



Les classes abstraites, c'est fini !

(et c'est la faute à TDD)

Dora l'exploratrice



<http://www.nickelodeonjunior.fr/dora-l-exploratrice/>

Dorra Bartaguiz

Dev / Craft coach

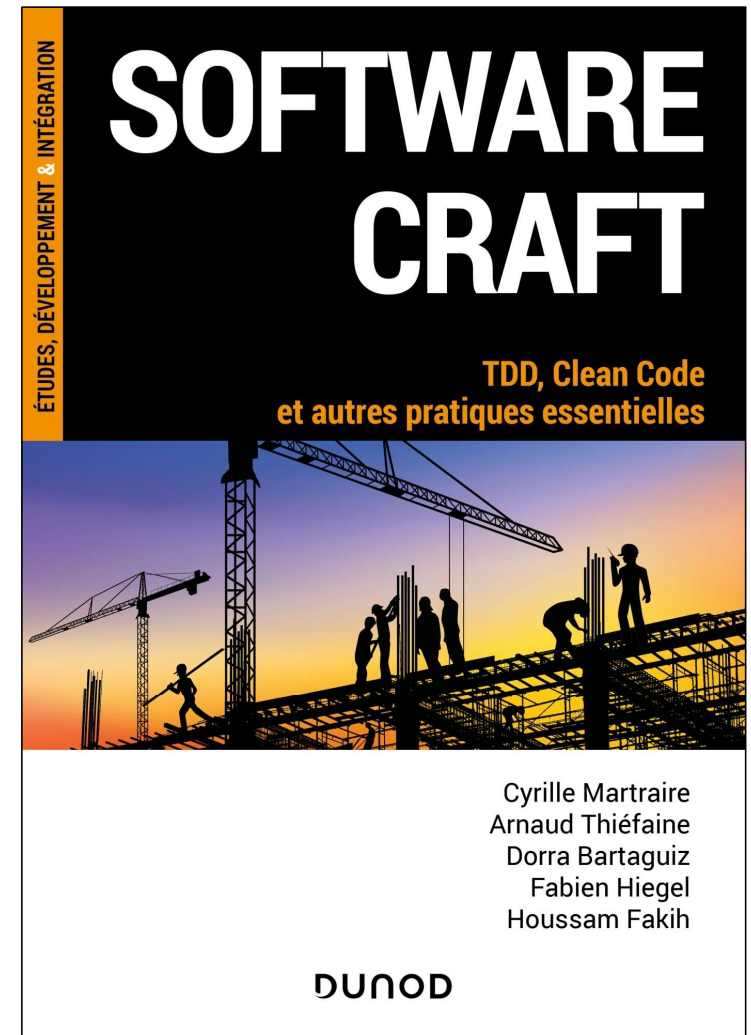
Co-auteure & illustratrice

 @DorraBartaguiz

 iAmDorra



@DorraBartaguiz



Dorra Bartaguiz

Rédactrice en chef du numéro 250

100 % féminin

Janvier 2022

 @DorraBartaguiz

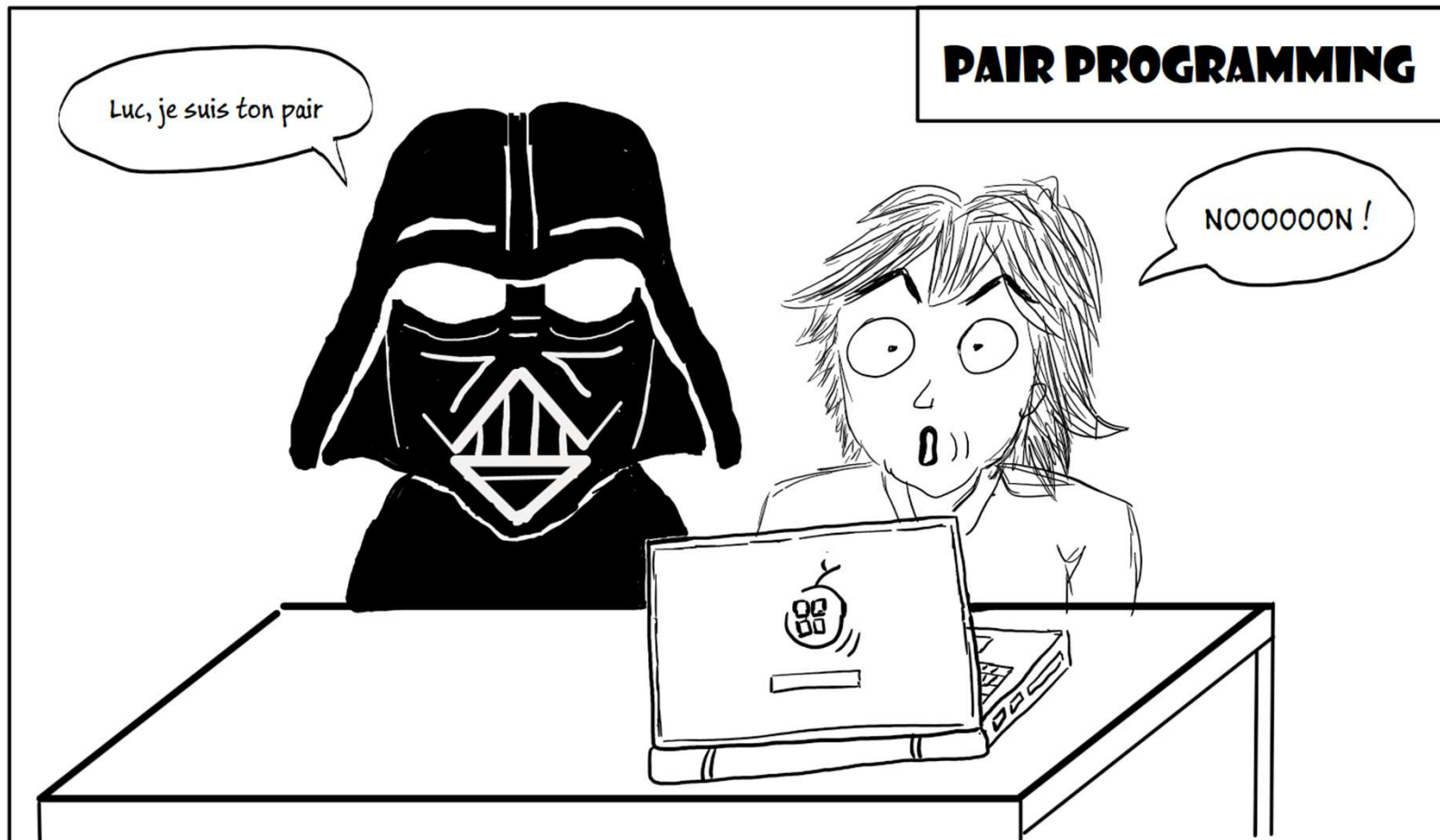
 iAmDorra



@DorraBartaguiz



Discussion



@DorraBartaguiz

Livre « Software Craft » - Edition Dunod

Besoin métier

Contexte

- Chaîne de garagistes (France & Luxembourg)
- Calcul total d'une prestation à partir d'un montant hors taxe
- Calcul de la taxe en fonction du pays
 - France 19.6%
 - Luxembourg 15% sur le matériel & 12% sur le service

Exemple

- En France
 - 150€ HT & 19.6% taxe
 - Total : 179,4€ TTC
- Au Luxembourg
 - 100€ HT de matériel & 15% taxe de matériel => total matériel : 115€
 - 50€ HT de service & 12% taxe de service => total service : 56€
 - Total : 171€

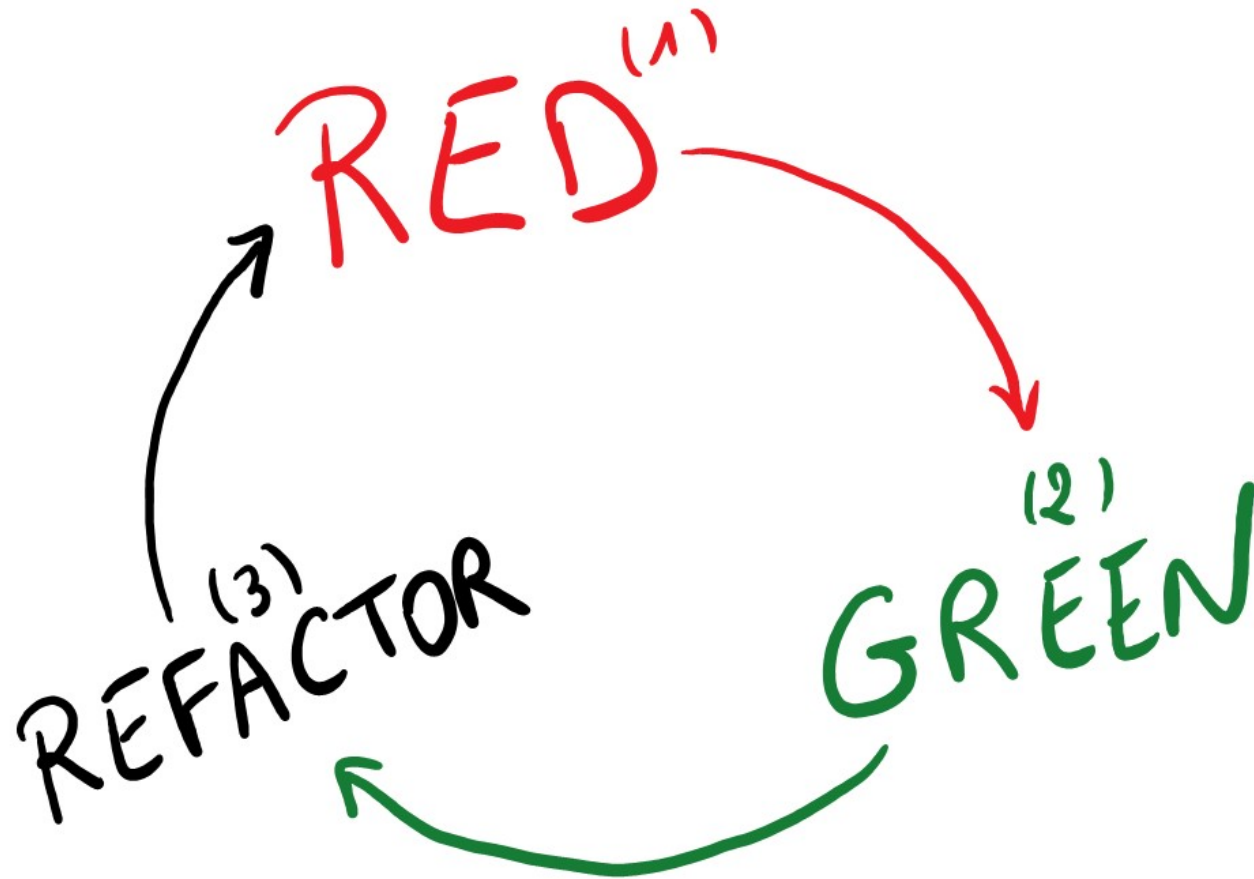
TDD

Test Driven Development

TDD

- Kent Beck – 1990
- Ecrire le code de test avant le code de production
- Fixer l'objectif du besoin
- Définir à l'avance le comportement attendu

TDD



Besoin du garagiste

- En France
 - 150€ HT & 19.6% taxe
 - Total : 179,4€ TTC
- Au Luxembourg
 - 100€ HT de matériel & 15% taxe de matériel => total matériel : 115€
 - 50€ HT de service & 12% taxe de service => total service : 56€
 - Total : 171€

Disclaimer

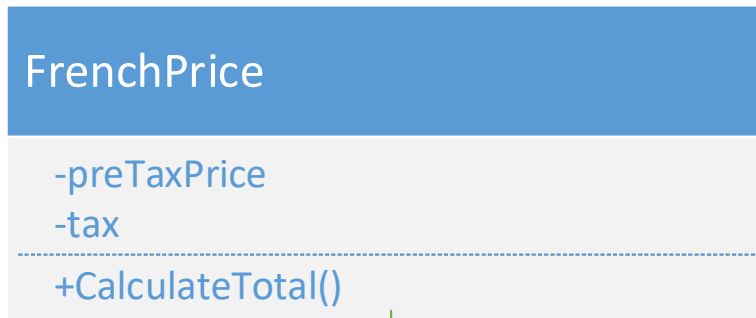
Garagiste en France

- 150€ HT & 19.6% taxe
- Total : 179,4€ TTC

```
Calculate_total_price_including_tax_for_french_mechanic()
{
    var frenchTax = 0.196;
    var preTaxPrice = 150;
    var frenchPrice = new FrenchPrice(frenchTax, preTaxPrice);
    var total = frenchPrice.CalculateTotal();
    Check.That(total).IsEqualTo(179.4);
}
```

Garagiste en France

- 150€ HT & 19.6% taxe
- => Total : 179,4€ TTC



```
public double CalculateTotal()
{
    return preTaxPrice + preTaxPrice * tax;
}
```

Garagiste au Luxembourg

- 100€ HT de matériel & 15% taxe de matériel => total matériel : 115€
- 50€ HT de service & 12% taxe de service => total service : 56€
- Total : 171€

```
Calculate_total_price_including_tax_for_luxembourgish_mechanic()  
{  
    var preTaxServiceAmount = 50;  
    var serviceTax = 0.12;  
    var preTaxMaterialAmount = 100;  
    var materialTax = 0.15;  
    var luxembourgishPrice = new LuxembourgishPrice(preTaxServiceAmount, serviceTax,  
preTaxMaterialAmount, materialTax);  
    var total = luxembourgishPrice.CalculateTotal();  
    Check.That(total).IsEqualTo(171);  
}
```


