**package** DecoratorPattern;

**public** **interface** Notifier {

**void** send(String message);

}

**package** DecoratorPattern;

**public** **abstract** **class** NotifierDecorator **implements** Notifier {

**protected** Notifier notifier;

**public** NotifierDecorator(Notifier notifier) {

**this**.notifier = notifier;

}

**public** **void** send(String message) {

notifier.send(message);

}

}

**package** DecoratorPattern;

**public** **class** EmailNotifier **implements** Notifier {

**public** **void** send(String message) {

System.***out***.println("Sending Email: " + message);

}

}

**package** DecoratorPattern;

**public** **class** SMSNotifierDecorator **extends** NotifierDecorator {

**public** SMSNotifierDecorator(Notifier notifier) {

**super**(notifier);

}

**public** **void** send(String message) {

**super**.send(message);

sendSMS(message);

}

**private** **void** sendSMS(String message) {

System.***out***.println("Sending SMS: " + message);

}

}  
  
**package** DecoratorPattern;

**public** **class** SlackNotifierDecorator **extends** NotifierDecorator {

**public** SlackNotifierDecorator(Notifier notifier) {

**super**(notifier);

}

**public** **void** send(String message) {

**super**.send(message);

sendSlack(message);

}

**private** **void** sendSlack(String message) {

System.***out***.println("Sending Slack Message: " + message);

}

}  
  
**package** DecoratorPattern;

**public** **class** Main {

**public** **static** **void** main(String[] args) {

Notifier notifier = **new** EmailNotifier();

Notifier multiChannelNotifier = **new** SlackNotifierDecorator(

**new** SMSNotifierDecorator(

notifier));

multiChannelNotifier.send("System maintenance at 10 PM tonight.");

}

}  
  
**OUTPUT:**

