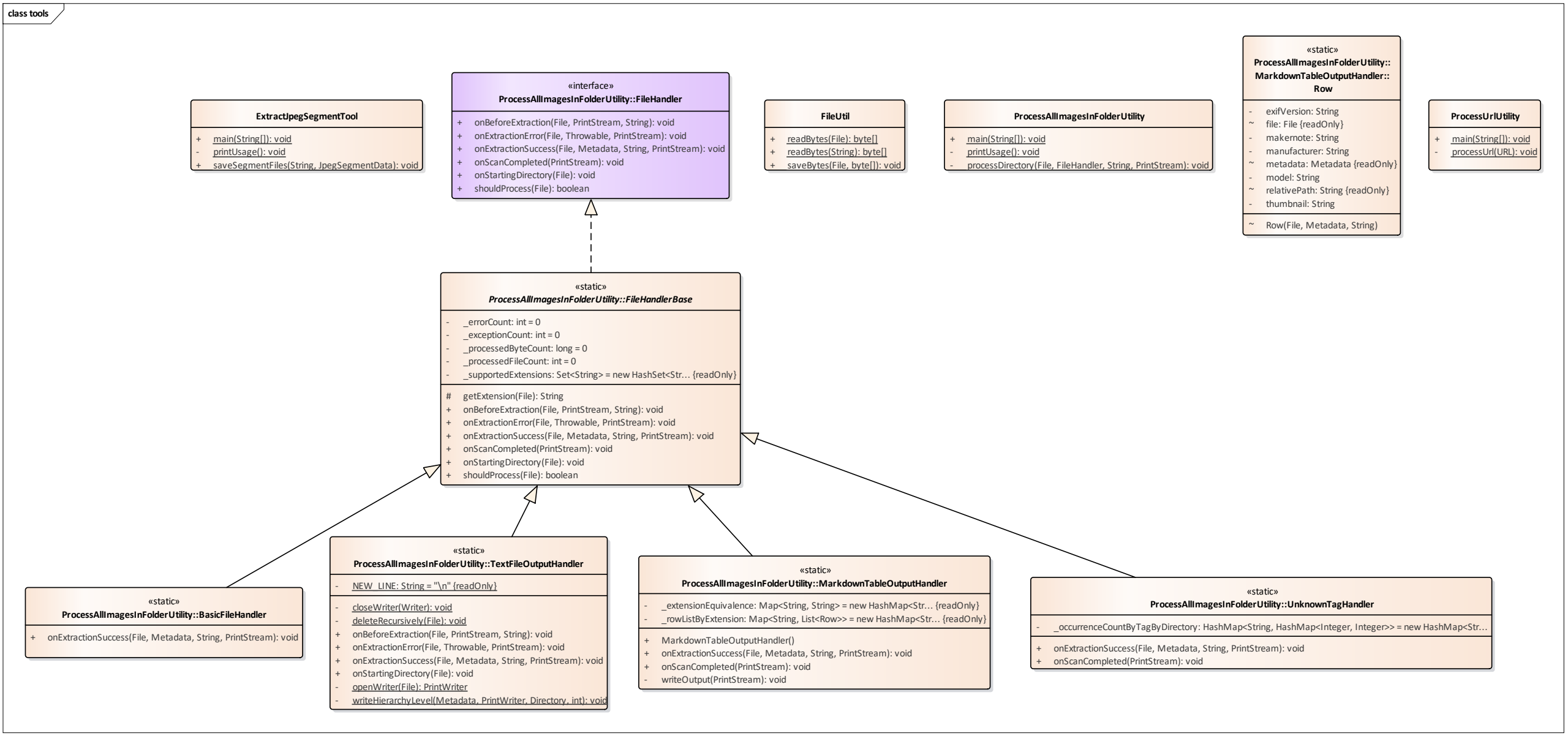
jpeg


class jpeg
HuffmanTablesDirectory::HuffmanTable
- lengthBytes: byte ([]) {readOnly}
- tableClass: HuffmanTableClass {readOnly}
- tableDestinationId: int {readOnly}
- tableLength: int {readOnly}
- valueBytes: byte ([]) {readOnly}
+ getLengthBytes(): byte[]
+ getTableClass(): HuffmanTableClass
+ getTableDestinationId(): int
+ getTableLength(): int
+ getValueBytes(): byte[]
+ HuffmanTable(HuffmanTableClass, int, byte[], byte[])
+ isOptimized(): boolean
+ isTypical(): boolean
#tables
0..*
Directory
HuffmanTablesDirectory
_tagNameMap: HashMap<Integer, String> = new HashMap<Int... {readOnly}
tables: List<HuffmanTable> = new ArrayList<H... {readOnly}
+ TAG_NUMBER_OF_TABLES: int = 1 {readOnly}
# TYPICAL_CHROMINANCE_AC_LENGTHS: byte ([]) = {... (byt... {readOnly}
# TYPICAL_CHROMINANCE_AC_VALUES: byte ([]) = {... (byt... {readOnly}
# TYPICAL_CHROMINANCE_DC_LENGTHS: byte ([]) = {... (byt... {readOnly}
# TYPICAL_CHROMINANCE_DC_VALUES: byte ([]) = {... (byt... {readOnly}
# TYPICAL_LUMINANCE_AC_LENGTHS: byte ([]) = {... (byt... {readOnly}
# TYPICAL_LUMINANCE_AC_VALUES: byte ([]) = {... (byt... {readOnly}
# TYPICAL_LUMINANCE_DC_LENGTHS: byte ([]) = {... (byt... {readOnly}
# TYPICAL_LUMINANCE_DC_VALUES: byte ([]) = {... (byt... {readOnly}
+ getName(): String
+ getNumberOfTables(): int
+ getTable(int): HuffmanTable
# getTables(): List<HuffmanTable>
# getTagNameMap(): HashMap<Integer, String>
+ HuffmanTablesDirectory()
+ isOptimized(): boolean
+ isTypical(): boolean
TagDescriptor
HuffmanTablesDescriptor
+ getDescription(int): String
+ getNumberOfTablesDescription(): String
+ HuffmanTablesDescriptor(HuffmanTablesDirectory)
HuffmanTableClass
DC
AC
UNKNOWN
+ typeOf(int): HuffmanTableClass
TagDescriptor
JpegCommentDescriptor
+ getJpegCommentDescription(): String
+ JpegCommentDescriptor(JpegCommentDirectory)
Directory
JpegCommentDirectory
_tagNameMap: HashMap<Integer, String> = new HashMap<Int... {readOnly}
+ TAG_COMMENT: int = 0 {readOnly}
+ getName(): String
# getTagNameMap(): HashMap<Integer, String>
+ JpegCommentDirectory()
