

Infograph Maker with Java

Although the author and publisher have made every effort to ensure that the information in this writing was correct at press time, the author and publisher do not assume and hereby disclaim any liability to any party for any loss, damage, or disruption caused by errors or omissions, whether such errors or omissions result from negligence, accident, or any other cause.

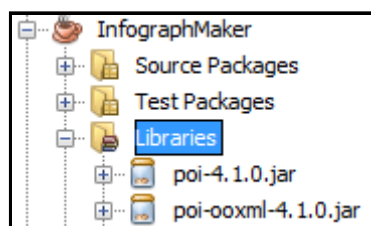
paid link



97 Day Money-Back Guarantee
Web & WordPress Hosting
\$2.59 /mo
DreamHost
Get Started

It cannot be argued that infographics is a powerful marketing tool. Many people respond better to visual information than text. Unfortunately, creating it is not an easy task. However, with the help of Java, we can try to automate the process. In this short writing, we will create a very simple infographic to visualize a small data set. The data is stored in an excel (.xls) file. It will be displayed as a bar chart.

Before anything else, you need to download *Apache POI*, a Java API for manipulating Microsoft Documents. Download the library here: <https://poi.apache.org/>. Unzip it and add the required jar files (e.g. *poi-4.1.0.jar* and *poi-ooxml-4.1.0.jar*) to your project.



Now, prepare the data. In this example, the data is about C:N ratio of common compost materials.

	A	B
1	C:N Ratio of Common Compost Materials	
2	Vegetable waste	12
3	Grass clippings	17
4	Dry leaves	40
5	Paper	170
6	Sawdust	400
7	Seaweed	19
8	Rotted manure	20
9	Grape pomace	28
10	Legume shells	30
11	Corn stalks	50
12	Oat straw	74
13	Cereal hay	32
14	Timothy hay	80

Here is the full program:

```
import java.awt.Color;
import java.awt.Font;
import java.awt.FontMetrics;
import java.awt.Graphics;
import java.awt.image.BufferedImage;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.util.ArrayList;
import javax.imageio.ImageIO;
import org.apache.poi.hssf.usermodel.HSSFWorkbook;
import org.apache.poi.ss.usermodel.Cell;
import org.apache.poi.ss.usermodel.Sheet;
import org.apache.poi.ss.usermodel.Workbook;

public class InfographMaker {
    public static void main(String[] args) throws FileNotFoundException, IOException{
        new InfographMaker();
    }

    public InfographMaker() throws FileNotFoundException, IOException{
        //get data
        File file = new File('C:\Documents\NetBeansProjects\InfographicsMaker\cnratio.xls');
        FileInputStream inputStream = new FileInputStream(file);

        Workbook workbook = new HSSFWorkbook(inputStream);
        Sheet sheet = workbook.getSheetAt(0);

        int numrow = sheet.getLastRowNum();
        int numcol = sheet.getRow(1).getLastCellNum();

        Cell cell=sheet.getRow(0).getCell(0);
        String title = cell.getStringCellValue();

        ArrayList<String> datname = new ArrayList();
        ArrayList<Integer> datnum = new ArrayList();

        for(int i=1;i<=numrow;i++){ //first row is for title
            for(int j=0;j<numcol;j++){
                cell=sheet.getRow(i).getCell(j);
                if(j == 0){
                    datname.add(cell.getStringCellValue());
                }else{
                    datnum.add((int)cell.getNumericCellValue());
                }
            }
        }

        //draw infograph (title)
```

```

int imgW = 600;
int imgH = 400;

BufferedImage bufimage = new BufferedImage(imgW,imgH,BufferedImage.TYPE_INT_ARGB);
Graphics g = bufimage.getGraphics();

g.setColor(Color.WHITE);
Font font = new Font('Arial Black',Font.BOLD,14);
g.setFont(font);

FontMetrics metrics = g.getFontMetrics(font);
int posX = (imgW - metrics.stringWidth(title)) / 2;
int posY = (int)metrics.getStringBounds(title, g).getHeight() + 10;

g.drawString(title, posX, posY);

//draw infograph (items)
int lastY = posY + 20;           //20 = vertical margin

font = new Font('Arial Black',Font.BOLD,12);
g.setFont(font);
metrics = g.getFontMetrics(font);

int datnumX = 0;                 //get the longest text for position
for (int i = 0;i < datname.size(); i++){
    if (metrics.stringWidth(datname.get(i)) > datnumX){
        datnumX = metrics.stringWidth(datname.get(i));
    }
}
datnumX += 20;                   //20 = horizontal margin

int datnumW = 0;                 //get the highest number for scaling
for (int i = 0;i < datnum.size(); i++){
    if (datnum.get(i) > datnumW){
        datnumW = datnum.get(i);
    }
}
datnumW = (imgW - datnumX - 10) / datnumW;

for (int i = 0;i < datname.size(); i++){
    int posnameX = 10;
    int textH = (int)metrics.getStringBounds(datname.get(0), g).getHeight();
    int posnameY = lastY + textH;
    g.drawString(datname.get(i), posnameX, posnameY);

    int rectH = 10;
    int rectW = datnum.get(i) * datnumW;
    int posnumX = datnumX;
    int posnumY = posnameY - rectH;
    g.drawRect(posnumX, posnumY, rectW, rectH);

    lastY = posnameY + 10;
}

g.dispose();

try{
    String filename = 'images/infograph.jpg';
    ImageIO.write(bufimage, 'jpg', new File(filename));
}
catch(Exception ex){
}
}
}