Garagiste au Luxembourg

- 100€ HT de matériel
- 15% taxe de matériel
=> total matériel : 115€

- 50€ HT de service
- 12% taxe de service
=> total service : 56€

=> Total : 171€

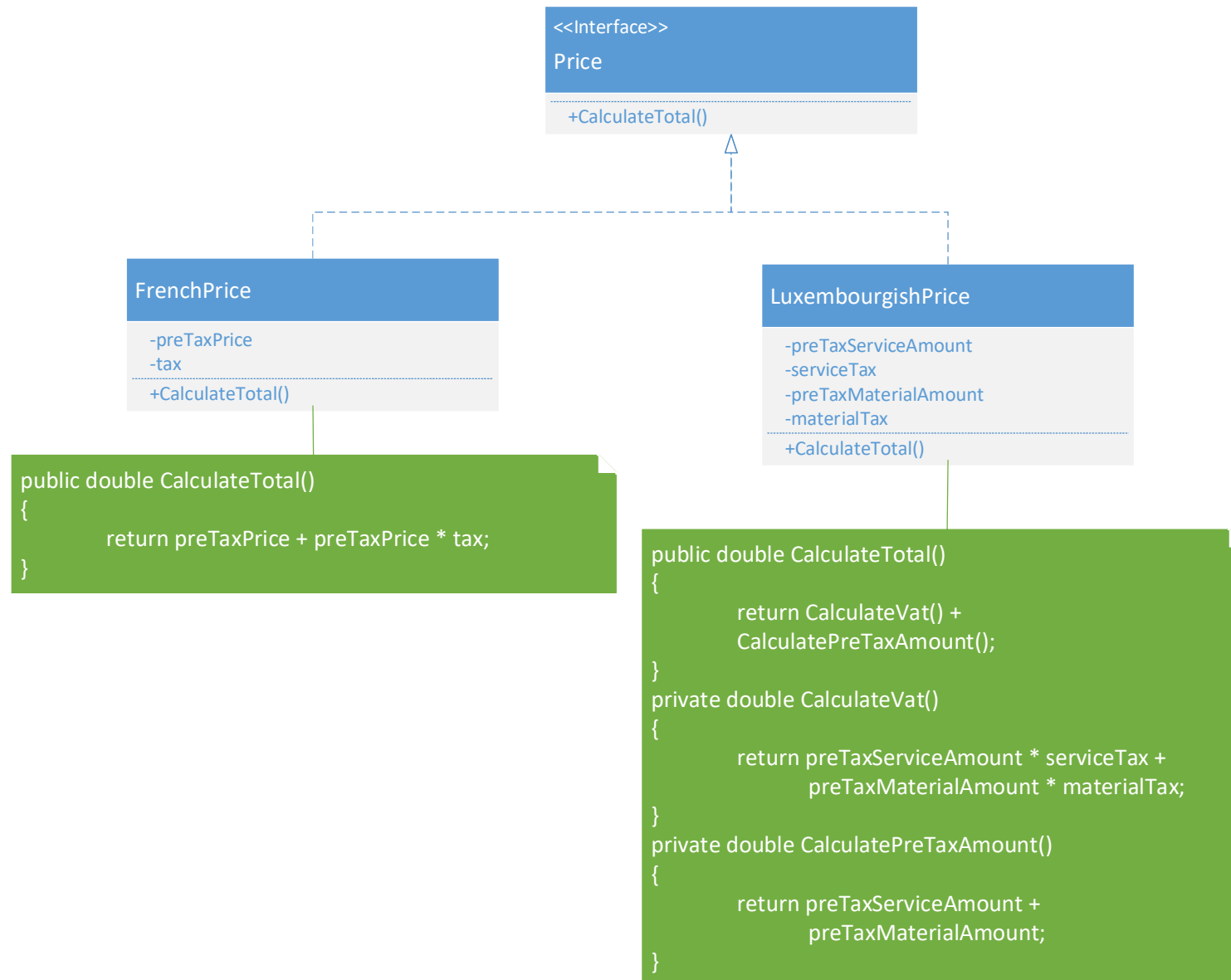
@DorraBartaguiz

LuxembourgishPrice

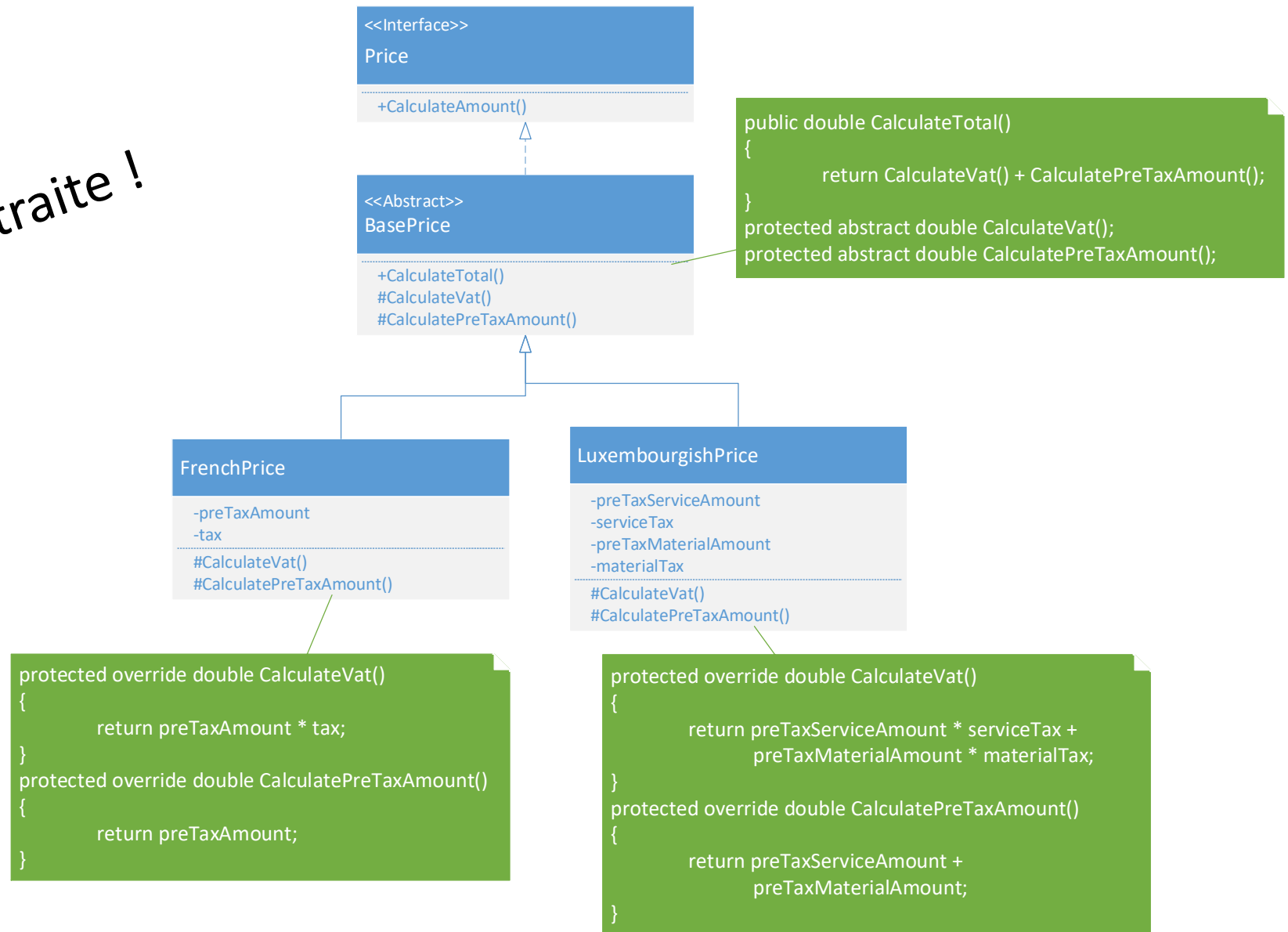
-preTaxServiceAmount
-serviceTax
-preTaxMaterialAmount
-materialTax

+CalculateTotal()

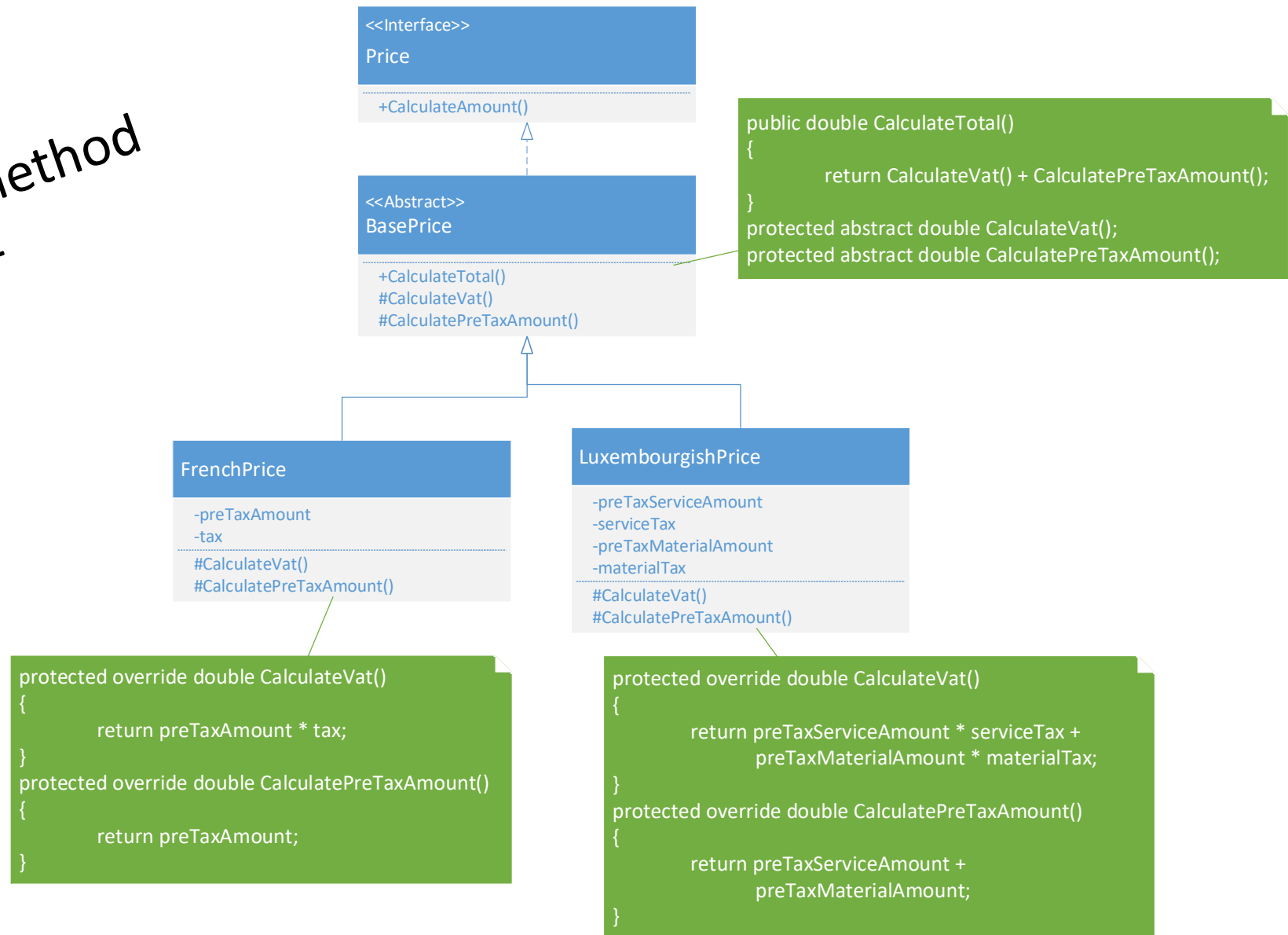
```
public double CalculateTotal()
{
    return CalculateVat() +
        CalculatePreTaxAmount();
}
private double CalculateVat()
{
    return preTaxServiceAmount * serviceTax +
        preTaxMaterialAmount * materialTax;
}
private double CalculatePreTaxAmount()
{
    return preTaxServiceAmount +
        preTaxMaterialAmount;
}
```



Classe abstraite !



Template method GoF



@DorraBartaguiz

« *Design Patterns: Elements of Reusable Object-Oriented Software* » écrit par [Erich Gamma](#), Richard Helm, [Ralph Johnson](#), et [John Vlissides](#), avec une préface de [Grady Booch](#)

Défauts du choix



Défauts du choix



```
<<Interface>>
Price
+CalculateAmount()
```

```
<<Abstract>>
BasePrice
+CalculateTotal()
#CalculateVat()
#CalculatePreTaxAmount()
```

```
public double CalculateTotal()
{
    return CalculateVat() + CalculatePreTaxAmount();
}
protected abstract double CalculateVat();
protected abstract double CalculatePreTaxAmount();
```

```
FrenchPrice
-preTaxAmount
-tax
#CalculateVat()
#CalculatePreTaxAmount()
```

```
LuxembourgishPrice
-preTaxServiceAmount
-serviceTax
-preTaxMaterialAmount
-materialTax
#CalculateVat()
#CalculatePreTaxAmount()
```

```
protected override double CalculateVat()
{
    return preTaxAmount * tax;
}
protected override double CalculatePreTaxAmount()
{
    return preTaxAmount;
}
```

```
protected override double CalculateVat()
{
    return preTaxServiceAmount * serviceTax +
        preTaxMaterialAmount * materialTax;
}
protected override double CalculatePreTaxAmount()
{
    return preTaxServiceAmount +
        preTaxMaterialAmount;
}
```

**Non
instanciable !**

Défauts du choix



```
<<Interface>>
Price
+CalculateAmount()
```

```
<<Abstract>>
BasePrice
+CalculateTotal()
#CalculateVat()
#CalculatePreTaxAmount()
```

```
public double CalculateTotal()
{
    return CalculateVat() + CalculatePreTaxAmount();
}
protected abstract double CalculateVat();
protected abstract double CalculatePreTaxAmount();
```

```
FrenchPrice
-preTaxAmount
-tax
#CalculateVat()
#CalculatePreTaxAmount()
```

```
protected override double CalculateVat()
{
    return preTaxAmount * tax;
}
protected override double CalculatePreTaxAmount()
{
    return preTaxAmount;
}
```

```
LuxembourgishPrice
-preTaxServiceAmount
-serviceTax
-preTaxMaterialAmount
-materialTax
#CalculateVat()
#CalculatePreTaxAmount()
```

```
protected override double CalculateVat()
{
    return preTaxServiceAmount * serviceTax +
        preTaxMaterialAmount * materialTax;
}
protected override double CalculatePreTaxAmount()
{
    return preTaxServiceAmount +
        preTaxMaterialAmount;
}
```

**Tester
unitairement les
sous calculs ?!**

Défauts du choix



```
<<Interface>>
Price
+CalculateAmount()
```

```
<<Abstract>>
BasePrice
+CalculateTotal()
#CalculateVat()
#CalculatePreTaxAmount()
```

```
public double CalculateTotal()
{
    return CalculateVat() + CalculatePreTaxAmount();
}
protected abstract double CalculateVat();
protected abstract double CalculatePreTaxAmount();
```

```
FrenchPrice
-preTaxAmount
-tax
#CalculateVat()
#CalculatePreTaxAmount()
```

```
protected override double CalculateVat()
{
    return preTaxAmount * tax;
}
protected override double CalculatePreTaxAmount()
{
    return preTaxAmount;
}
```

```
LuxembourgishPrice
-preTaxServiceAmount
-serviceTax
-preTaxMaterialAmount
-materialTax
#CalculateVat()
#CalculatePreTaxAmount()
```

```
protected override double CalculateVat()
{
    return preTaxServiceAmount * serviceTax +
        preTaxMaterialAmount * materialTax;
}
protected override double CalculatePreTaxAmount()
{
    return preTaxServiceAmount +
        preTaxMaterialAmount;
}
```

**Le même test
se répète !**



**Supprimer du code,
jamais de la vie !**

On le garde au cas où !