JpegCommentReader
+ getSegmentTypes(): Iterable<JpegSegmentType>
+ readJpegSegments(Iterable<byte[]>, Metadata, JpegSegmentType): void
Serializable
JpegComponent
- _componentId: int {readOnly}
- _quantizationTableNumber: int {readOnly}
- _samplingFactorByte: int {readOnly}
- serialVersionUID: long = 61121257899091914L {readOnly}
+ getComponentId(): int
+ getComponentName(): String
+ getHorizontalSamplingFactor(): int
+ getQuantizationTableNumber(): int
+ getVerticalSamplingFactor(): int
+ JpegComponent(int, int, int)
+ toString(): String
TagDescriptor
JpegDescriptor
+ getComponentDataDescription(int): String
+ getDataPrecisionDescription(): String
+ getDescription(int): String
+ getImageCompressionTypeDescription(): String
+ getImageHeightDescription(): String
+ getImageWidthDescription(): String
+ JpegDescriptor(JpegDirectory)
JpegDhtReader
+ extract(SequentialReader, Metadata): void
- getBytes(SequentialReader, int): byte[]
+ getSegmentTypes(): Iterable<JpegSegmentType>
+ readJpegSegments(Iterable<byte[]>, Metadata, JpegSegmentType): void
Directory
JpegDirectory
_tagNameMap: HashMap<Integer, String> = new HashMap<Int... {readOnly}
+ TAG_COMPONENT_DATA_1: int = 6 {readOnly}
+ TAG_COMPONENT_DATA_2: int = 7 {readOnly}
+ TAG_COMPONENT_DATA_3: int = 8 {readOnly}
+ TAG_COMPONENT_DATA_4: int = 9 {readOnly}
+ TAG_COMPRESSION_TYPE: int = -3 {readOnly}
+ TAG_DATA_PRECISION: int = 0 {readOnly}
+ TAG_IMAGE_HEIGHT: int = 1 {readOnly}
+ TAG_IMAGE_WIDTH: int = 3 {readOnly}
+ TAG_NUMBER_OF_COMPONENTS: int = 5 {readOnly}
+ getComponent(int): JpegComponent
+ getImageHeight(): int
+ getImageWidth(): int
+ getName(): String
+ getNumberOfComponents(): int
# getTagNameMap(): HashMap<Integer, String>
+ JpegDirectory()
JpegDnlReader
+ extract(byte[], Metadata, JpegSegmentType): void
+ getSegmentTypes(): Iterable<JpegSegmentType>
+ readJpegSegments(Iterable<byte[]>, Metadata, JpegSegmentType): void
JpegReader
+ extract(byte[], Metadata, JpegSegmentType): void
+ getSegmentTypes(): Iterable<JpegSegmentType>
+ readJpegSegments(Iterable<byte[]>, Metadata, JpegSegmentType): void

