```java
public class XlStudy {
    // ONE
    public String readXl() throws IOException {
        String str = null;

        File f = new File("");
        FileInputStream stream = new FileInputStream(f);
        Workbook workbook = new XSSFWorkbook(stream);
        Sheet sheet = workbook.getSheet("");
        Row row = sheet.getRow(1);
        int physicalNumberOfCells = row.getPhysicalNumberOfCells();
        Cell cell = row.getCell(2);
        CellType cellType = cell.getCellType();

        switch (cellType) {
        case STRING:
            String stringCellValue = cell.getStringCellValue();

            break;
        case NUMERIC:
            if (DateUtil.isCellDateFormatted(cell)) {
                Date dateCellValue = cell.getDateCellValue();
                SimpleDateFormat dateFormat = new SimpleDateFormat("dd-MM-YY");
                String format = dateFormat.format(dateCellValue);
            } else {
                double numericCellValue = cell.getNumericCellValue();
                BigDecimal bigDecimal = new BigDecimal(numericCellValue);
                String string = bigDecimal.toString();

            }

            break;
        default:
            break;
        }
        return str;
    }
```

```java
// 3
public void writeNewXl() throws IOException {

    Workbook workbook = new XSSFWorkbook();
    Sheet sheet = workbook.getSheet("");
    Row row = sheet.getRow(1);
    Cell cell = row.getCell(1);
    cell.setCellValue("");

    File file = new File("");
    FileOutputStream fileOutputStream = new FileOutputStream(file);
    workbook.write(fileOutputStream);

}
```

```java
// 2
public void updateXl() throws IOException {

    File file = new File("");
    FileInputStream stream = new FileInputStream(file);
    Workbook workbook = new XSSFWorkbook();
    Sheet sheet = workbook.getSheet("");
    Row row = sheet.getRow(0);
    Cell cell = row.getCell(2);
    String stringCellValue = cell.getStringCellValue();
    CellType cellType = cell.getCellType();
    if (stringCellValue.equals("")) {
        cell.setCellValue("");

    }
    FileOutputStream outputStream = new FileOutputStream(file);
    workbook.write(outputStream);

}
```

```java
// 4
public void createNewCellAndRow() throws IOException {

    File file = new File("");
    FileInputStream stream = new FileInputStream(file);
    Workbook workbook = new XSSFWorkbook();
    Sheet sheet = workbook.getSheet("");
    Row row = sheet.createRow(5);
    Cell cell = row.createCell(6);
    cell.setCellValue("");
    FileOutputStream fileOutputStream = new FileOutputStream(file);
    workbook.write(fileOutputStream);

}
```

```java
// 5
public void createNewCell() throws IOException {
    File file = new File("");
    FileInputStream stream = new FileInputStream(file);
    Workbook workbook = new XSSFWorkbook();
    Sheet sheet = workbook.getSheet("");
    Row row = sheet.getRow(1);
    Cell createCell = row.createCell(9);
    createCell.setCellValue("");
    FileOutputStream fileOutputStream = new FileOutputStream(file);
    workbook.write(fileOutputStream);

}
```