driver – Gabe

add – Padi

remove – Adam

removeOverload – Adam

move – Bart

moveOverload – Bart

moveToEnd – Padi

moveToEndOverload – Padi

rename –

index – Adam

sheetName –

display –

length –

* **remove** – an overloaded version of remove that uses an index position instead of a name.

**public String remove(int index)**

The remove operation is passed the index position of a sheet to be removed. If the index value is out of range (i.e. not in the range 1 >= index <= list length) the remove operation does nothing (i.e. the list is unaffected). If the index is valid then the sheet name at the index position should be removed from the list, unless it is the only name in the list. The spreadsheet workbook must always have at least one sheet so the name list must always have at least one name entry.

If the remove operation is successful the method returns the name of the sheet removed. Otherwise it returns a null/empty string.

* **move** – allows a sheet name to be moved from one position in the list to another.

**public int move(String from, String to, boolean before)**

move is passed two sheet names. The sheet should only be moved if BOTH sheet names are in the list and they are not the same (i.e. it is not the same sheet name twice). If any one or both of the names are not found then the move operation does nothing (i.e. the list is unaffected).

The from name is the sheet to be moved. The to name is the sheet before or after which the from sheet should be moved to. If the third parameter, before, is true the sheet should be moved to before the to sheet, otherwise it should be moved to after the to sheet.

If the move operation is successful the method returns the index of the position the sheet was moved **to**. Otherwise it returns -1.

* **move** – an overloaded version of the move using indices instead of names

**public String move(int from, int to, boolean before)**

move is passed two sheet indices. The move should only be performed if BOTH sheet indices exist (i.e. they are in the range 1 >= index <= list length) and they are not equal (i.e. it is not the same sheet index twice). If any one or both of the indices do not exist the move operation should be ignored (i.e. the list is unaffected).

The from index is the index position of the sheet to be moved. The to index is the index position of the sheet before or after which the first sheet should be moved to. If the third parameter, before, is true the sheet should be moved to before the to sheet, otherwise it should be moved to after the to sheet.

If the move operation is successful the method returns the name of the from sheet (i.e. the sheet that was moved). Otherwise it returns a null/empty string.

* **moveToEnd** – allows a sheet to be moved from its current position to the end of the list

**public String moveToEnd(int from)**

moveToEnd is passed the index position of a sheet. The moveToEnd should only be performed if the sheet index exists (i.e. is in the range 1 >= index <= list length). If it does not exist the moveToEnd operation does nothing (i.e. the list is unaffected).

The from index is the position of the sheet to be moved. The sheet at this position should be moved to the end of the list (i.e. it should become the last entry in the list or, if you prefer, it should be moved to after the current last entry in the list).

If the moveToEnd operation is successful the method returns the name the sheet that was moved. Otherwise it returns a null/empty string.

* **moveToEnd** – an overloaded version of the moveToEnd using an index instead of a name

**public int moveToEnd(String from)**

moveToEnd is passed the name of a sheet. The moveToEnd should only be performed if the sheet name exists (i.e. it is one of the names in the list). If it does not exist the moveToEnd operation does nothing (i.e. the list is unaffected).

The from string contains the name of the sheet to be moved. The sheet should be moved to the end of the list (i.e. it should become the last entry in the list or, if you prefer, it should be moved to after the current last entry in the list).

If the moveToEnd operation is successful the method returns the index position of the sheet that was moved. Otherwise it returns -1.

* **rename** – changes the name of an existing sheet

**public int rename(String currentName, String newName)**

rename is passed two sheet names. Renaming should only be performed if the currentName sheet name is in the list and the newName sheet name isn’t. Otherwise rename does nothing (i.e. the list is unaffected).

If the currentName is successfully changed to the newName then the method returns the index position of the sheet renamed. Otherwise it returns -1.

* **index** – returns the index position of a name in the list

**public int index(String sheetName)**

index is passed the name of a sheet. It searches the list for the sheet name. If it is found index returns the position of the name in the list. If it is not found index returns -1.

* **sheetName** – returns the name of the sheet at the specified index position.

**public String sheetName(int index)**

sheetName is passed an index position. If the index position exists (i.e. is in the range 1 >= index <= list length) the method returns the name of the sheet at that position. Otherwise it returns a null/empty string.

* **display** – displays the names in the list on the screen.

**public void Display()**

* **length** - returns an integer value representing the number of items in the list.

**public int length()**