Class Properties	Type	Access	default	Explanation
defaultSeparator	String	*public static*	"\t"	
defaultHasHeaderLine	boolean	*public static*	false	
defaultAllowErrorOutput	boolean	*public static*	true	
defaultIgnoreOutOfBounds	boolean	*public static*	true	
defaultIgnoreDiffColumns	boolean	*public static*	true	
defaultPrintStackTrace	boolean	*public static*	true	If true, exception stack trace will be sent to System out.
Protected Properties	**Type**	**Access**	**default**	**Explanation**
errorOutputComponent	JTextComponent	*protected*	null	If this is defined, errors will written to this component. Else they will be sent to System.out.
dataTable	ArrayList<String[]>	*protected*	new	Array with all the data lines. Might includ header as first element.
Instance Properties	**Type**	**Access**	**default**	**Explanation**
file	File	file( )	*constructor parameter*	CSV file currently used for reading/writing by the instance.
separator	String	separator( )	defaultSeparator	Separator between two column entries in a CSV data line. Check existing files for their separator if you want to read them.
numColumns	int	numColumns( )	0	Column size of the last line that was added.
hasHeaderLine	boolean	hasHeaderLine( )	defaultHasHeaderLine	If true, the first line in dataTable is threaten as header.
allowErrorOutput	boolean	allowErrorOutput( )	defaultAllowErrorOutput	If true, the class will notify you about any errors. All methods try to proceed even when errors occur.
ignoreOutOfBounds	boolean	ignoreOutOfBounds( )	defaultIgnoreOutOfBounds	If true, outOfBoundsExceptions will be ignored. If false, the exceptions will be threaten as errors. The methods don't care of NullPointerExceptions.
ignoreDiffColumns	boolean	ignoreDiffColumns( )	defaultIgnoreDiffColumns	If true, the class will notify you about different columns when data is added to the dataTable. However, the class can handle it.
Constructors				
	CSVManager	*public*	String / file, (opt.) String	First parameter: File object or String giving the path name for the CSV file. Second parameter: Optional separator used for this instance.
Setter Methods	**Type**	**Access**	**Parameter**	**Explanation**
setSeparator	void	*public*	String	Changes the column separator used in input & output.
setHasSetHeaderLine	void	*public*	boolean	Change if the first line is threaten as header.
setFile	void	*public*	File / String	You can call this with a file object or a string for the path.
setOutputComponent	void	*public*	JTextComponent	Defines an errorOutputComponent.
Definition Methods	**Type**	**Access**		**Explanation**
initDefaults	void	*protected*		Sets instances properties to their default value stored in default properties.
resetOutputComponent	void	*public*		Sets errorOutputComponent to null.
resetData	void	*public*		Creates a new, clean dataTable and sets numColums to 0.
Operating Methods (IO)	**Type**	**Access**	**Parameter**	**Explanation**
readCSV	boolean	*public*	(opt.) boolean	Reads the current file and adds data into the dataTable. Call it with *true* to delete existing data before reading. Default: Read data is added to existing data. Returns true if the IO operation was successful.
writeCSV	boolean	*public*	(opt.) boolean	Writes the data in dataTable to the current file. Call it with *true* to use the append option when writing. Default: Overwrite existing file data. Returns true if the IO operation was successful.
Operating Methods (Editing)	**Type**	**Access**	**Parameter**	**Explanation**
addDataLine	void	*public*	(opt.) int, String[ ]	Adds all elements of the String field as new line to dataTable. You may send an optional integer index giving the position.
addDataLines	void	*public*	(opt.) int, List<String[ ]>	Calls addDataLine for each String[ ] element of the list. You may send an optional integer index giving the position.
addCSVLine	void	*public*	(opt.) int, String	Splits the String using the defined separator and calls addDataLine using the resulting String[ ]. You may send an optional integer index giving the position.
addCSVLines	void	*public*	(opt.) int, List<String>	Calls addCSVLine for each String element of the list. You may send an optional integer index giving the position.
addHeaderDataLine	void	*public*	String[ ]	Adds all elements of the String field as header line to dataTable. The header will be added as first element. Nothing is replaced. If hasHeaderLine was false, it is set true.
addHeaderCSVLine	void	*public*	String	Splits the String using the defined separator and calls addHeaderDataLine using the resulting String[ ].
addIntegerLine	void	*public*	(opt.) int, int[ ]	Adds all elements of the integer field as new line to dataTable. You may send an optional integer index giving the position.
addIntegerLines	void	*public*	(opt.) int, List<int[ ]>	Calls addIntegerLine for each String[ ] element of the list. You may send an optional integer index giving the position.

	Type	Access	Parameter	Explanation
addDoubleLine	void	*public*	(opt.) int, double[ ]	Adds all elements of the double field as new line to dataTable. You may send an optional integer index giving the position.
addDoubleLines	void	*public*	(opt.) int, List<double[ ]>	Calls addDoubleLine for each String[ ] element of the list. You may send an optional integer index giving the position.
removeLine	boolean	*public*	int / String[ ]	Removes the line at the given integer index or removes the given String[ ] if the dataTable contains it. Returns true if the element was removed or false if the element could not be found.
removeHeaderLine	boolean	*public*	–	If hasHeaderLine = true and there is at least one line in the dataTable, the header line is removed from the dataTable Returns true in this case, otherwise it returns false. HasHeaderLine is set to false if removing was successful.
Access Methods	**Type**	**Access**	**Parameter**	**Explanation**
size	int	*public*	–	Returns the size of the dataTable (Java List method).
numLines	int	*public*	–	Returns the amount of data lines in dataTable without header.
numColumnsInLine	int	*public*	int	Returns the amount of columns in the line at the given index in the dataTable.
Access Methods (To String)	**Type**	**Access**	**Parameter**	**Explanation**
headerToString	String	*public*	–	Returns the header as CSV line containing the defined separator.
lineToString	String	*public*	int	Returns the line at the given index in the dataTable as CSV line containing the defined separator. Index 0 always addresses the first data line, never the header.
toString	String	*public*	–	Returns the dataTable as CSV lines with \n in between.