

```
//Title:      HW 2
//Author:     Furrukh Khan
//Company:    OSU
```

```
package circuit;
import linearEq.*;
```

```
public class CircuitClient {
```

```
    public static void main(String[] args) throws SolveLinearEqException{
```

```
        EComponent[] cir = new EComponent[6];
```

```
        /* Think of a loop and a current flowing in this loop in a given reference direction.
        * Add a voltage source with a negative or positive
        * reference polarity, depending on whether the source is "in the direction" of
        * the loop current, or in the "opposite direction". If you add a resistance
        * with a positive reference polarity, then the voltage across it has the
        * same polarity direction as the reference direction of the loop current.
        */
```

```
        SeriesCircuit serCircuit = new SeriesCircuit();
```

```
        EComponent vs1 = new VoltageSource(4.0D);
        vs1.setPolarity(-1);
        vs1.setName("VS1");
        serCircuit.addEComponent(vs1);
```

```
        EComponent r1 = new Resistor(2.0D);
        r1.setName("R1");
        serCircuit.addEComponent(r1);
```

```
        EComponent r2 = new Resistor(6.0D);
        r2.setName("R2");
        serCircuit.addEComponent(r2);
```

```
        EComponent vs2 = new VoltageSource(2.0D);
        vs2.setPolarity(1);
        vs2.setName("VS2");
        serCircuit.addEComponent(vs2);
```

```
        EComponent r3 = new Resistor(2.0D);
        r3.setName("R3");
        serCircuit.addEComponent(r3);
```

```
        EComponent r4 = new Resistor(10.0D);
        r4.setName("R4");
        serCircuit.addEComponent(r4);
```

```
        serCircuit.solve();
        System.out.println(serCircuit);
```

```
    }
```

```
}
```

```
/* --- OUTPUT ---
```

```
Voltage Source: VS1, -4.0 Volts , 0.1 Amps
Resistor: R1, 0.2 Volts , 0.1 Amps
Resistor: R2, 0.6000000000000001 Volts , 0.1 Amps
Voltage Source: VS2, 2.0 Volts , 0.1 Amps
Resistor: R3, 0.2 Volts , 0.1 Amps
Resistor: R4, 1.0 Volts , 0.1 Amps
```

```
*/
```

```
//Title:      HW 2
//Author:     Furrukh Khan
//Company:    OSU
```

```
package circuit;
import linearEq.*;

public interface Circuit {

    public void solve() throws SolveLinearEqException;
}
```

```
//Title:      HW 2
//Author:     Furrukh Khan
//Company:    OSU
```

```
package circuit;
```

```
import java.util.*;
import linearEq.*;
```

```
public class SeriesCircuit implements Circuit{
```

```
    Vector eComponents = new Vector();
```

```
    public void addEComponent(EComponent eComp) {
        eComponents.addElement(eComp);
    }
```

```
    public void solve() throws SolveLinearEqException {
```

```
        LinearExp exp = ((EComponent)eComponents.elementAt(0)).getVoltageExp();
        for(int i = 1; i < eComponents.size(); ++i) {
            exp = exp.add(((EComponent)eComponents.elementAt(i)).getVoltageExp());
        }
```

```
        SysLinearEq1by1 sys = new SysLinearEq1by1(exp);
```

```
        double current = sys.solve();
```

```
        for(int i = 0; i < eComponents.size(); ++i) {
            ((EComponent)eComponents.elementAt(i)).setCurrent(current);
        }
```

```
    }
```

```
    public String toString() {
```

```
        String str = "";
```

```
        for(int i = 0; i < eComponents.size(); ++i) {
```

```
            str = str +
```

```
                ((EComponent)eComponents.elementAt(i)).getType() + ": " +
```

```
                ((EComponent)eComponents.elementAt(i)).getName() + ", "+
```

```
                ((EComponent)eComponents.elementAt(i)).getVoltage()+" Volts " + ", " +
```

```
