

EX NO: 00

DATE:

JAVA APPLICATION FOR R-L-C PARALLEL SERIES CIRCUITS

AIM:

To develop a Java console application to find the total resistance, inductance, and capacitance of series and parallel combination of resistor, inductor, capacitors and display the result.

REQUIREMENT:

Develop a Java application to find the effective resistance, inductance, and capacitance of parallel series combination of resistor, inductor, and capacitors.

Create a class Resistor with the following

Data member: resistance,

Members functions: readValue, printValue, parallel, series, and constructors.

Create a class Inductor with the following

Data member: inductance,

Members functions: readValue, printValue, parallel, series, and constructors.

Create a class Capacitor with the following

Data member: capacitance,

Members functions: readValue, printValue, parallel, series, and constructors.

Create a class Calculation2 with static member function: main and calculate the parallel series combinations of resistor, inductor, capacitors and display the result.

ALGORITHM:

STEP 1: Declare class Resistor with resistance data member, readValue, printValue, series, parallel member functions and constructors.

STEP 2: Declare class Inductor with inductance data member, readValue, printValue, series, parallel member functions and constructors.

STEP 3: Declare class Capacitor with capacitance data member, readValue, printValue, series, parallel member functions and constructors.

STEP 4: Declare class Calculation2 with a static main function.

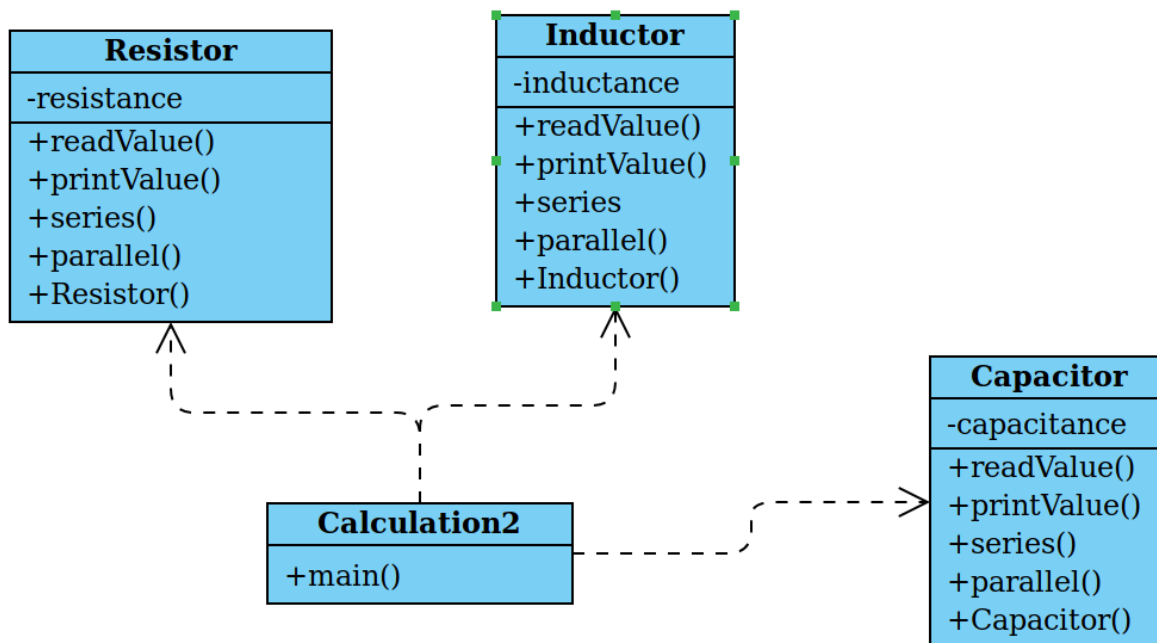
STEP 5: Create objects of type Resistor, Inductor, and Capacitor.

STEP 6: Get the input from user.

STEP 7: Calculate the parallel series combinations.

STEP 8: Display the results.

CLASS DIAGRAM



PROGRAM:

Resistor.java

```
/**
 * Program to represent resistor
 *
 * Developed by
 * C. Obed Otto,
 * Associate Professor, Saveetha Engineering College
 * obedotto@saveetha.ac.in
 */
package electricalcomponents;

import java.io.*;
import java.util.*;

/*****
 * To represent resistor
 */
```

```

* @author obedotto@saveetha.ac.in
*
*/
public class Resistor {

    /**
     * Resistance value in ohm
     */
    double resistance;

    /**
     * To initialize the resistor with initial value
     *
     * @param v value of the resistance in ohm
     */
    public Resistor(double v)
    {
        resistance=v;
    }

    /**
     * Create a resistor with zero resistance
     */
    public Resistor()
    {
        resistance=0;
    }

    /**
     * To read the resistance value from the user
     */
    void readValue()
    {
        Scanner sc=new Scanner(System.in);
        System.out.print("\nEnter the resistance value ");
        resistance=sc.nextDouble();
    }

    /**
     * To print the resistance value
     */
    void printValue()
    {

```

```

        System.out.print(" "+resistance+" ohm");
    }

    /**
     * To calculate effective resistance value of two parallel resistors
     *
     * @param v Resistor object which is to be connected in parallel
     * @return Returns the effective total resistance
     */
    Resistor parallel(Resistor v)
    {
        Resistor r;

        r=new Resistor();
        r.resistance=(v.resistance*this.resistance)/(v.resistance+this.resistance);

        return r;
    }

    /**
     * To calculate effective resistance value of two series resistors
     *
     * @param v Resistor object which is to be connected in series
     * @return Returns the effective total resistance
     */
    Resistor series(Resistor v)
    {
        Resistor r;

        r=new Resistor();
        r.resistance=v.resistance+this.resistance;

        return r;
    }
}

