

R5.06.Maintenance applicative
Exercice de refactoring « Superclass to Delegate »

Soient les classes suivantes :

```
// Superclass
class Printer {
    public void print(String message) {
        this.whoAmI();
        System.out.println(message);
    }
    protected void whoAmI() {
        System.out.println("Je suis un objet de la classe Printer");
    }
}

// Subclass
class LoggingPrinter extends Printer {
    @Override
    public void print(String message) {
        // Add logging before printing
        System.out.println("[LOG] Printing: " + message);
        super.print(message); // Calls the superclass print method
    }
    @Override
    protected void whoAmI() {
        System.out.println("Je suis un objet de la classe LoggingPrinter");
    }
}

// Main class
public class Main {
    public static void main(String[] args) {
        Printer printer = new LoggingPrinter();
        printer.print("Hello, World!");
    }
}
```

Nous aimerions supprimer la relation d'héritage et la remplacer par une relation de composition/délégation (dans l'une des classes, on met un attribut de type l'autre classe). En d'autres termes, nous aimerions qu'un objet LoggingPrinter soit composé d'un Printer et non que sa classe soit une spécialisation de Printer. Quel sera le résultat ?