imaging


class imaging
«enumeration»
FileType
Unknown
Jpeg
Tiff
Psd
Png
Bmp
Gif
Ico
Pcx
Riff
Arw
Crw
Cr2
Nef
Orf
Raf
Rw2
FileTypeDetector
- _root: ByteTrie<FileType> {readOnly}
+ detectFileType(BufferedInputStream): FileType
- FileTypeDetector()
ImageMetadataReader
- ImageMetadataReader()
+ main(String[]): void
+ readMetadata(InputStream): Metadata
+ readMetadata(InputStream, long): Metadata
+ readMetadata(File): Metadata
CompoundException
ImageProcessingException
- serialVersionUID: long = -91156691822099... {readOnly}
+ ImageProcessingException(String)
+ ImageProcessingException(String, Throwable)
+ ImageProcessingException(Throwable)
PhotographicConversions
{leaf}
+ ROOT_TWO: double = Math.sqrt(2) {readOnly}
+ apertureToFStop(double): double
- PhotographicConversions()
+ shutterSpeedToExposureTime(double): double

tools


class tools
ExtractJpegSegmentTool
+ main(String[]): void
- printUsage(): void
+ saveSegmentFiles(String, JpegSegmentData): void
«interface»
ProcessAllImagesInFolderUtility::FileHandler
+ onBeforeExtraction(File, PrintStream, String): void
+ onExtractionError(File, Throwable, PrintStream): void
+ onExtractionSuccess(File, Metadata, String, PrintStream): void
+ onScanCompleted(PrintStream): void
+ onStartingDirectory(File): void
+ shouldProcess(File): boolean
FileUtil
+ readBytes(File): byte[]
+ readBytes(String): byte[]
+ saveBytes(File, byte[]): void
ProcessAllImagesInFolderUtility
+ main(String[]): void
- printUsage(): void
- processDirectory(File, FileHandler, String, PrintStream): void
«static»
ProcessAllImagesInFolderUtility::MarkdownTableOutputHandler::Row
- exifVersion: String
~ file: File {readOnly}
- makernote: String
- manufacturer: String
~ metadata: Metadata {readOnly}
- model: String
~ relativePath: String {readOnly}
- thumbnail: String
~ Row(File, Metadata, String)
ProcessUrlUtility
+ main(String[]): void
- processUrl(URL): void
«static»
ProcessAllImagesInFolderUtility::FileHandlerBase
- _errorCount: int = 0
- _exceptionCount: int = 0
- _processedByteCount: long = 0
- _processedFileCount: int = 0
- _supportedExtensions: Set<String> = new HashSet<Str… {readOnly}
getExtension(File): String
+ onBeforeExtraction(File, PrintStream, String): void
+ onExtractionError(File, Throwable, PrintStream): void
+ onExtractionSuccess(File, Metadata, String, PrintStream): void
+ onScanCompleted(PrintStream): void
+ onStartingDirectory(File): void
+ shouldProcess(File): boolean
«static»
ProcessAllImagesInFolderUtility::BasicFileHandler
+ onExtractionSuccess(File, Metadata, String, PrintStream): void
«static»
ProcessAllImagesInFolderUtility::TextFileOutputHandler
- NEW_LINE: String = "\n" {readOnly}
- closeWriter(Writer): void
- deleteRecursively(File): void
+ onBeforeExtraction(File, PrintStream, String): void
+ onExtractionError(File, Throwable, PrintStream): void
+ onExtractionSuccess(File, Metadata, String, PrintStream): void
+ onStartingDirectory(File): void
- openWriter(File): PrintWriter
- writeHierarchyLevel(Metadata, PrintWriter, Directory, int): void
«static»
ProcessAllImagesInFolderUtility::MarkdownTableOutputHandler
- _extensionEquivalence: Map<String, String> = new HashMap<Str… {readOnly}
- _rowListByExtension: Map<String, List<Row>> = new HashMap<Str… {readOnly}
+ MarkdownTableOutputHandler()
+ onExtractionSuccess(File, Metadata, String, PrintStream): void
+ onScanCompleted(PrintStream): void
- writeOutput(PrintStream): void
«static»
ProcessAllImagesInFolderUtility::UnknownTagHandler
- _occurrenceCountByTagByDirectory: HashMap<String, HashMap<Integer, Integer>> = new HashMap<Str…
+ onExtractionSuccess(File, Metadata, String, PrintStream): void
+ onScanCompleted(PrintStream): void