Défauts du choix



```
<<Interface>>
Price
+CalculateAmount()
```

```
<<Abstract>>
BasePrice
+CalculateTotal()
#CalculateVat()
#CalculatePreTaxAmount()
```

```
public double CalculateTotal()
{
    return CalculateVat() + CalculatePreTaxAmount();
}
protected abstract double CalculateVat();
protected abstract double CalculatePreTaxAmount();
```

```
FrenchPrice
-preTaxAmount
-tax
#CalculateVat()
#CalculatePreTaxAmount()
```

```
LuxembourgishPrice
-preTaxServiceAmount
-serviceTax
-preTaxMaterialAmount
-materialTax
#CalculateVat()
#CalculatePreTaxAmount()
```

```
protected override double CalculateVat()
{
    return preTaxAmount * tax;
}
protected override double CalculatePreTaxAmount()
{
    return preTaxAmount;
}
```

```
protected override double CalculateVat()
{
    return preTaxServiceAmount * serviceTax +
        preTaxMaterialAmount * materialTax;
}
protected override double CalculatePreTaxAmount()
{
    return preTaxServiceAmount +
        preTaxMaterialAmount;
}
```

**Le même test
se répète !**

Défauts du choix



```
<<Interface>>
Price
+CalculateAmount()
```

```
<<Abstract>>
BasePrice
+CalculateTotal()
#CalculateVat()
#CalculatePreTaxAmount()
```

```
public double CalculateTotal()
{
    return CalculateVat() + CalculatePreTaxAmount();
}
protected abstract double CalculateVat();
protected abstract double CalculatePreTaxAmount();
```

```
FrenchPrice
-preTaxAmount
-tax
#CalculateVat()
#CalculatePreTaxAmount()
```

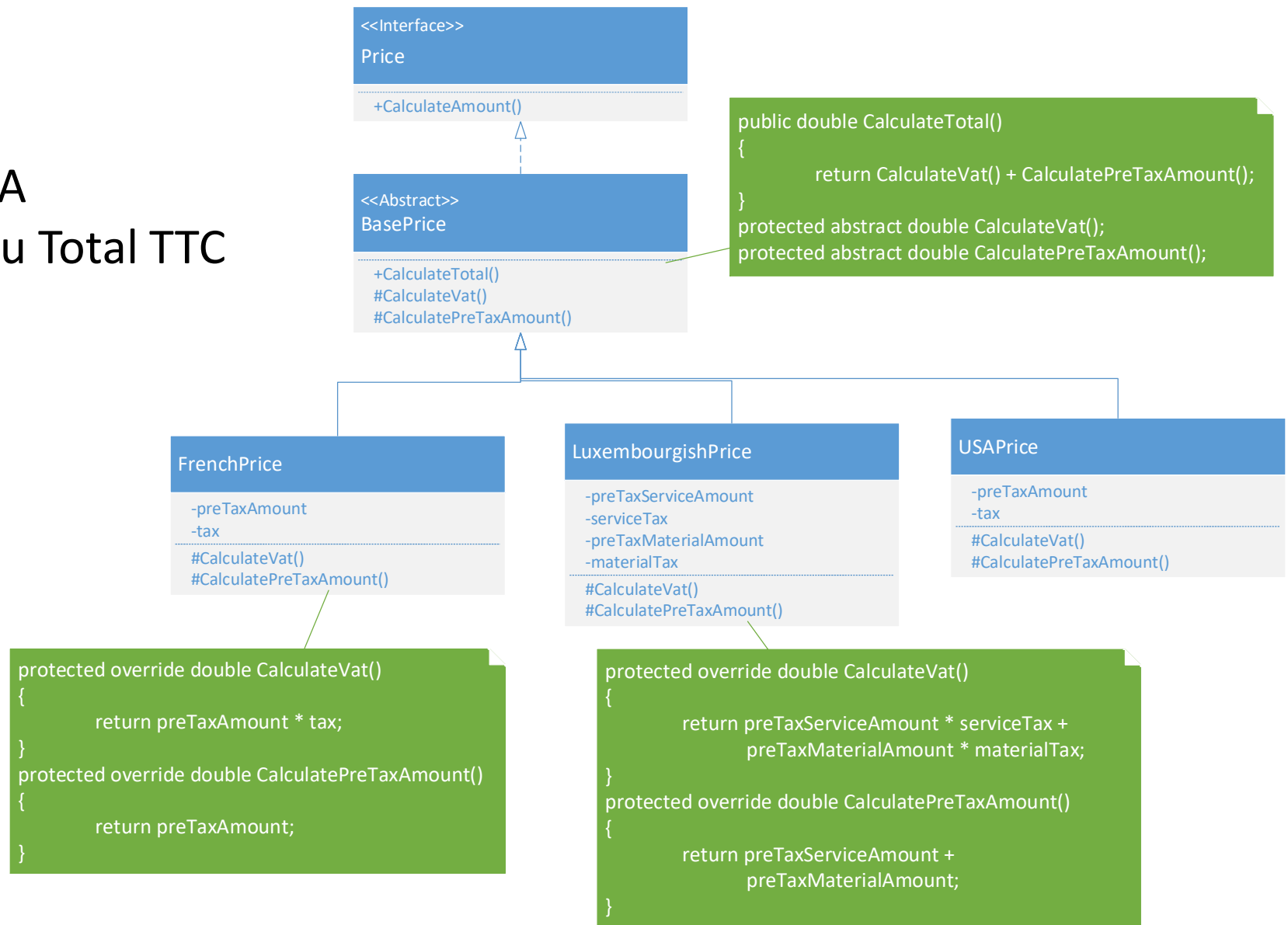
```
protected override double CalculateVat()
{
    return preTaxAmount * tax;
}
protected override double CalculatePreTaxAmount()
{
    return preTaxAmount;
}
```

```
LuxembourgishPrice
-preTaxServiceAmount
-serviceTax
-preTaxMaterialAmount
-materialTax
#CalculateVat()
#CalculatePreTaxAmount()
```

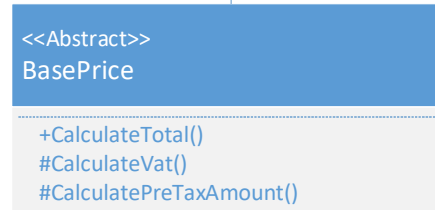
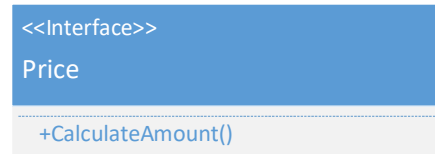
```
protected override double CalculateVat()
{
    return preTaxServiceAmount * serviceTax +
        preTaxMaterialAmount * materialTax;
}
protected override double CalculatePreTaxAmount()
{
    return preTaxServiceAmount +
        preTaxMaterialAmount;
}
```

Couplage Dépendance

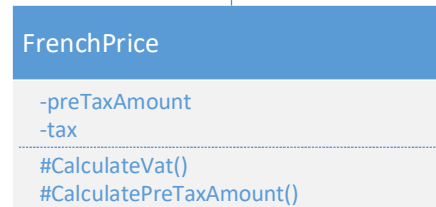
USA
Tips = 10% du Total TTC



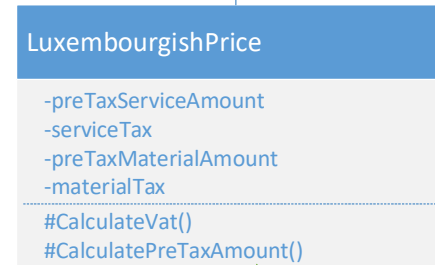
Défauts du choix



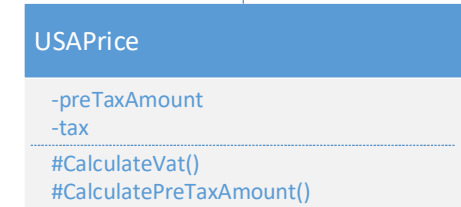
```
public double CalculateTotal()
{
    if(this.GetType() == typeof(USAPrice))
    {
        var total = CalculateVat() + CalculatePreTaxAmount();
        return total + total * 0.10;
    }
    return CalculateVat() + CalculatePreTaxAmount();
}
protected abstract double CalculateVat();
protected abstract double CalculatePreTaxAmount();
```



```
protected override double CalculateVat()
{
    return preTaxAmount * tax;
}
protected override double CalculatePreTaxAmount()
{
    return preTaxAmount;
}
```



```
protected override double CalculateVat()
{
    return preTaxServiceAmount * serviceTax +
        preTaxMaterialAmount * materialTax;
}
protected override double CalculatePreTaxAmount()
{
    return preTaxServiceAmount +
        preTaxMaterialAmount;
}
```



Défauts du choix



```
<<Interface>>
Price
+CalculateAmount()
```

```
<<Abstract>>
BasePrice
+CalculateTotal()
#CalculateVat()
#CalculatePreTaxAmount()
```

```
public virtual double CalculateTotal()
{
    return CalculateVat() + CalculatePreTaxAmount();
}
protected abstract double CalculateVat();
protected abstract double CalculatePreTaxAmount();
```

```
FrenchPrice
-preTaxAmount
-tax
#CalculateVat()
#CalculatePreTaxAmount()
```

```
LuxembourgishPrice
-preTaxServiceAmount
-serviceTax
-preTaxMaterialAmount
-materialTax
#CalculateVat()
#CalculatePreTaxAmount()
```

```
USAPrice
-preTaxAmount
-tax
#CalculateVat()
#CalculatePreTaxAmount()
```

```
protected override double CalculateVat()
{
    return preTaxAmount * tax;
}
protected override double CalculatePreTaxAmount()
{
    return preTaxAmount;
}
```

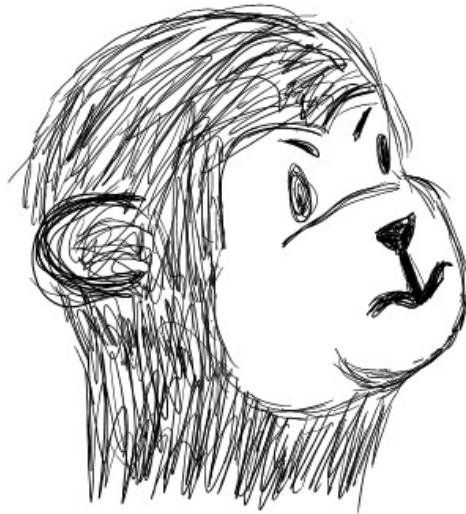
```
protected override double CalculateVat()
{
    return preTaxServiceAmount * serviceTax +
        preTaxMaterialAmount * materialTax;
}
protected override double CalculatePreTaxAmount()
{
    return preTaxServiceAmount +
        preTaxMaterialAmount;
}
```

```
public override double CalculateTotal()
{
    var total = CalculateVat() + CalculatePreTaxAmount();
    // var total = base.CalculateTotal();
    return total + total * 0.10;
}
protected override double CalculateVat()
{
    return preTaxAmount * tax;
}
protected override double CalculatePreTaxAmount()
{
    return preTaxAmount;
}
```

SOLID

Principe de substitution de Liskov

Classe abstraite



**Non
instanciable !**

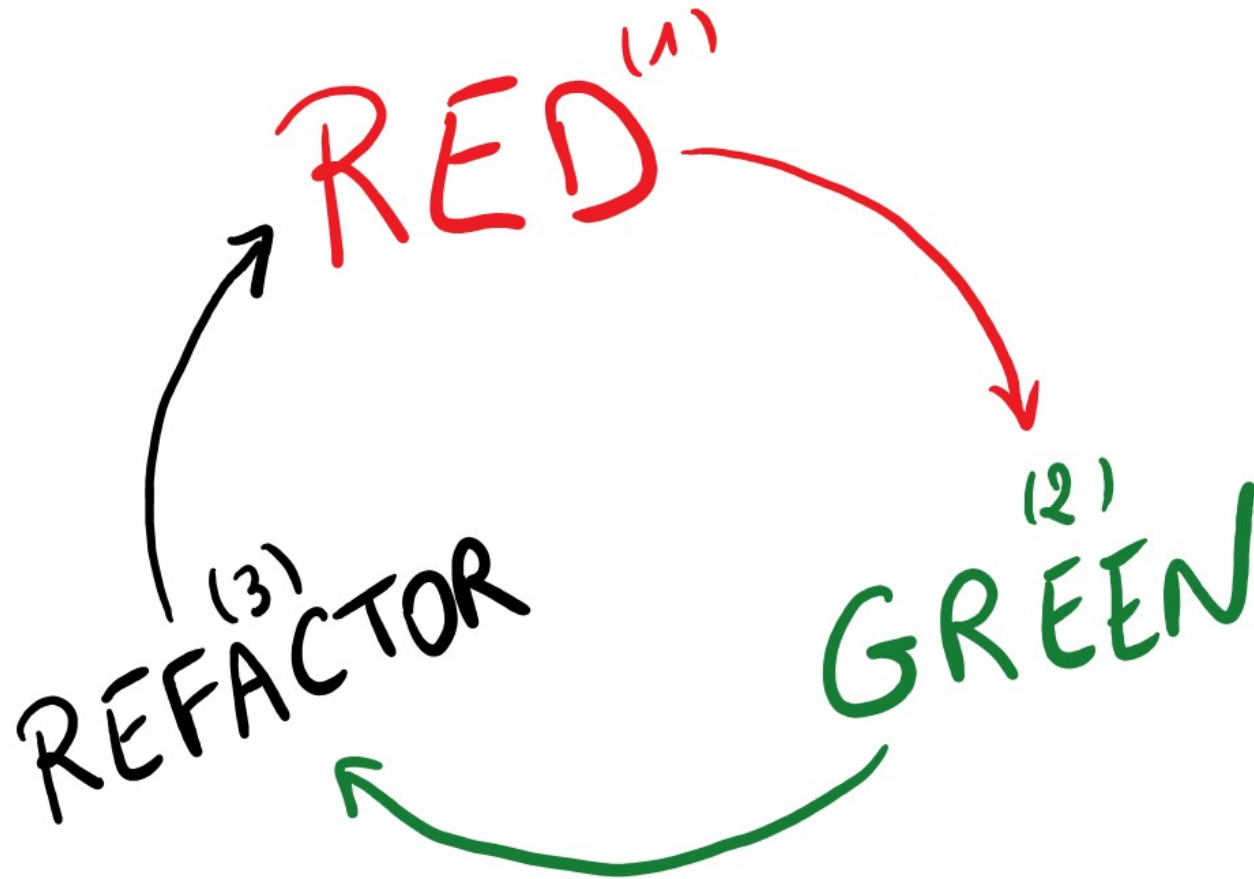
**Couplage
Dépendance**

**Le même test
se répète !**

Les classes abstraites,
c'est fini !