                ((EComponent)eComponents.elementAt(i)).getCurrent()+" Amps "+"\\n";
```

```
        }
```

```
        return str;
```

```
    }
```

```
}
```

```
//Title:      HW 2
//Author:     Furrukh Khan
//Company:    OSU
```

```
package circuit;
import linearEq.*;

public interface EComponent {

    public LinearExp getVoltageExp();
    public double getVoltage();
    public double getCurrent();
    public void setVoltage(double vol);
    public void setCurrent(double curr);
    public void setPolarity(int pol);
    public void setName(String name);
    public String getName();
    public String getType();
}
```

```
//Title:      HW 2
//Author:     Furrukh Khan
//Company:    OSU
```

```
package circuit;
```

```
public interface Source extends EComponent {
}
```

```
//Title:      HW 2
//Author:     Furrukh Khan
//Company:    OSU
```

```
package circuit;
import linearEq.*;

public class Resistor implements EComponent {

    private double voltage = 0.0D;
    private double current = 0.0D;
    private double resistance = 0.0D;

    private String name = null;
    private static String type = "Resistor";

    private int polarity = 1;

    public Resistor(double res) {
        resistance = res;
    }

    public void setName(String name) {
        this.name = name;
    }
    public String getName() {
        return name;
    }
    public String getType() {
        return type;
    }

    public void setPolarity(int pol) {
        if(pol >= 0 ) { polarity = 1; }
        else { polarity = -1; }
        resistance = resistance * polarity ;
    }

    public void setResistance(double res) {
        resistance = res;
    }

    public LinearExp getVoltageExp() {
        return new LinearExp(0.0D, 0.0D, resistance);
    }

    public LinearExp getCurrentExp() {
        return new LinearExp(0.0D, 1.0D/resistance, 0.0D);
    }

    public double getVoltage() {
        return voltage;
    }

    public double getCurrent() {
        return current;
    }

    public void setVoltage(double vol) {
        voltage = vol;
        current = vol/resistance;
    }

    public void setCurrent(double curr) {
        current = curr;
        voltage = curr*resistance;
    }
}
```

```
//Title:      HW 2
//Author:     Furrukh Khan
//Company:    OSU
```

```
package circuit;
import linearEq.*;

public class VoltageSource implements Source {
    private double voltage = 0.0D;
    private double current = 0.0D;

    private int polarity = 1;

    private String name = null;
    private static String type = "Voltage Source";

    public VoltageSource(double vol) {
        voltage = vol;
    }

    public void setPolarity(int pol) {
        if(pol >= 0 ) { polarity = 1; }
        else { polarity = -1; }
        voltage = Math.abs(voltage) * polarity ;
    }

    public void setName(String name) {
        this.name = name;
    }
    public String getName() {
        return name;
    }
    public String getType() {
        return type;
    }

    public LinearExp getVoltageExp() {
        return new LinearExp(voltage, 0.0D, 0.0D);
    }

    public LinearExp getCurrentExp() {
        return null;
    }

    public double getVoltage() {
        return voltage;
    }

    public double getCurrent() {
        return current;
    }

    public void setVoltage(double vol) {

    }

    public void setCurrent(double curr) {
        current = curr;
    }
}
```

```
//Title:      HW 2, 694T
//Author:     Furrukh Khan
//Company:    OSU
```

```
package linearEq;
```

```
public class LinEqClient {
```

```
    public static void main(String[] args) throws SolveLinearEqException{
```

```
