



$$\frac{4 \cdot 1}{4 \cdot 2} + \frac{3 \cdot 2}{4 \cdot 2}$$

①

```
public Fraction add(Fraction f2) {  
    Fraction res = new Fraction((this.num * f.denom)  
                                + (f2.num * this.denom),  
                                this.denom * f2.denom);  
}
```

②

```
public class Program {  
    public static void main(String[] args) {  
        Fraction f1 = new Fraction(1, 2);  
        Fraction f2 = new Fraction(3, 4);  
        Fraction f3 = f1.add(f2);  
        System.out.print(f3.toString + " = " + "10/8");  
    }  
}
```

③ 1. class OptionA<T> implements Option<T> {

private boolean is-def;

private T value;

public OptionA<T>() {

this.is-def = false;

}

public OptionA<T>(T val) {

this.is-def = true;

this.value = val;

}

public T getOrElse(T t) {

if (this.is-def) {

return this.value;

}

else { return t; }

}

public boolean isDefined() {

return this.is-def;

}

}

```

2. abstract class OptionB<T>{
    public abstract boolean isDefined();
    public abstract T getOrElse(T t);
}

class QuelqueChose<T> extends OptionB<T>{
    private T value;
    public QuelqueChose<T>(T val){
        this.value = val;
    }

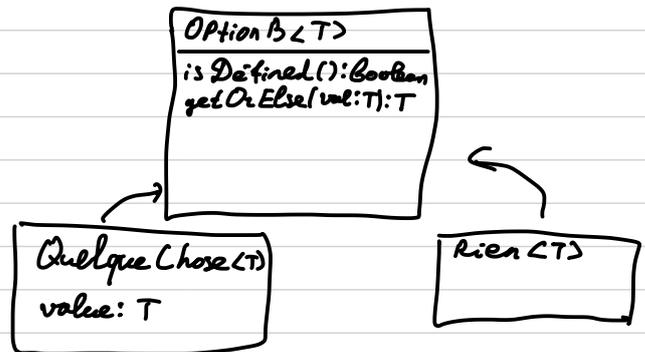
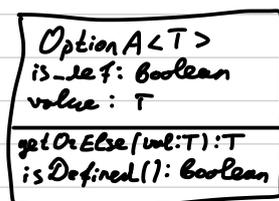
    public boolean isDefined(){ return true;}
    public T getOrElse(T val){ return this.value;}
}

class Rien<T> extends OptionB<T>{

    public Rien<T>(){}
    public boolean isDefined(){ return false;}
    public T getOrElse(T val){ return val;}
}

```

4



⑤ 1. Oui, il est un acteur

2. client

caissier

Centre d'autorisation multi-banque

Système de

3.

1. enregistrer id → afficher le prix et libellé

2. initialiser la caisse

3. présenter coupon

4. signaler la fin

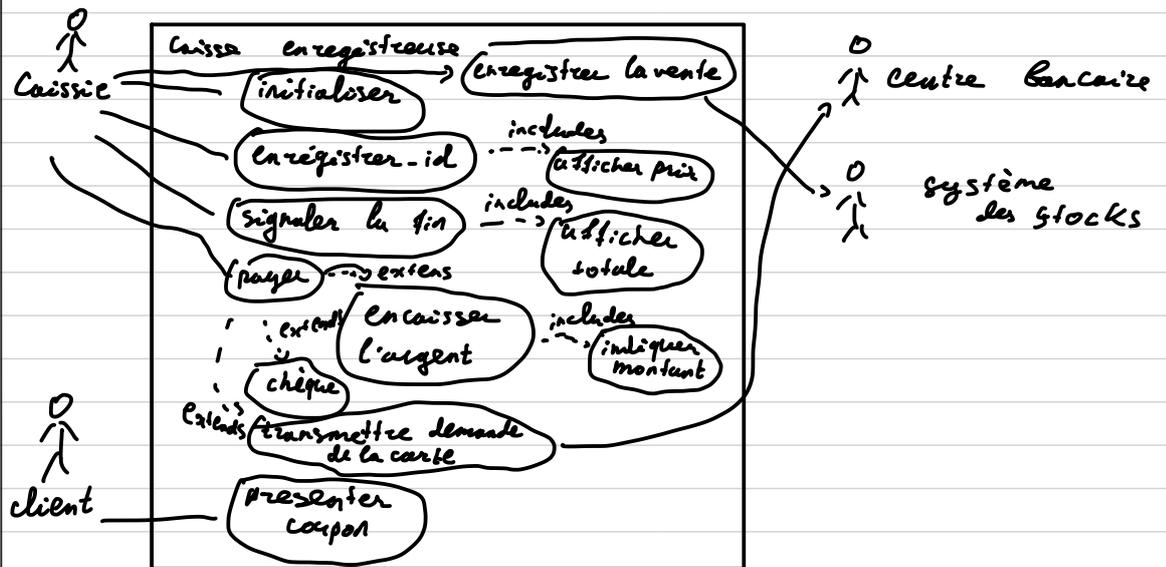
5. afficher total

6. encaisser l'argent → indiquer montant éventuel

7. enregistrer montant sur la cheque

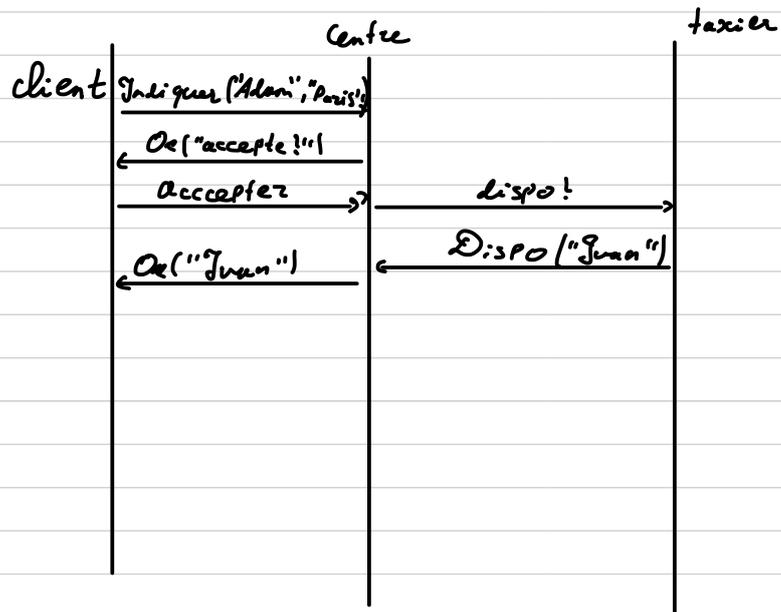
8. transmettre demande

9. enregistrer la vente et imprimer ticket



6

7



8