Educational Offering

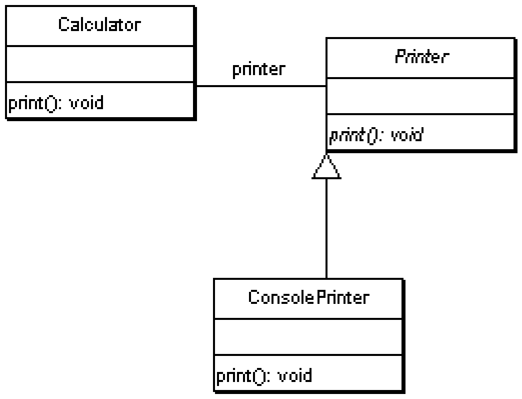
Design Pattern Primer

Exercise: Decorator

Overview
In this exercise, you will apply the Decorator pattern to the calculator implementation.

User Requirement
You need to utilize Java compiler and byte code interpreter to create and compile classes, methods, and fields.

Introduction
The Decorator patterns intent is to support adding disparate functionality to an existing implementation, dynamically without using inheritance. This exercise will abstract the printing function of the calculator implementation setting the stage for different types of printing capabilities. By default printing functionality will be console based. UML is shown below.



Additional functionality will be applied to by applying the Decorator pattern to Print abstraction so that an open-ended number of "decorating" behavior can be applied in a dynamic manner.

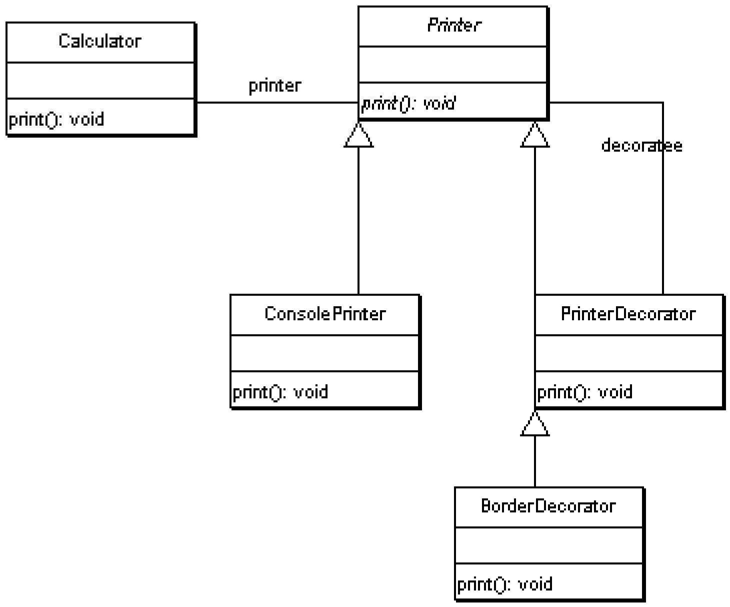
Source for this exercise is defined in the **db.lab.decorator** package.

Exercise Instructions
1. Print design
In the ***db.lab*.decorator** package execute the Tester classes main method. Study the *Printer* class hierarchy and the role it plays in the Calculator's *print()* method implementation.

Copyright © 2014 Keyhole Software, LLC. All rights reserved. Tel 877-521-7769 https://keyholesoftware.com

2. Implement the Decorator

Additional printing functionality can be applied using the Decorator pattern. This is accomplished by extending the abstract *Printer* with a *PrinterDecorator* class. Decorating classes are created with a reference to another *Printer* instance, the decorated instance. UML for this design is shown below, along with the class implementation for a *BorderDecorator* class. Implement this design.



```
public class BorderDecorator extends DecoratedPrinter {


      public BorderDecorator(Printer d) {
            super(d);
      }

      public void print(Calculator calc) {

            // output border lines

            System.out.println("=====================");
            getDecoratee().print(calc);
            System.out.println("=====================");

      }

}
```

3. Apply Decorator

Add the highlighted code shown to the Test classes main method. These expressions decorate the calculators *ConsolePrinter* with *BroderDecorator*. Execute the main method; basic calculator results displayed to the console should now be bordered.

Copyright © 2014 Keyhole Software, LLC. All rights reserved. Tel 877-521-7769 https://keyholesoftware.com

```
public static void main(String[] args) {
            // basic calculator

            System.out.println("* * Basic Calculator * *");

            Calculator calc = PrototypeFactory.basic();
            // install operations

            // Create Printer
            Printer printer = new ConsolePrinter();
            // Decorate Printer with Borders
            printer = new BorderDecorator(printer);

            calc.setPrinter( printer );


            calc.execute("+", 10.0);
            calc.print();
            calc.execute("/", 2.0);
            calc.print();
            // Scientific Calculator

            calc = PrototypeFactory.scientific();

            System.out.println("* * Scientific Calculator * *");

            //calc.log(10.0)
            calc.execute("+", 10.0);
            calc.execute("sin", 20.0);
            calc.print();
            calc.execute("log", 100);
            calc.print();

            // Programmer Calculator

            calc = PrototypeFactory.programmer();

            System.out.println("* * Programmers Calculator * *");

            //calc.log(10.0)
            calc.execute("+", 10.0);
            calc.execute("+", 20.0);
            calc.execute("bin");
            calc.print();

      }
```

4. Create a Date and Time Decorator

Create two more decorator classes, using the *BorderDecorator* as a guide. To test, modify the Tester main method so that these new Decorator classes decorate the current *BorderDecorator*. The results should therefore output border, date, time, and the current calculation.

Copyright © 2014 Keyhole Software, LLC. All rights reserved. Tel 877-521-7769 https://keyholesoftware.com