(et c'est la faute à TDD)

TDD



Prérequis



Décomposition

Exemples

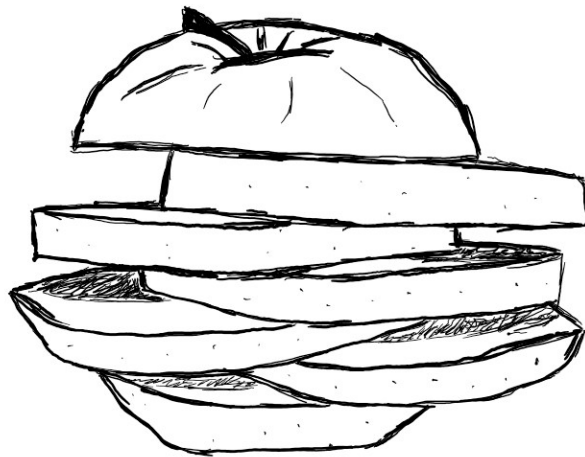
Priorisation

Baby steps

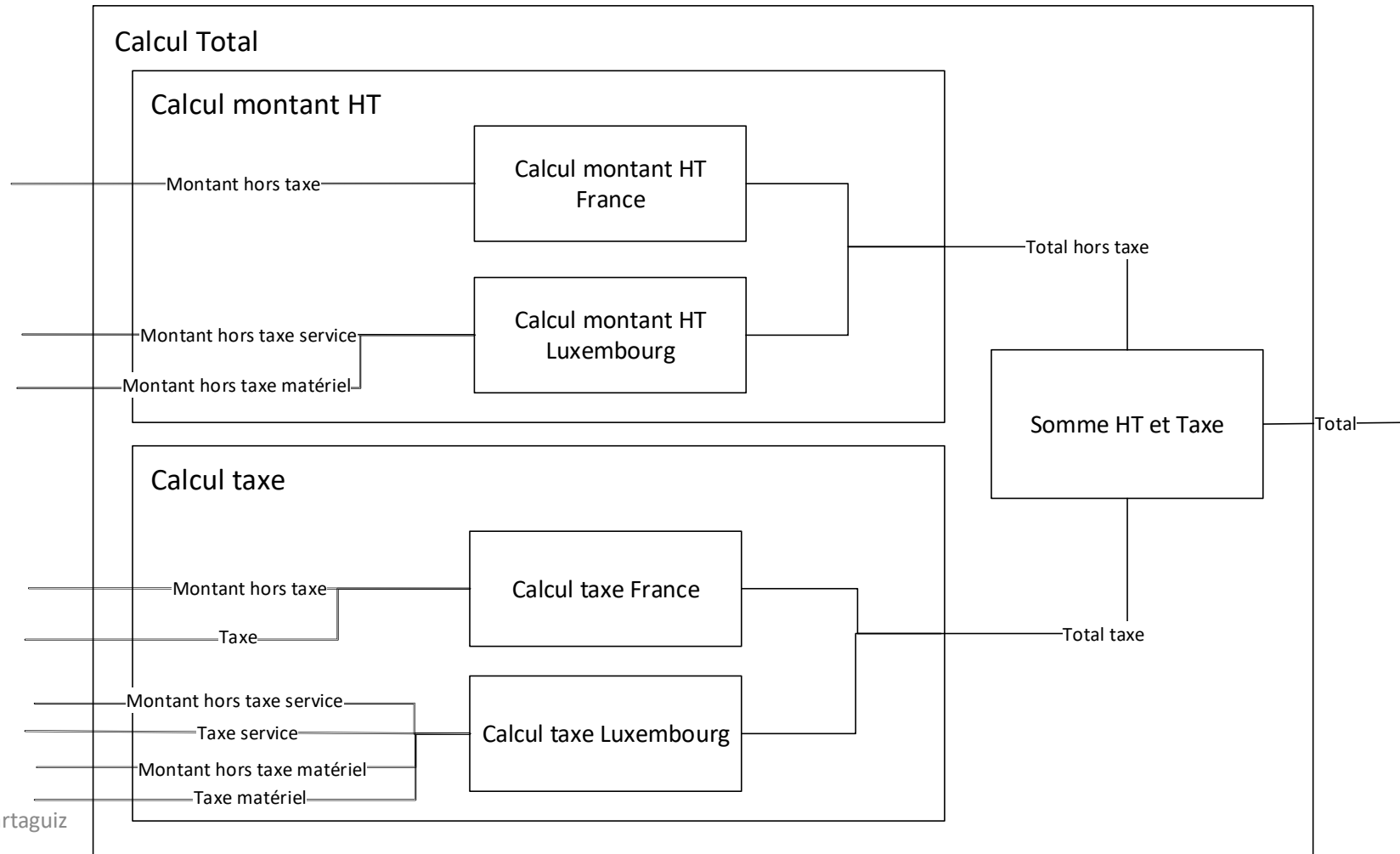
Besoin du garagiste

- En France
 - 150€ HT & 19.6% taxe
 - Total : 179,4€ TTC
- Au Luxembourg
 - 100€ HT de matériel & 15% taxe de matériel => total matériel : 115€
 - 50€ HT de service & 12% taxe de service => total service : 56€
 - Total : 171€

Décomposer



Identifier les composants



Identifier les composants

- Calcul du total
- Calcul du montant HT
 - France
 - Luxembourg
- Calcul de la taxe
 - France
 - Luxembourg

