

Exercice - Gestion de commandes : le pattern « State »

Une commande n'est pas systématiquement payée à son enregistrement. Cependant, elle ne peut être livrée qu'après paiement. Les commandes ne sont plus remboursables après livraison.

Ces règles de gestion peuvent évoluer dans le futur (par exemple, la livraison peut devenir possible même sans paiement et le remboursement peut être possible jusqu'à deux semaines après livraison).

La classe suivante a été proposée pour répondre à ce cahier des charges :



```
public class Commande {  
    private int numero;  
    private Date dateLiv = null;  
    private boolean payee = false;  
    public void payer() {this.payee = true}  
    public void livrer(){  
        if (! this.payee)  
            throw new CommandeNonLivvableEx();  
        this.dateLiv = new Date();  
    }  
    public void rembourser() {  
        if (! this.payee || this.dateLiv != null)  
            throw new CommandeNonRemboursableEx ();  
        this.payee = false;  
    }  
}
```

Travail demandé

1. Faire une analyse de maintenabilité de l'application proposée.
2. Proposer un diagramme de classes à base du pattern « State » afin d'améliorer la maintenabilité de cette application.
3. Coder les classes en Java.
4. Valider votre solution par des tests automatiques.