```
package movingWater;

import java.io.File;
import java.io.IOException;

import jxl.Workbook;
import jxl.write.*;
import jxl.write.Number;

public class MovingWater {

	public double kPaTom(double kpa) {
		return kpa * .102;
	}

	public static double staticHead(double disLevel, double WL) {

		return disLevel - WL;
	}

	public static double dynamicHead(double lossCoef, double pipeVelo, double gravity ) {
		double V = Math.pow(pipeVelo, 2);
		double DH = (lossCoef * V)/2 * gravity;
		return DH;
```

```
}

public static double velocity(double flowR, double crossSec) {

        return (flowR / 3600) / crossSec;

}

public static double crossec(double diamiter) {
        double diamater = Math.pow(diamiter, 2);
        double cossSec = (Math.PI * diamater) / 4;
        return cossSec;
}

public static double lossC(double fitting, double pipe) {
        return fitting + pipe;
}

public static double pipe(double fric, double length, double diamiter) {

        return (fric * length) / diamiter;
}

public static double fricCof(double roughfac, double reynoldNum, double diamiter) {
```

```
        double frac1 = roughfac / (3.7 * diamiter);
        double frac2 = 5.74 / Math.pow(reynoldNum, .9);
        double log = Math.log(frac1 + frac2);
        double coff = 0.25 / Math.pow(log, 2);
        return coff;
    }

    public static double Reynold(double veloc, double dimater, double kinematicV) {
        return (veloc * dimater) / kinematicV;
    }

    public static double fricHead(double fricFac, double length, double diamiter, double velo,
double gravity) {
        return ((fricFac * length)/diamiter)*(Math.pow(velo, 2)/(2 * gravity));
    }



    public static void main(String[] args) throws Exception {
        File f = new File("TDH.xls");

        WritableWorkbook myX = Workbook.createWorkbook(f);
        WritableSheet sheeet = myX.createSheet("graph", 0);
        Label L1 = new Label(0, 0, "Power");
```

```
Label L2 = new Label(1, 0, "Total Head Max");
Label L3 = new Label(2, 0, "Total Head Min");
sheeet.addCell(L1);
sheeet.addCell(L2);
sheeet.addCell(L3);

double[] lengths = new double[] {120000, 220000, 320000, 420000, 520000,
620000, 720000, 820000, 920000, 1020000, 1120000} ;
double[] elevation = new double[] {84, 36, 187, 170, 115, 122, 206, 230, 35, 367,
468};

double disLevel = 54.1104, TWL = 49.191672, BWL = 40.047672, SHmin =
staticHead(disLevel, TWL),
SHmax = staticHead(disLevel, BWL), length= 27.432, fittingVal =
0.75,
roughness = 0.000045, diamiter = 1, gravity = 9.8;
int col1 = 0, col2 = 1, col3 = 2;

int row = 1;

for(double i = 0; i < 2500; i += 10) {
double HD = dynamicHead(lossC(fittingVal,
pipe(fricCof(roughness,Reynold(velocity(i,
```

```
                                        crossec(diamiter)), diamiter, 0.00000131),
diamiter),length,diamiter)), velocity(i,
                                                    crossec(diamiter)), gravity);
                    double HTmax = SHmax + HD;
                    double HTmin = SHmin + HD;

                    Number pow = new Number(col1, row, i);
                    Number max = new Number(col2, row, HTmax);
                    Number min = new Number(col3, row, HTmin);
                    row++;

                    System.out.println("Max Total Head: "+ HTmax );
                    System.out.println("Minimum Total Head: "+ HTmin);

                    sheeet.addCell(pow);
                    sheeet.addCell(max);
                    sheeet.addCell(min);

                }

                for(int l = 0; l < lengths.length; l++) {
                    col1 += 4;
                    col2 += 4;
                    col3 += 4;
```

```
                    row = 1;

                    for(double i = 0; i < 2500; i += 10) {
                        double HF = fricHead(roughness, lengths[l], diamiter, velocity(i,
crossec(diamiter)), gravity);
                        double Tpres = elevation[l] + HF;

                        Number pow = new Number(col1, row, i);
                        Number max = new Number(col2, row, Tpres);
                        row++;

                        System.out.println("Total pressure: "+ Tpres );

                        sheeet.addCell(pow);
                        sheeet.addCell(max);

                    }
                }
                myX.write();
                myX.close();
            }

}
```