Inside out
vs
Outside in

Inside out

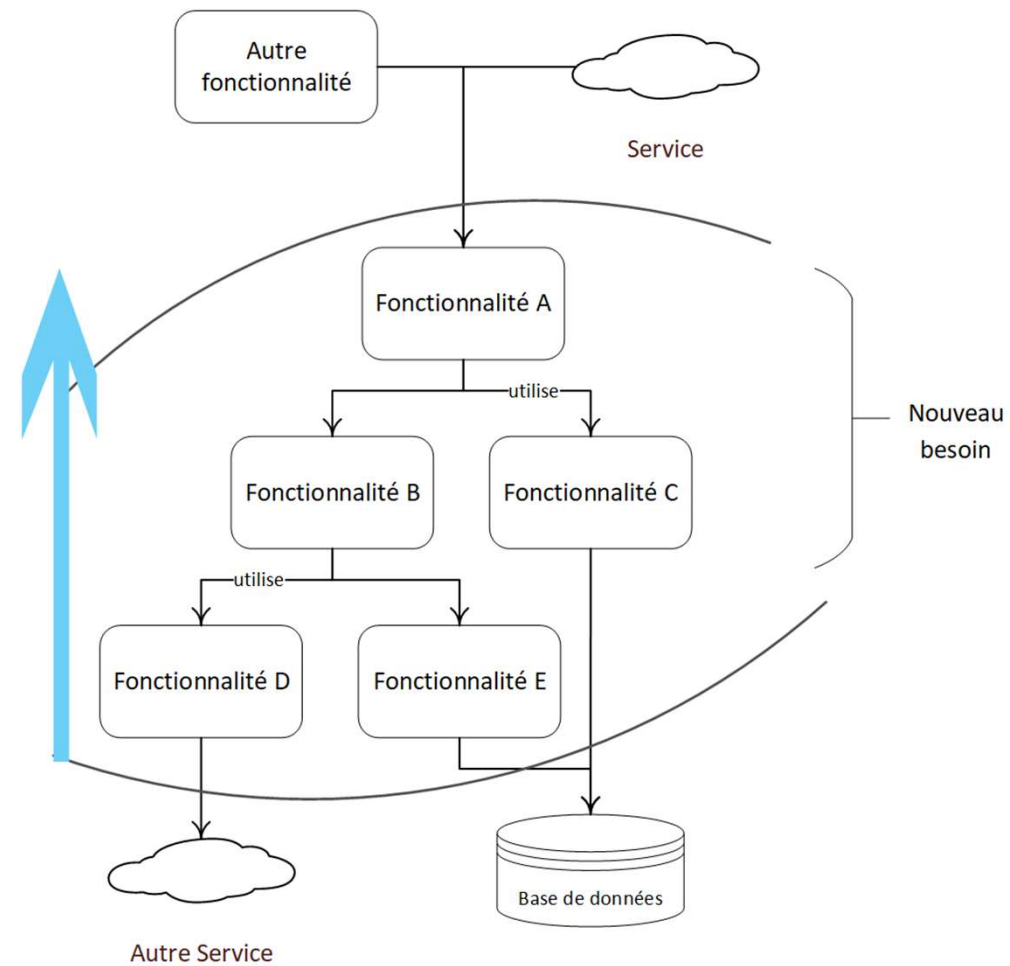
Classic approach



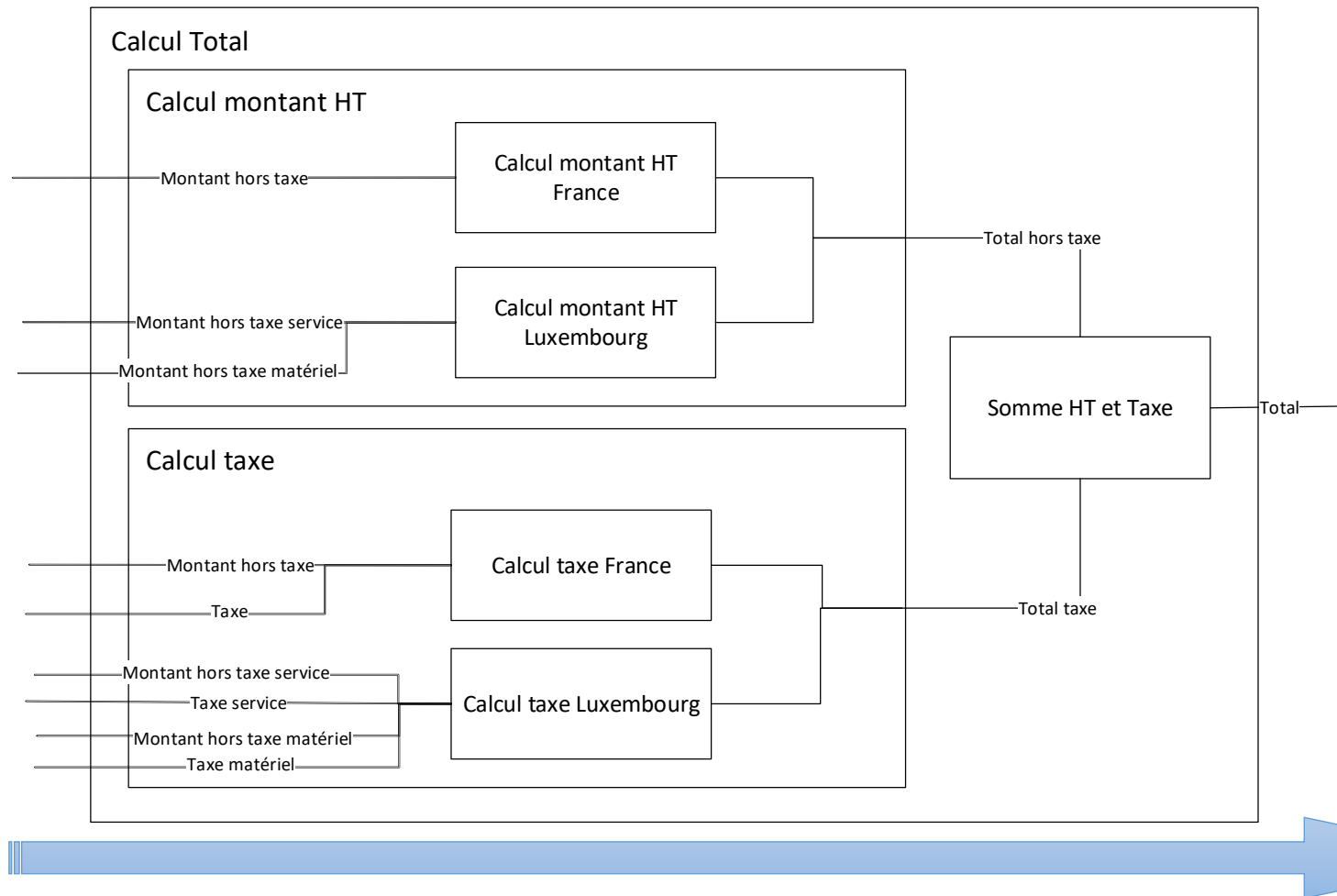
Chicago school

Bottom up

Inside out



Inside out



Outside in

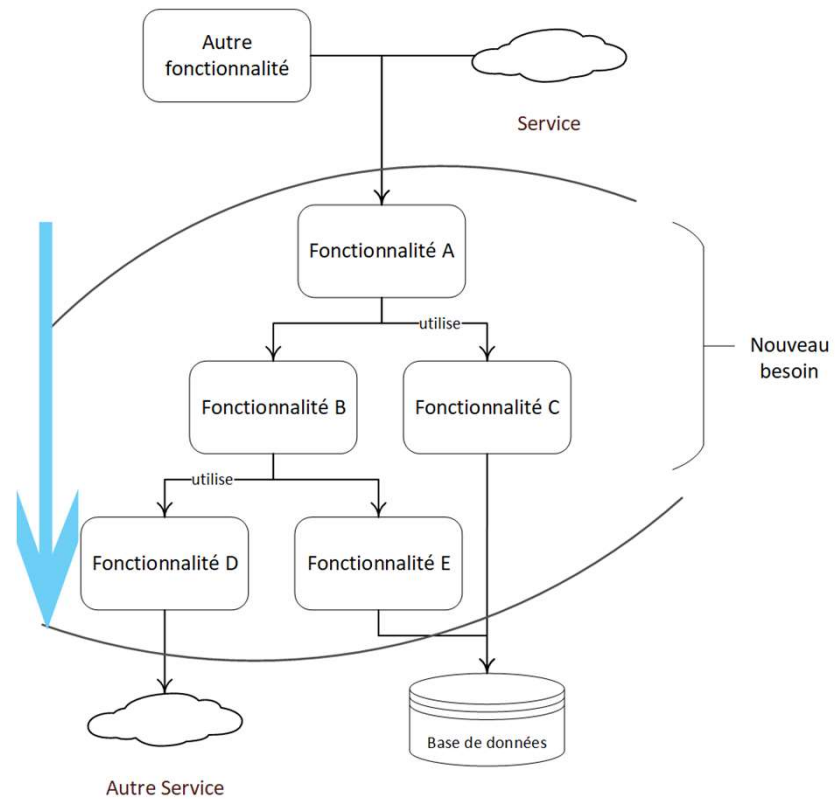
Mockist approach



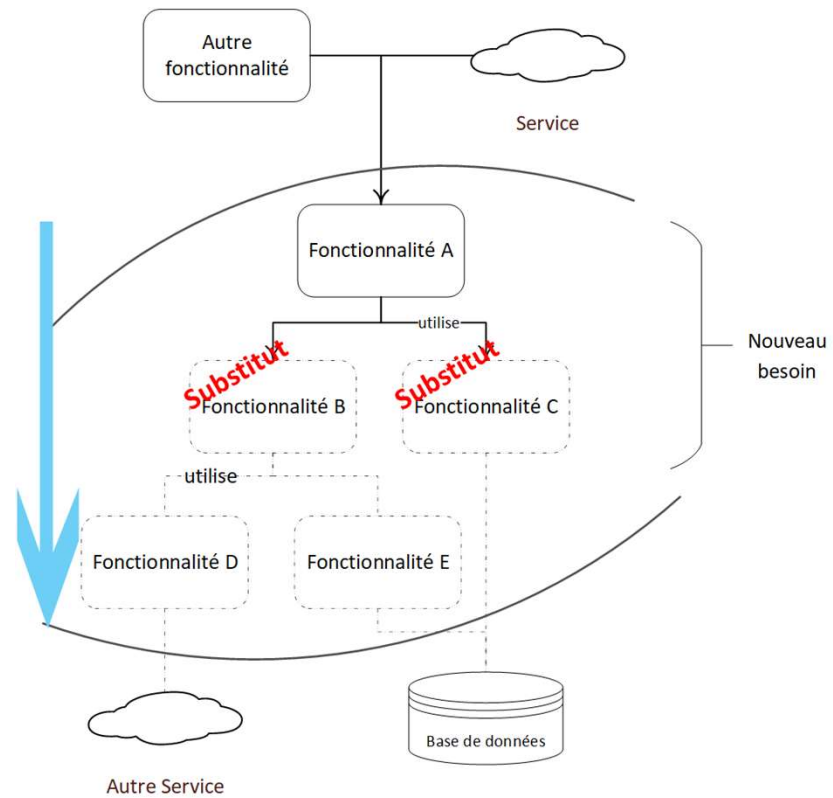
London school

Top down

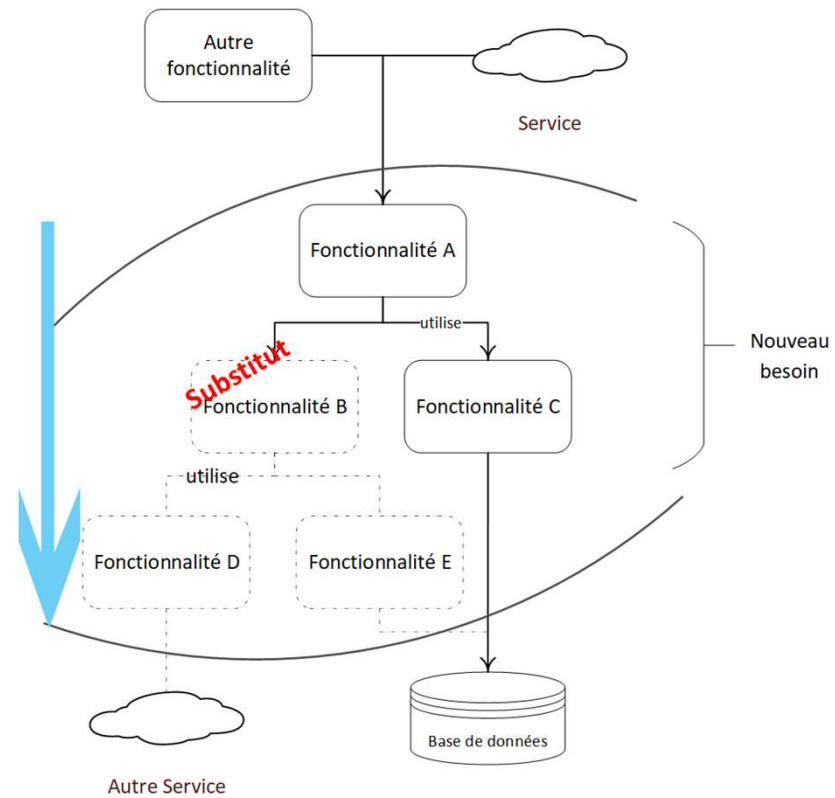
Outside in



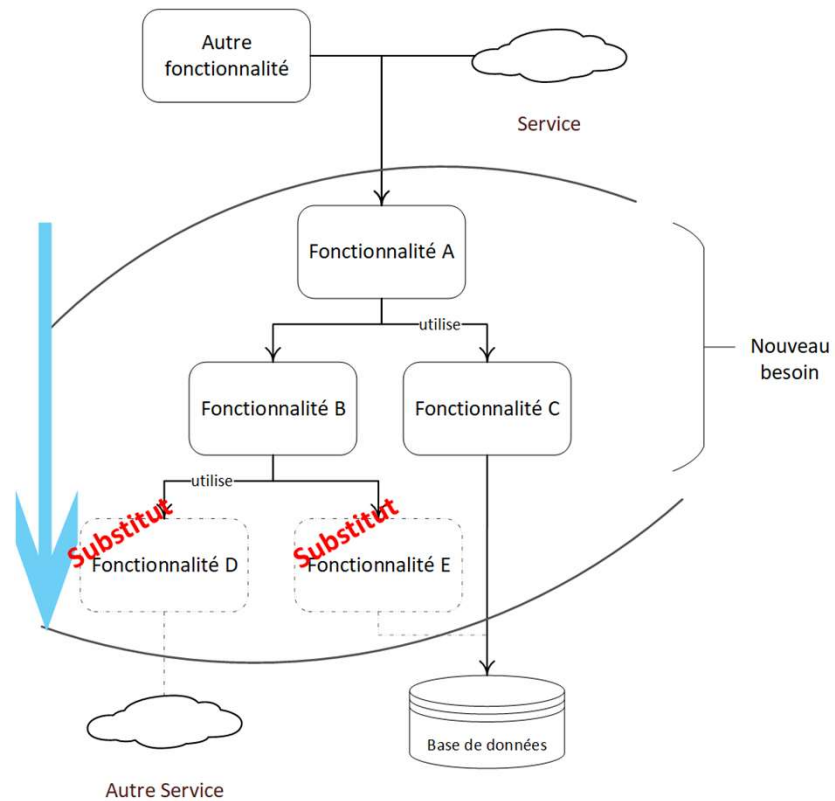
Outside in



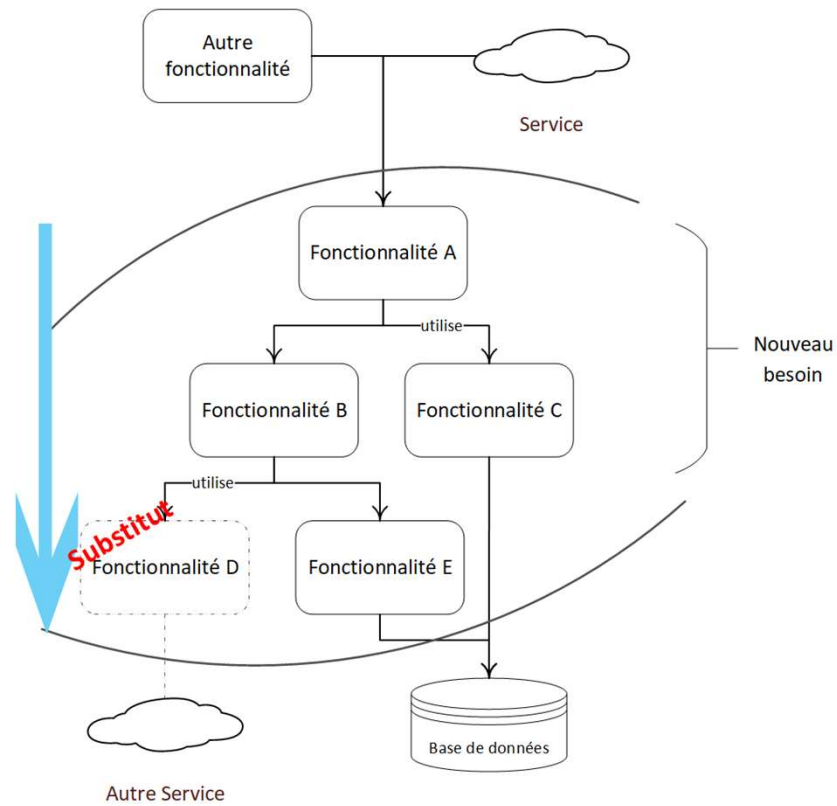
Outside in



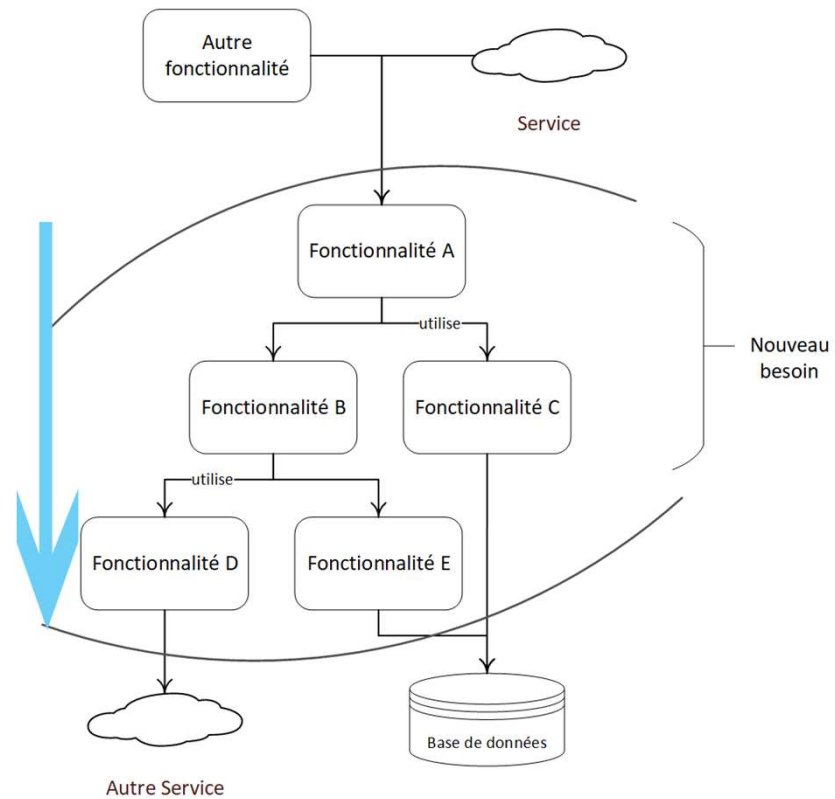
Outside in



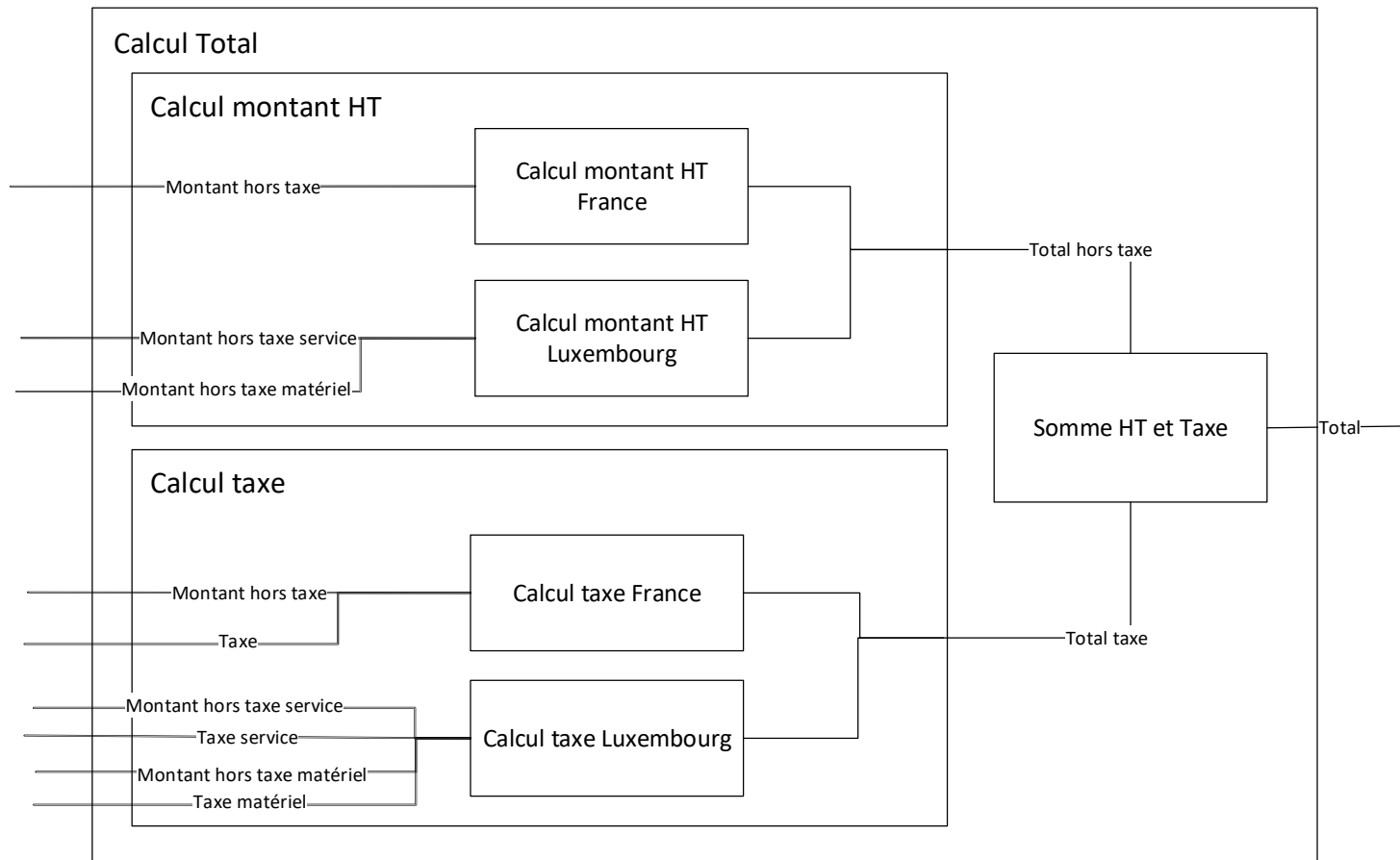
Outside in



Outside in

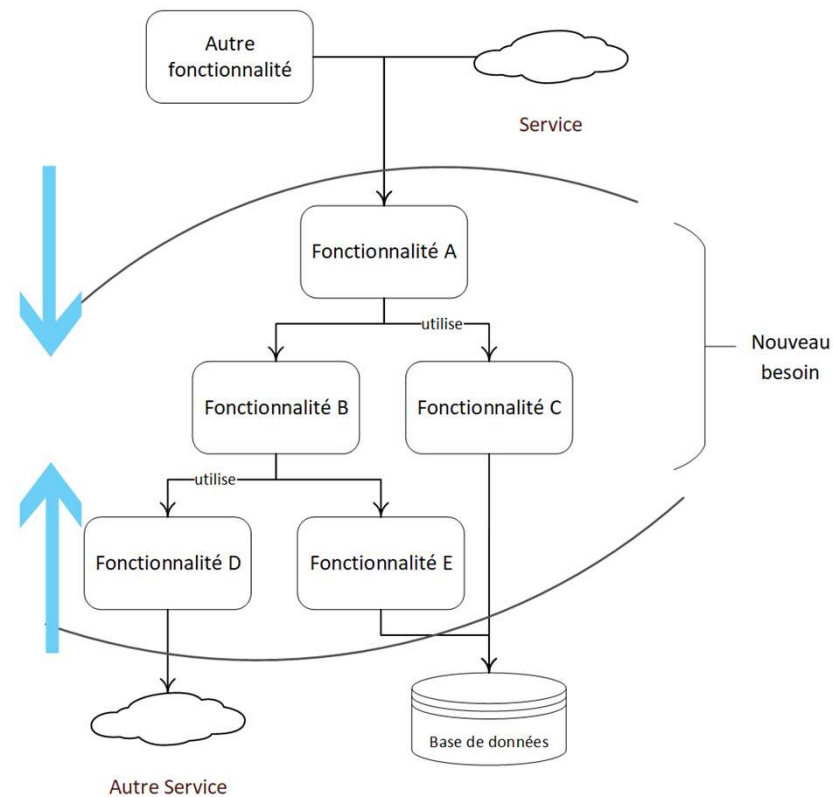


Outside in

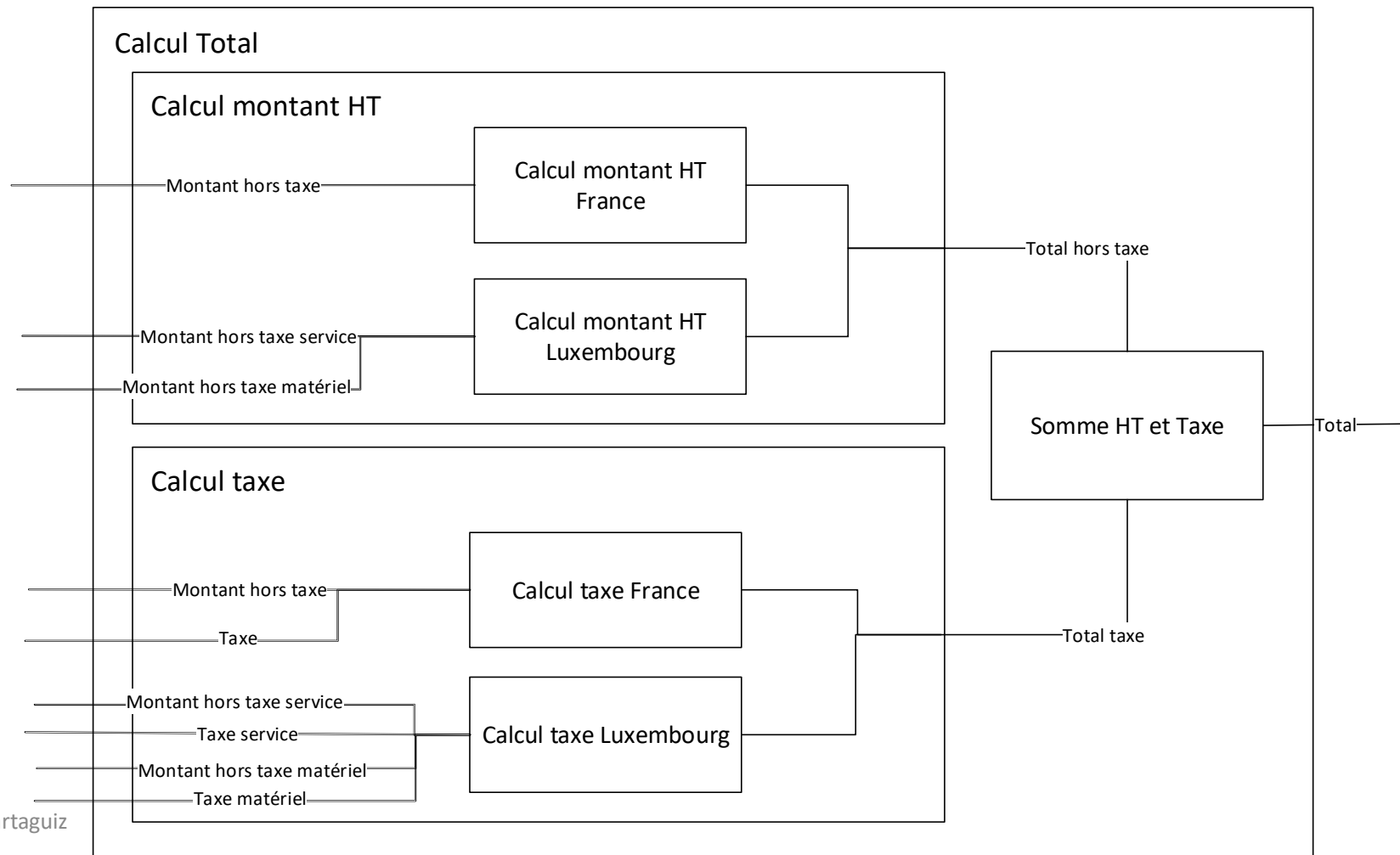


Inside out
vs
Outside in

Inside out vs Outside in



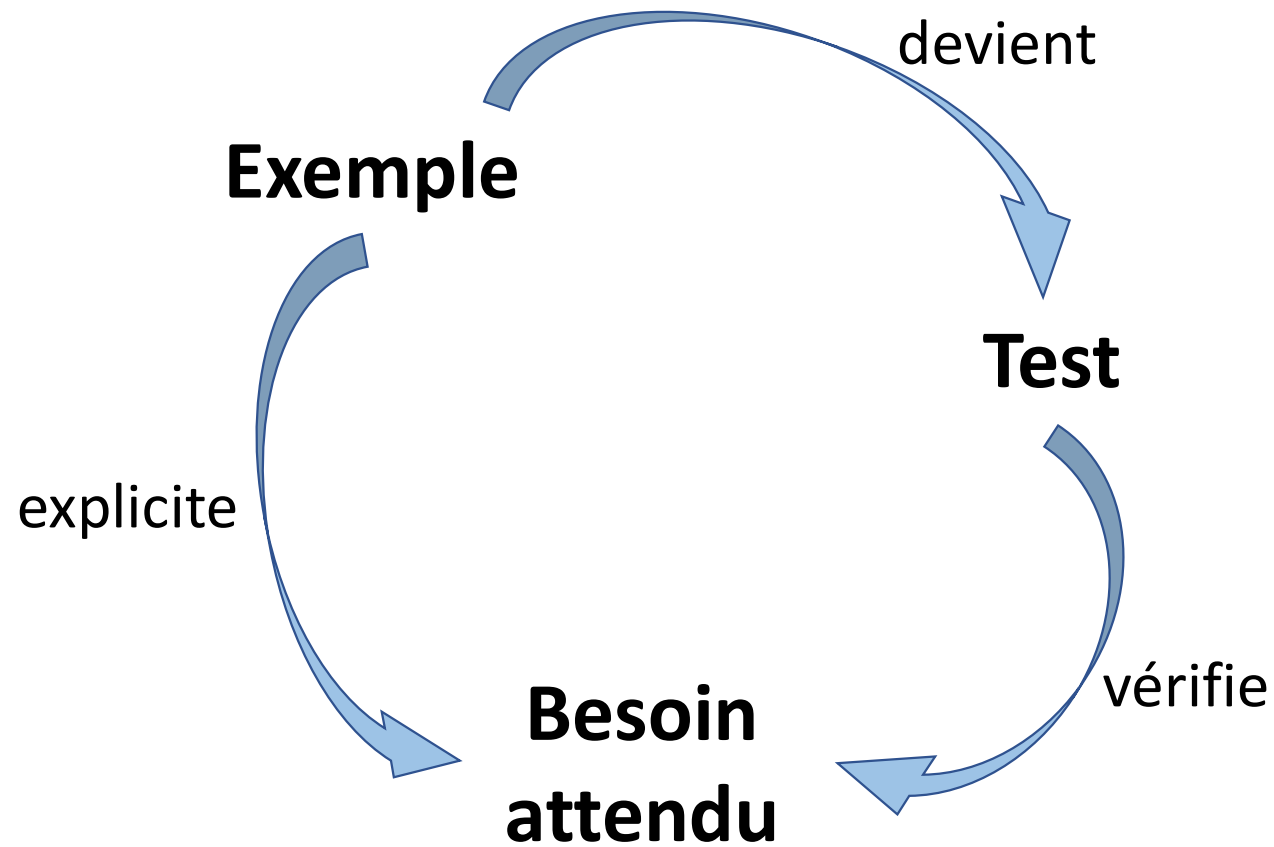
Inside out



Trouver des exemples



Bénéfices des exemples



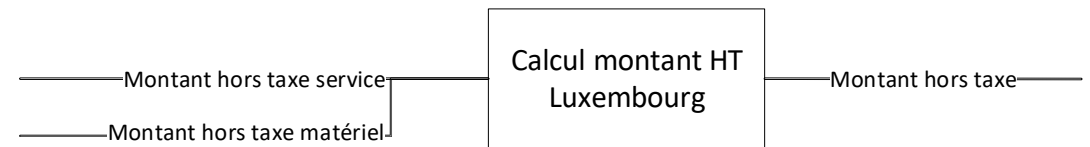


Les exemples à consommer
sans modération

Calcul montant HT

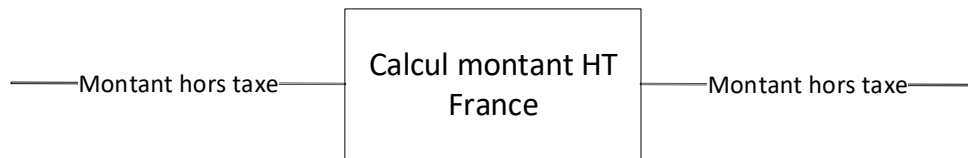


