JXL API

- Every Excel file is called as Excel Workbook.
- Each Workbook has number of Sheets. And each sheet may have number of Rows and Columns.
- A combination of row and column is called as Cell. Cell has actual content.
- Using JXL we read, write, create and modify excel sheets.
- This API can perform operations on .xls files only (Excel 97-2003 format only)
- It has several Interfaces and Classes.
- Useful interfaces and classes of JXL:
    1. Workbook – Abstract Class
    2. WritableWorkbook – Abstract
    3. Sheet – Interface
    4. WritableSheet – Abstract class
    5. Cell -- Interface

1. **<u>Workbook:</u>**
    - This is an abstract class
    - When we want to read excel workbook then we can create object of Workbook.
    - It is used to represent a workbook.
    - This class has several methods using which we can perform different operations on a workbook.

Methods of Workbook Class:

- getWorkbook(File file)
    o This method gets workbook specified by File object.
    o This method is overloaded.
- getSheet(int index):
    o This method reads a sheet at specified index.
    o Its return type is sheet object.
- getSheet(String name)
    o This method reads a sheet having specified name
    o Its return type is sheet object.
- getSheetNames()
    o This method returns String array of all sheets present in workbook.
- getSheets()
    o This method returns array of Sheet objects.

- o This array will contains objects for all available sheets in workbook.
2. **WritableWorkbook:**
    - This is abstract class
    - When we want to write something to Excel workbook, then we should create object of WritableWorkbook.

Methods of WritableWorkbook class:

- createSheet(String name, int index)
    - o This method is used to create a writable sheet in a workbook
    - o It takes two arguments, Sheet name and index
- getSheet(int index):
    - o This method reads a sheet at specified index.
    - o Its return type is WritableSheet object.
- getSheet(String name)
    - o This method reads a sheet having specified name
    - o Its return type is WritableSheet object.
- getSheetNames()
    - o This method returns String array of all WritableSheets present in workbook.
- getSheets()
    - o This method returns array of Sheet objects.
    - o This array will contains objects for all available WritableSheet in workbook.
3. **Sheet:**
    - This is an interface.
    - It represents a sheet within book.
    - It can provide a handle for individual cells of a Sheet.

Methods of Sheet Interface:

- getCell(int column, int row)
    - o This method returns a Cell object specified by column number and row number.
    - o If a column/row combination forms part of a merged group of cells then a blank cell will be returned
- getColumn(int col)
    - o This method returns all cells in a specified column.
    - o Its return type is array of cell objects.
- getColumns()
    - o This method returns total number of active columns from a excel sheet.
    - o Its return type is int.

- getName()
  - This method returns the name of Sheet

  - Its return type is String.
- getRow(int row)
  - This method returns all the cells in specified row.
  - Its return type is Cell[ ] (Cell array)
- getRows()
  - This method returns total number of active rows from the excel sheet.
  - Its return type is int.

4. **WritableSheet**
   - This is sub-interface of Sheet interface.
   - When we want to write something to sheet, then we should create instance of WritableWorkbook and then WritableSheet.

Methods of WritableSheet:

- addCell(WritableCell cell)
  - This method adds a specified WritableCell toa sheet.
  - This methodmay throw 'RowsExceededException' when attempt to write too many rows is made.

5. **Cell:**
   - This method represents an individual Cell within a Sheet.
   - This is an interface.

Methods of Cell Interface:

- getColumn()
  - This method returns the column number of this cell.
  - Its return type is int
- getRow()
  - This method returns the row number of this cell.
  - Its return type is int.
- getContents()
  - This method returns contents of the cell in String format only. No other formats are supported by JXL
  - Its return type is String.

Write a Program to Read data from existing Excel Sheet.

```java
public class ExcelTest {

    public static void main(String[] args) throws BiffException,
IOException {

            File fin = new File("D:\\Study\\Sample.xls");

            Workbook book = Workbook.getWorkbook(fin);

            Sheet sh = book.getSheet(0);

            int col = sh.getColumns();

            int row = sh.getRows();

            System.out.println("Cols >> " + col);

            System.out.println("Rows >> " + row);

            for (int i = 0; i < row; i++) {

                for (int k = 0; k < col; k++) {

                    System.out.println();

                    Cell cell = sh.getCell(k, i);

                    System.out.println("Row >>" + i);

                    System.out.println("Column >>" + k);

                    System.out.println(cell.getContents());

                }

            }

    }
}
```

Write a Program to write content to Excel Sheet

```java
public class WriteToExcel {

    public static void main(String[] args) throws IOException,
WriteException {

            File file = new File("D:\\Sample.xls");

            WritableWorkbook book = Workbook.createWorkbook(file);

            WritableSheet sheet = book.createSheet("Library", 0);

Object[][] library = { { "StudentName", "PhoneNumber", "CourseName" },

{ "Nitin", "9970707070", "Automation Testing" },

{ "Abhilasha", "7878909090", "Automation Testing" },

{ "Mahesh", "790908900", "J2EE" },

{ "Prashant", "8898989790", "Automation Testing" },

{ "Shafi", "9989897786", "J2EE" }, };

        int rowCount = 0;

        for (int i = 0; i <= 5; i++) {

            for (int j = 0; j <= 2; j++) {

                    WritableCell cell = sheet.getWritableCell(j, i);

                    String value = (String) library[i][j];

            jxl.write.Label label = new jxl.write.Label(j, i, value);

            sheet.addCell(label);

            }

        }

        book.write();

        System.out.println("Writing has been done Successfuly !");

        book.close();

        }

}
```

Assignments

1. Write a program to copy contents of first excel file to other excel file.
2. Write a program to fetch contents of specified column only
3. Print name of all sheets available in a workbook
4. Write a program to set size of column