        LinearExp exp1 = new LinearExp(2.0D, 3.0D, 0.0D); //exp1 = 2.0 + 3.0 x+0.0 y
        System.out.println(exp1);
```

```
        exp1.replaceByValue(1, 5.0D); // in exp1 replace x by value 5.0
        System.out.println(exp1); // exp1 = 17.0 + 0.0 x + 0.0 y
```

```
        LinearExp exp2 = new LinearExp(0.0D, 4.0D, 0.0D); // 0.0 + 4.0 x + 0.0 y
        LinearExp exp3 = exp1.add(exp2); // add exp2 to exp1
        System.out.println(exp3); // 17.0 + 4.0 x + 0.0 y
```

```
        SysLinearEq1by1 sys = new SysLinearEq1by1(exp3);
        double sol = sys.solve();
        System.out.println(sol); // sol: x = -17.0/4.0
```

```
        LinearExp exp4 = new LinearExp(2.0D, 0.0D, 4.0D); // exp4 = 2.0 + 0.0 x+4.0 y
        sys = new SysLinearEq1by1(exp4);
        sol = sys.solve();
        System.out.println(sol); // sol: y = -2.0/4.0
```

```
        LinearExp exp5 = new LinearExp(2.0D, 2.0D, 4.0D); // exp4 = 2.0 + 2.0 x+4.0 y
        sys = new SysLinearEq1by1(exp5);
        // sol = sys.solve(); throws exception
```

```
        LinearExp exp6 = new LinearExp(2.0D, 0.0D, 0.0D); // exp4 = 2.0 + 0.0 x+0.0 y
        sys = new SysLinearEq1by1(exp6);
        // sol = sys.solve(); throws exception
```

```
    }
}
```

```
//Title:      HW 2, 694T
//Author:     Furrukh Khan
//Company:    OSU
```

```
package linearEq;

public final class SysLinearEq1by1 {
    private LinearExp exp = null;

    public SysLinearEq1by1(LinearExp exp) {
        this.exp = exp;
    }

    public double solve() throws Solvelby1Exception {

        if( exp == null ) {
            throw new Solvelby1Exception();
        }

        double[] coeff = exp.getCoeff();

        if( coeff[1] != 0.0D && coeff[2] != 0.0D) {
            throw new Solvelby1Exception();
        }

        if( coeff[1] == 0.0D && coeff[2] == 0.0D) {
            throw new Solvelby1Exception();
        }

        if( coeff[1] == 0.0D ) {
            return -coeff[0]/coeff[2];
        }

        return -coeff[0]/coeff[1];
    }
}
```

```
//Title:      HW 2, 694T
//Author:     Furrukh Khan
//Company:    OSU
```

```
package linearEq;

public final class LinearExp {
    private double[] coeff = new double[3];

    public LinearExp(double d1, double d2, double d3) {
        coeff[0] = d1;
        coeff[1] = d2;
        coeff[2] = d3;
    }

    public double[] getCoeff() {
        return (double[]) coeff.clone();
    }

    public LinearExp add(LinearExp exp) {
        return new LinearExp(coeff[0]+exp.coeff[0],
                             coeff[1]+exp.coeff[1],
                             coeff[2]+exp.coeff[2]);
    }

    public LinearExp sub(LinearExp exp) {
        return new LinearExp(coeff[0]-exp.coeff[0],
                             coeff[1]-exp.coeff[1],
                             coeff[2]-exp.coeff[2]);
    }

    public void replaceByValue(int index, double value) {
        if(index == 1 || index == 2) {
            coeff[0] = coeff[0] + coeff[index]*value;
            coeff[index] = 0.0D;
        }
    }

    public String toString() {
        return "["+coeff[0]+", "+coeff[1]+", "+coeff[2]+"]";
    }
}
```

```
//Title:      HW 2, 694T
//Author:     Furrukh Khan
//Company:    OSU
```

```
package linearEq;
```

```
public class SolveLinearEqException extends Exception {
    public SolveLinearEqException(String str) {
        super(str);
    }
}
```

```
//Title:      HW 2, 694T
//Author:     Furrukh Khan
//Company:    OSU
```

```
package linearEq;
```

```
public class Solvelby1Exception extends SolveLinearEqException {
```

```
    public Solvelby1Exception() {
        super(" [Not enough info to solve Sys1by1]");
    }
}
```

```
}
```