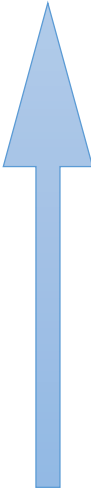
- 150 € => 150 €
- 10.5 € => 10.5 €
- 0 € => 0 €

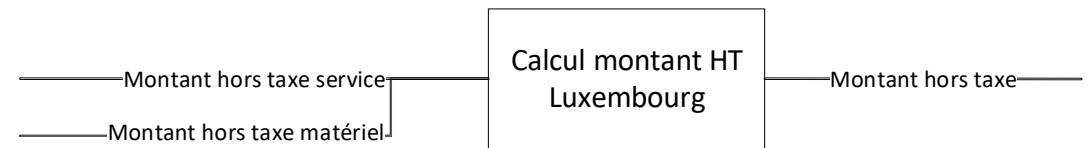


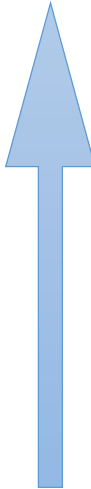
- 100 €, 50 € => 150 €
- 100 €, 0 € => 100 €
- 10.5 €, 0 € => 10.5 €
- 0 €, 0 € => 0 €

Calcul montant HT



- 
- 150 € => 150 €
 - 10.5 € => 10.5 €
 - 0 € => 0 €



- 
- 100 €, 50 € => 150 €
 - 100 €, 0 € => 100 €
 - 10.5 €, 0 € => 10.5 €
 - 0 €, 0 € => 0 €



Baby steps

L'art d'identifier des exemples pour avancer par petits pas

Calcul montant HT

FrenchPrice

+FrenchPrice(double preTaxAmount)

-preTaxAmount

+CalculatePreTaxAmount()

```
public double CalculatePreTaxAmount()
{
    return preTaxAmount;
}
```

LuxembourgishPrice

+LuxembourgishPrice(
double preTaxServiceAmount,
double preTaxMaterialAmount)

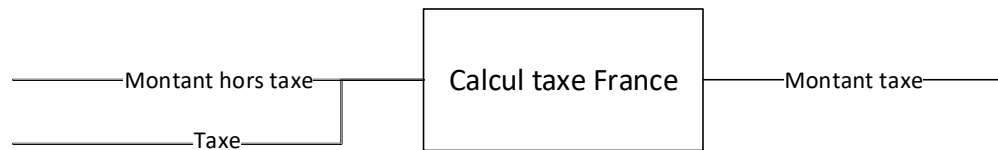
-preTaxServiceAmount

-preTaxMaterialAmount

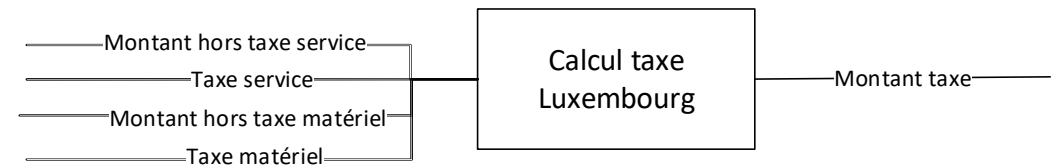
+CalculatePreTaxAmount()

```
public double CalculatePreTaxAmount()
{
    return preTaxServiceAmount +
           preTaxMaterialAmount;
}
```