```

Explanation:

In the following code, the excel file is opened for reading.

```
File file = new File('C:\Documents\NetBeansProjects\InfographicsMaker\cnratio.xls');
```

```
FileInputStream inputStream = new FileInputStream(file);
```

The worksheet is then fetched.

```
Workbook workbook = new HSSFWorkbook(inputStream);  
Sheet sheet = workbook.getSheetAt(0);
```

Next, we get the number of rows and columns that are used to store the data. Note that the title of the data set is stored in the first row.

```
int numRows = sheet.getLastRowNum();  
int numCols = sheet.getRow(1).getLastCellNum();
```

The infographic's header is stored in the variable *title*.

```
Cell cell=sheet.getRow(0).getCell(0);  
String title = cell.getStringCellValue();
```

Two *Arrays* are used to store the data. The first array (*datname*) is for the content of the first column, while the second array (*datnum*) is for the second column.

```
ArrayList<String> datname = new ArrayList();  
ArrayList<Integer> datnum = new ArrayList();
```

Have a look at the for loop. Below is the code to fetch cell content.

```
cell=sheet.getRow(i).getCell(j);
```

The X,Y position of the title is determined in the following code.

```
FontMetrics metrics = g.getFontMetrics(font);  
int posX = (imgW - metrics.stringWidth(title)) / 2;  
int posY = (int)metrics.getStringBounds(title, g).getHeight() + 10;
```

The title is then drawn.

```
g.drawString(title, posX, posY);
```

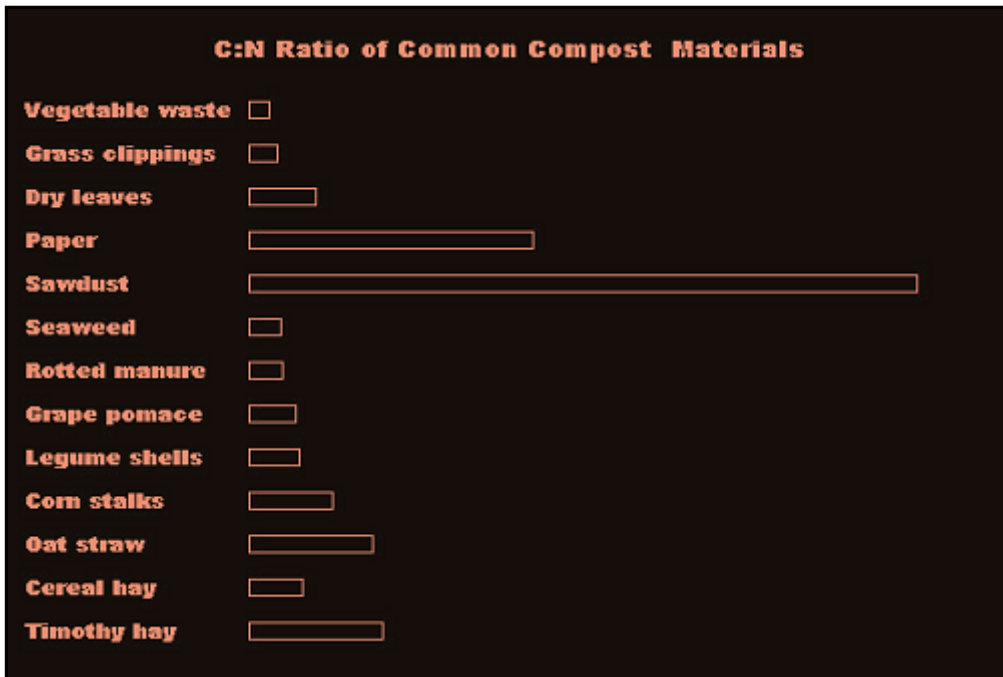
Now, it is the time to process the data. The variable *lastY* holds the the Y coordinate of the last element.

```
int lastY = posY + 20;
```

As mentioned before, the array *datnum* holds the content of the second column of the data set. The X coordinate for the data is stored in the variable *datnumX*. The variable *datnumW* is used for scaling purpose so that the bar chart can be drawn properly in respect to the image's width.

The next for loop draws the data. Data items are drawn vertically so there is no need to change the X coordinates. For each iteration, the variable *lastY* is assigned a new value based on the previous item's position (Y coordinate).

Here is the output:



advertisement

DreamHost WordPress Hosting

Reliable, lightning-fast hosting solutions specifically optimized for WordPress.

[Click here](#)

www.liberpaper.com