

```
using System;
using System.Collections.Generic;
using System.Text;

namespace LEDCircuitSim2
{
    public class LEDCircuit : Circuit
    {
        // Attributes
        LED itsLED ;
        bool itsOverMax ;

        // Constructors
        public LEDCircuit( ) : base( )
        {
            itsLED = null ;
            itsOverMax = false ;
        }

        public LEDCircuit( LED theLED ) : base( )
        {
            itsLED = theLED ;
            itsOverMax = false ;
        }

        public LEDCircuit( LED theLED, double resistance,
            double initialVoltage ) :
            base( initialVoltage, resistance )
        {
            itsLED = theLED ;
            itsOverMax = false ;
        }

        public bool isLEDOn( )
        {
            return itsLED.isOn(voltage());
        }

        public override void calc( )
        {
            if ( itsLED == null )
            {
                throw new Circuit.CircuitException(
                    "No LED is selected"
                ) ;
            }

            double R = resistanceInOhms( ) ;
            if ( R <= 0.0 )
            {
                throw new Circuit.CircuitException(
                    "Invalid resistance", R
                ) ;
            }

            if ( isLEDOn( ) )
            {
                itsCurrent = Circuit.calcCurrent(
                    itsVoltage - itsLED.Vfwd( ), R
                );

                itsPower = Circuit.calcPower( itsCurrent, R );

                itsOverMax =
                    itsLED.isCurrentMax( itsVoltage, R ) ||
                    itsLED.isPowerMax( itsCurrent ) ;
            }
            else

```

```
        {
            itsCurrent = 0.0 ;
            itsPower = 0.0 ;
            itsOverMax = false ;
        }

// Accessors
public bool isLEDOverLimit( )      { return itsOverMax ; }
public LED getLED( )              { return itsLED ; }

// Mutilator
public void setLED( LED theLED )  { itsLED = theLED ; }
}
}
```