Calcul taxe

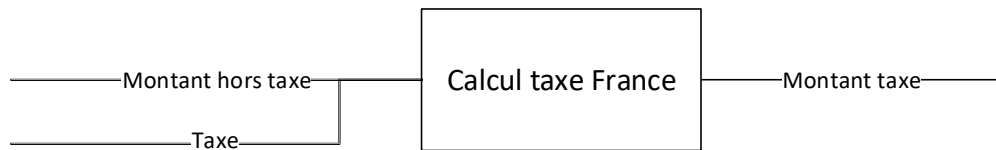


- 150 €, 19.6% => 29.4 €
- 0 €, 19.6% => 0 €

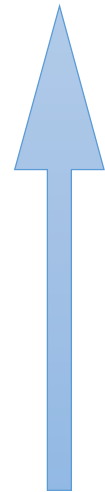
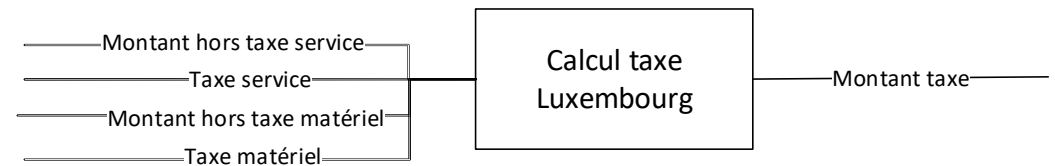


- 100 €, 15%, 50 €, 12% => 21 €
- 100 €, 15%, 0 €, 12% => 15 €
- 0 €, 15%, 0 €, 12% => 0 €

Calcul taxe

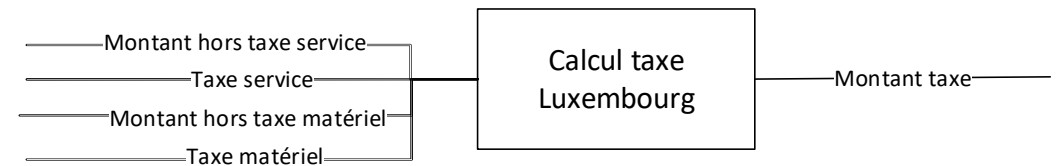
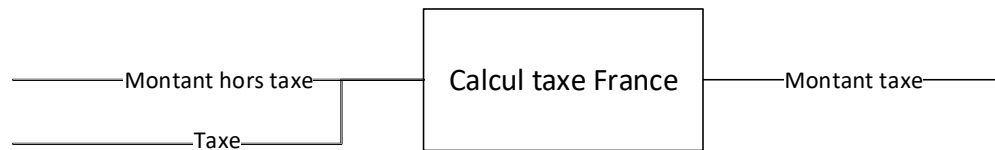


- 150 €, 19.6% => 29.4 €
- 0 €, 19.6% => 0 €



- 100 €, 15%, 50 €, 12% => 21 €
- 100 €, 15%, 0 €, 12% => 15 €
- 0 €, 15%, 0 €, 12% => 0 €

Calcul taxe



- 150 €, 19.6% => 29.4 €
- 0 €, 19.6% => 0 €

- 100 €, 15%, 50 €, 12% => 21 €
- 100 €, 15%, 0 €, 12% => 15 €
- 0 €, 15%, 0 €, 12% => 0 €

19.6% VS 0.196

15% VS 0.15

12% VS 0.12

Calcul taxe

- 0 €, 19.6% => 0 €

FrenchPrice

```
+FrenchPrice(  
double preTaxAmount,  
double tax)  
  
-preTaxAmount  
-tax  
-----  
+CalculatePreTaxAmount()  
+CalculateVat()
```

```
public double CalculatePreTaxAmount()  
{  
    return preTaxAmount;  
}  
public double CalculateVat()  
{  
    return preTaxAmount;  
}
```

@DorraBartaguis

- 0 €, 15%, 0 €, 12% => 0 €

LuxembourgishPrice

```
+LuxembourgishPrice(double preTaxServiceAmount,  
double preTaxMaterialAmount,  
double serviceTax, double materialTax)  
  
-preTaxServiceAmount  
-preTaxMaterialAmount  
-serviceTax  
-materialTax  
-----  
+CalculatePreTaxAmount()  
+CalculateVat()
```

```
public double CalculatePreTaxAmount()  
{  
    return preTaxServiceAmount +  
        preTaxMaterialAmount;  
}  
public double CalculateVat()  
{  
    return preTaxServiceAmount;  
}
```

Calcul taxe

- 150 €, 19.6% => 29.4 €

FrenchPrice

+FrenchPrice(
double preTaxAmount,
double tax)

-preTaxAmount
-tax

+CalculatePreTaxAmount()
+CalculateVat()

```
public double CalculatePreTaxAmount()
{
    return preTaxAmount;
}
public double CalculateVat()
{
    return preTaxAmount * tax;
}
```

@DorraBartaguis

- 100 €, 15%, 0 €, 12% => 15 €

LuxembourgishPrice

+LuxembourgishPrice(double preTaxServiceAmount,
double preTaxMaterialAmount,
double serviceTax, double materialTax)

-preTaxServiceAmount
-preTaxMaterialAmount
-serviceTax
-materialTax

+CalculatePreTaxAmount()
+CalculateVat()

```
public double CalculatePreTaxAmount()
{
    return preTaxServiceAmount +
        preTaxMaterialAmount;
}
public double CalculateVat()
{
    return preTaxServiceAmount * serviceTax;
}
```

Calcul taxe

- 150 €, 19.6% => 29.4 €

FrenchPrice

+FrenchPrice(
double preTaxAmount,
double tax)

-preTaxAmount
-tax

+CalculatePreTaxAmount()
+CalculateVat()

```
public double CalculatePreTaxAmount()  
{  
    return preTaxAmount;  
}  
public double CalculateVat()  
{  
    return preTaxAmount * tax;  
}
```

@DorraBartaguis

- 100 €, 15%, 50 €, 12% => 21 €

LuxembourgishPrice

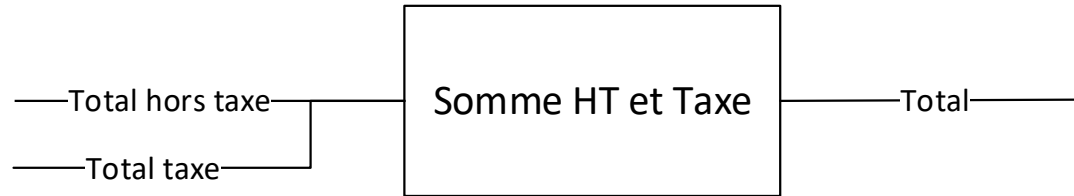
+LuxembourgishPrice(double preTaxServiceAmount,
double preTaxMaterialAmount,
double serviceTax, double materialTax)

-preTaxServiceAmount
-preTaxMaterialAmount
-serviceTax
-materialTax

+CalculatePreTaxAmount()
+CalculateVat()

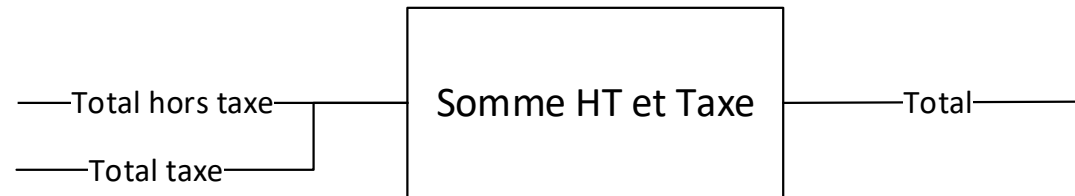
```
public double CalculatePreTaxAmount()  
{  
    return preTaxServiceAmount +  
        preTaxMaterialAmount;  
}  
public double CalculateVat()  
{  
    return preTaxServiceAmount * serviceTax  
        + preTaxMaterialAmount * materialTax;  
}
```

Somme HT et taxe



- 150 €, 29.4 € => 179.4 €
- 150 €, 0 € => 150 €

Somme HT et taxe



- 150 €, 29.4 € => 179.4 €
- 150 €, 0 € => 150 €

TotalCalculator

+CalculateTotal(double preTaxAmount, double taxAmount)

```
public double CalculateTotal(double preTaxAmount,
double taxAmount)
{
    return preTaxAmount + taxAmount;
}
```

Côté appellant

```
public void CalculateTotalPrice()
{
    var preTaxServiceAmount = 50;
    var serviceTax = 0.12;
    var preTaxMaterialAmount = 100;
    var materialTax = 0.15;
    var luxembourgishPrice = new LuxembourgishPrice(preTaxServiceAmount, serviceTax,
preTaxMaterialAmount, materialTax);

    var preTaxAmount = luxembourgishPrice.CalculatePreTaxAmount();
    var taxAmount = luxembourgishPrice.CalculateVat();

    var totalCalculator = new TotalCalculator();
    var total = totalCalculator.CalculateTotal(preTaxAmount, taxAmount);
    Check.That(total).IsEqualTo(171);
}
```

Simplifier l'appelant

```
public void CalculateTotalPrice_For_Luxembourg()
{
    var preTaxServiceAmount = 50;
    var serviceTax = 0.12;
    var preTaxMaterialAmount = 100;
    var materialTax = 0.15;
    var luxembourgishPrice = new LuxembourgishPrice(preTaxServiceAmount, serviceTax,
preTaxMaterialAmount, materialTax);

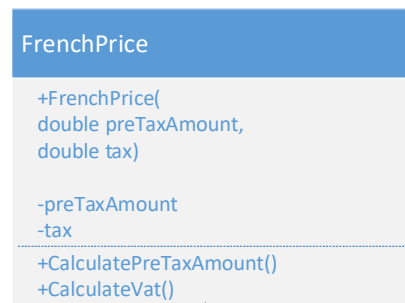
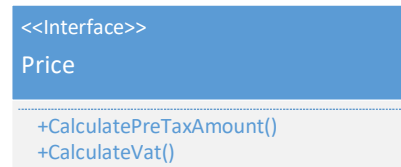
    var totalCalculator = new TotalCalculator(luxembourgishPrice);
    var totalCalculator.CalculateTotal();
    Check.That(total).IsEqualTo(171);
}
```

Simplifier l'appelant

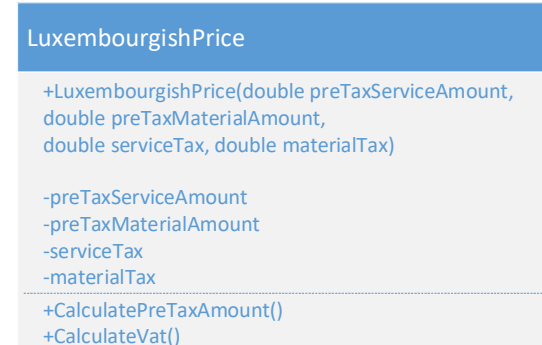
```
public void CalculateTotalPrice_For_France()
{
    var preTaxAmount = 150;
    var tax = 0.196;
    var frenchPrice = new FrenchPrice(preTaxAmount, tax);

    var totalCalculator = new TotalCalculator(frenchPrice);
    var totalCalculator.CalculateTotal();
    Check.That(total).IsEqualTo(179.4);
}
```


Interface

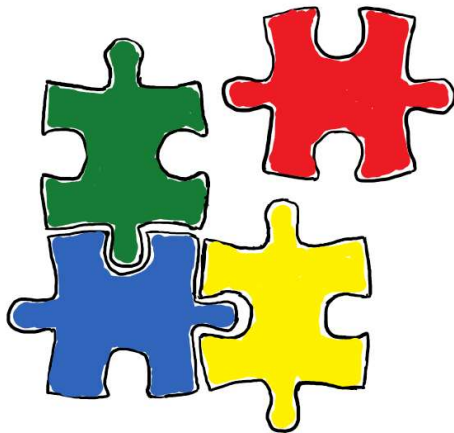


```
public double CalculatePreTaxAmount()
{
    return preTaxAmount;
}
public double CalculateVat()
{
    return preTaxAmount * tax;
}
```



```
public double CalculatePreTaxAmount()
{
    return preTaxServiceAmount +
        preTaxMaterialAmount;
}
public double CalculateVat()
{
    return preTaxServiceAmount * serviceTax
        + preTaxMaterialAmount * materialTax;
}
```