```

Inductor.java

```
/**
```

```
* Program to represent inductor
*
* Developed by
* C. Obed Otto,
* Associate Professor, Saveetha Engineering College
* obedotto@saveetha.ac.in
*/
package electricalcomponents;
```

```
import java.util.Scanner;
```

```

/*****
* To represent inductor
*
* @author obedotto@saveetha.ac.in
*
*/
public class Inductor {

    /**
     * Inductance value in Henry
     */
    double inductance;

    /**
     * To initialize the inductor with initial value
     *
     * @param v value of the capacitance in henry
     */
    public Inductor(double v)
    {
        inductance=v;
    }

    /**
     * Create a inductor with zero inductance
     */
    public Inductor()
    {
        inductance=0;
    }
}
```

```
}
```

```
/******
```

```
 * To read the inductance value from the user
```

```
 */
```

```
void readValue()
```

```
{
```

```
    Scanner sc=new Scanner(System.in);
```

```
    System.out.print("\nEnter the inductance value ");
```

```
    inductance=sc.nextDouble();
```

```
}
```

```
/******
```

```
 * To print the inductance value
```

```
 */
```

```
void printValue()
```

```
{
```

```
    System.out.print(" "+inductance+" henry");
```

```
}
```

```
/******
```

```
 * To calculate effective inductance value of two parallel inductors
```

```
 *
```

```
 * @param v Inductor object which is to be connected in parallel
```

```
 * @return Returns the effective total inductance
```

```
 */
```

```
Inductor parallel(Inductor v)
```

```
{
```

```
    Inductor i;
```

```
        i=new Inductor();
```

```
        i.inductance=(v.inductance*this.inductance)/(v.inductance+this.inductance);
```

```
        return i;
```

```
}
```

```
/******
```

```
 * To calculate effective inductance value of two series inductors
```

```
 *
```

```
 * @param v Inductor object which is to be connected in series
```

```
 * @return Returns the effective total inductance
```

```
 */
```

```

Inductor series(Inductor v)
{
    Inductor i;

    i=new Inductor();
    i.inductance=v.inductance+this.inductance;

    return i;
}
}

```

Capacitor.java

```

/**
 * Program to represent Capacitor
 *
 * Developed by
 * C. Obed Otto,
 * Associate Professor, Saveetha Engineering College
 * obedotto@saveetha.ac.in
 */
package electricalcomponents;
import java.util.Scanner;

/*****
 * To represent capacitor
 *
 * @author obedotto@saveetha.ac.in
 */
public class Capacitor {
    /**
     * Capacitance value in Farad
     */
    double capacitance;

    /**
     * To initialize the capacitor with initial value
     *
     * @param v value of the capacitance in farad

```



```

    */
    public Capacitor(double v)
    {
        capacitance=v;
    }

    /**
     * Create a capacitor with zero capacitance
     */
    public Capacitor()
    {
        capacitance=0;
    }

    /**
     * To read the capacitance value from the user
     */
    void readValue()
    {
        Scanner sc=new Scanner(System.in);

        System.out.print("\nEnter the capacitance value ");
        capacitance=sc.nextDouble();
    }

    /**
     * To print the capacitance value
     */
    void printValue()
    {
        System.out.print(" "+capacitance+" farad");
    }

    /**
     * To calculate effective capacitance value of two parallel capacitors
     *
     * @param v Capacitor object which is to be connected in parallel
     * @return Returns the effective total capacitance
     */
    Capacitor parallel(Capacitor v)
    {
        Capacitor c;
    }

```

```

        c=new Capacitor();
        c.capacitance=v.capacitance+this.capacitance;

        return c;
    }

    /**
     * To calculate effective capacitance value of two series capacitors
     *
     * @param v Capacitor object which is to be connected in series
     * @return Returns the effective total capacitance
     */
    Capacitor series(Capacitor v)
    {
        Capacitor c;

        c=new Capacitor();
        c.capacitance=(v.capacitance*this.capacitance)/(v.capacitance+this.capacitance);

        return c;
    }
}

```

Calculation2.java

```

/**
 * Program to calculate the effective resistance, inductance and capacitance values
 *
 * Developed by
 * C. Obed Otto,
 * Associate Professor, Saveetha Engineering College
 * obedotto@saveetha.ac.in
 */
package electricalcomponents;

/**
 * To perform series, parallel RLC calculation
 *
 * @author obedotto@saveetha.ac.in
 */
public class Calculation2 {

```

```

/*****
 * Program Entry Point
 * @param args command line parameters
 */
public static void main(String[] args) {
    Resistor r1,r2,r3,r4;

    r1=new Resistor();
    r1.readValue();

    r2=new Resistor();
    r2.readValue();

    r3=r1.parallel(r2);
    System.out.print("Effective parallel resistance:");
    r3.printValue();

    r4=r1.series(r2);
    System.out.print("\nEffective series resistance:");
    r4.printValue();
    }
}

```

OUTPUT:

Enter the resistance value 100

Enter the resistance value 100

Effective parallel resistance: 50.0 ohm

Effective series resistance: 200.0 ohm

RESULT:

Thus a Java console application is developed to find the overall resistance, inductance, and capacitance of parallel, series resistor, inductor, and capacitors.