Composition



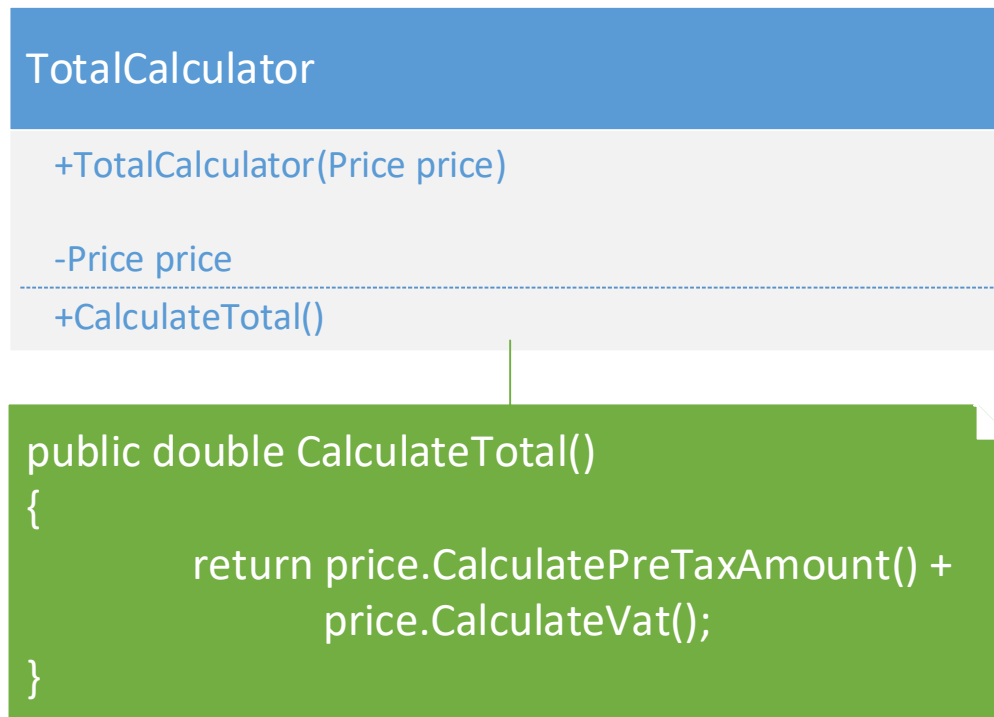
Pas de composition

TotalCalculator

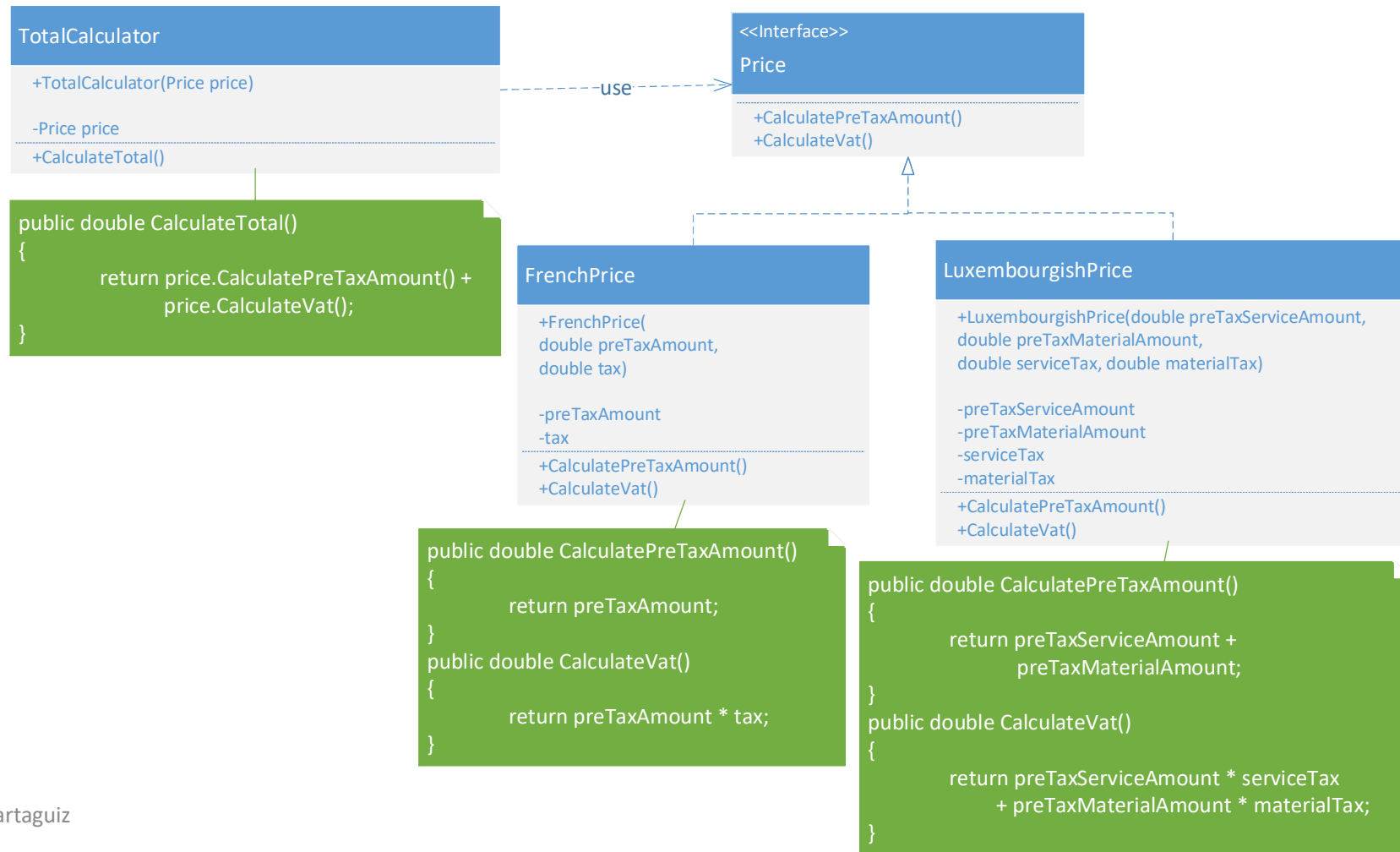
+CalculateTotal(double preTaxAmount, double taxAmount)

```
public double CalculateTotal(double preTaxAmount,  
double taxAmount)  
{  
    return preTaxAmount + taxAmount;  
}
```

Composition



Composition



Non instanciable

Le même test se répète

Couplage

Dépendance



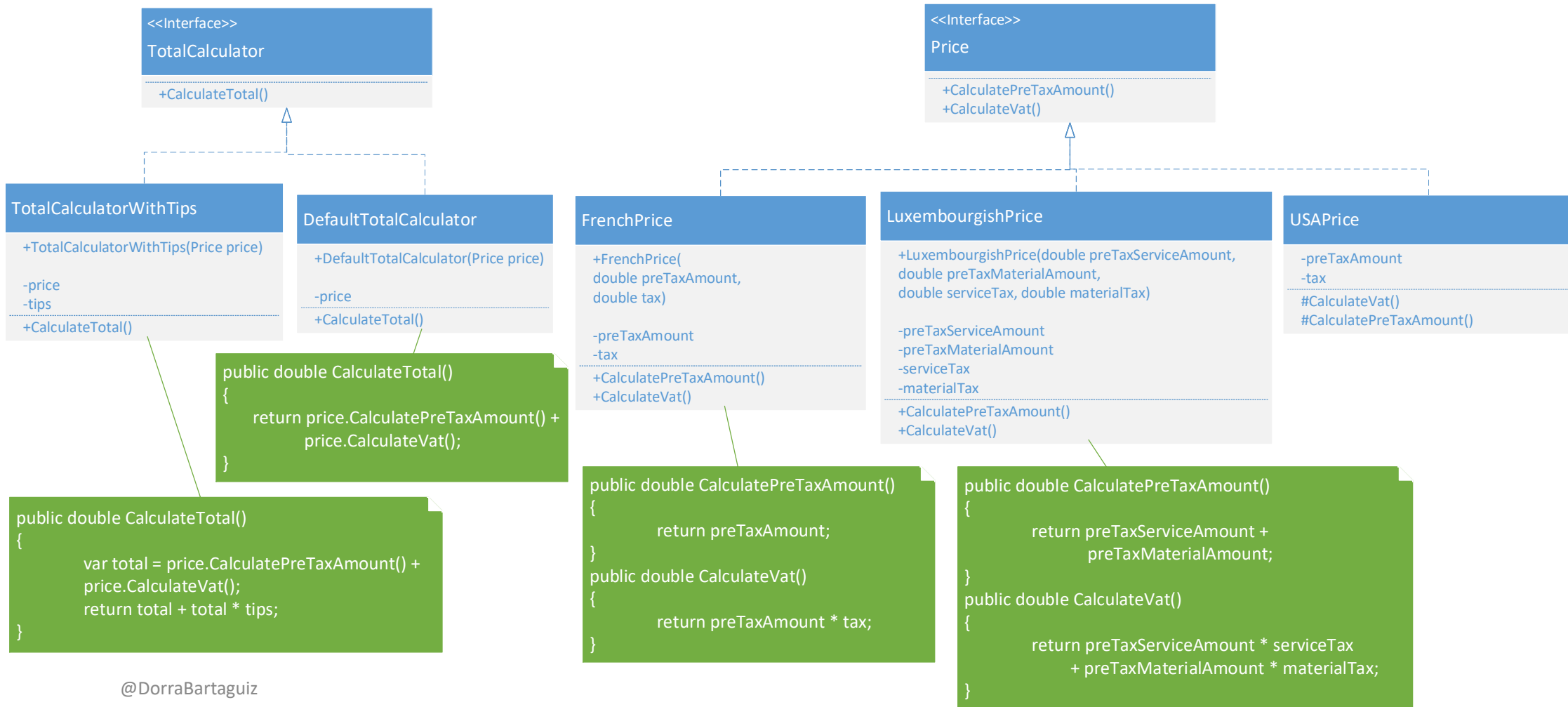
@DorraBartaguiz

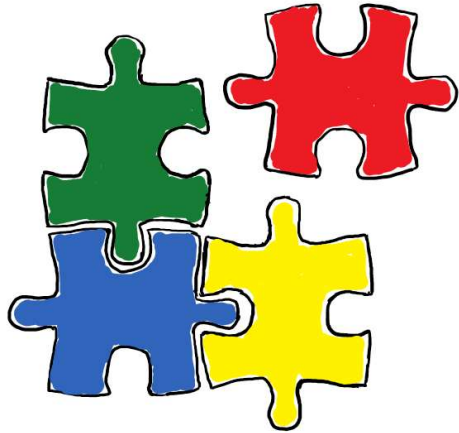




Tout est instanciable
Pas de test qui se répète
Dépendance inversée

Pourboires = Total TTC + 10%





Composition over Inheritance

« *Design Patterns: Elements of Reusable Object-Oriented Software* »
écrit par Erich Gamma, Richard Helm, Ralph Johnson, et John Vlissides,
avec une préface de Grady Booch

Composition



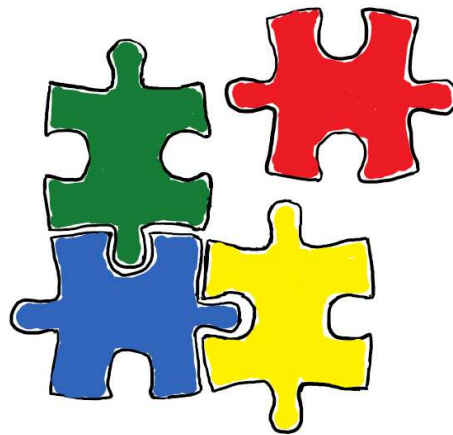
Make big things from small things in the same way.
That's the power of composition.

Scott Wlaschin



Les classes abstraites, c'est fini !

(et c'est la faute à **TDD**)



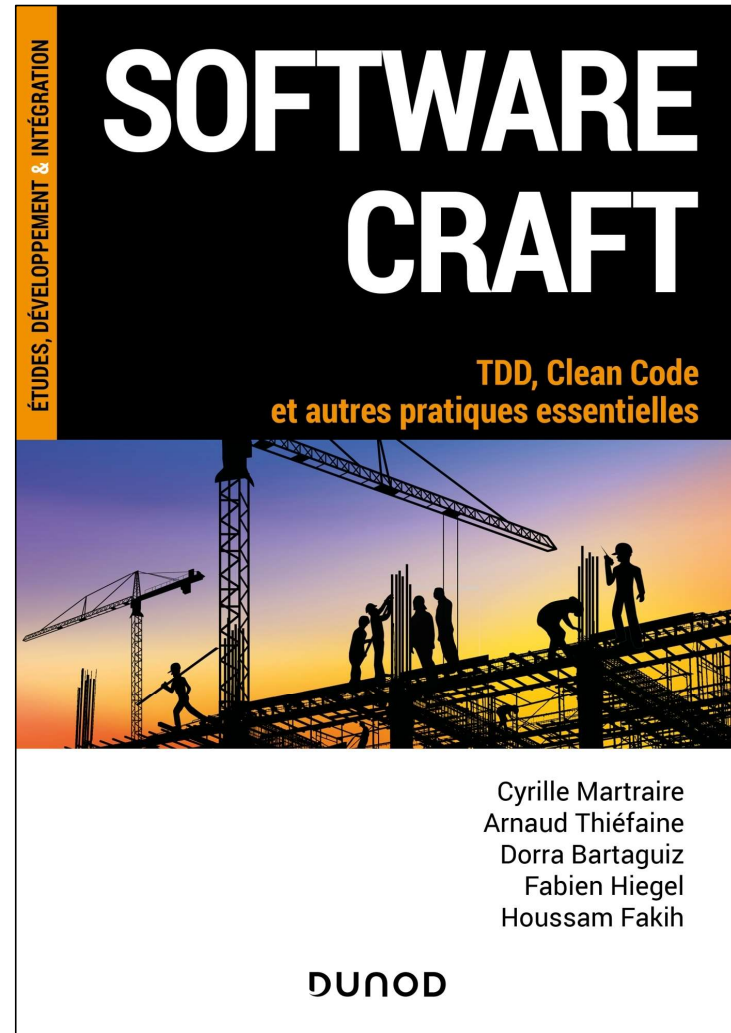
Baby steps

Composition

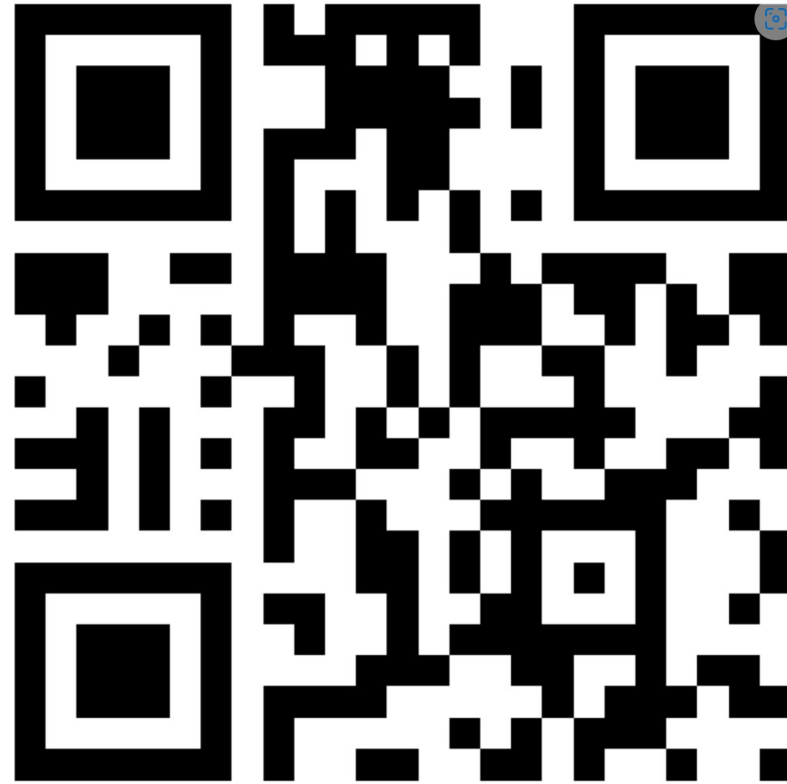
Buy the book!



@DorraBartaguiz



Merci !



<https://roti.express/r/alp-19>

@DorraBartaguiz