

Langage C++

- CIR2 -



---

## TP5 : Révision pour le DS1

Ayoub KARINE (ayoub.karine@isen-ouest.yncrea.fr)

---

### Compétences travaillées dans ce TP :

- Encapsulation
- Constructeurs
- Destructeurs

### Consignes :

- Tester dans le main, à chaque fois, chaque méthode développée
  - Utiliser le débogage
- 

### Exercice 1: Classe Rational

Créer une classe Rational qui a

- deux attributs entiers privés : numerator et denominator
- deux méthodes privés pour calculer le pgcd et le ppcm de deux entiers. Ces méthodes vont être utilisées par différentes méthodes publiques
- deux setters (vérifier bien si le dénominateur est différent de 0)
- deux getters

- une méthode pour afficher les fractions. Exemple d'affichage : 5/6
- un constructeur par défaut qui utilise les setters
- un constructeur qui utilise les setters
- Un constructeur de copie
- une méthode sans argument qui permet de simplifier la fraction et retourner une nouvelle fraction :

**Rational reduce()**

- une méthode qui vérifie si deux fractions sont égales :

**Bool equFract(Rational r1)**

- une méthode qui calcule la somme de deux fractions :

**Rational sumFract(Rational r1)**

- une méthode qui calcule le produit de deux fractions :

**Rational prodFract(Rational r1)**

## Exercice 2: Classe Trafficlight

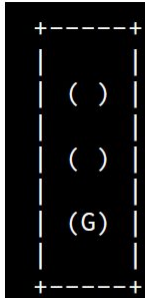
Soit l'entête ci-dessous de la classe Trafficlight:

```
enum class SignalColor { Red, Green, Yellow };
class Trafficlight {
private:
    SignalColor color;
Public:
    /* vérifier si initial_color a une des trois valeur de
    SignalColor. Si ce n'est le cas, la valeur par défaut est
    le rouge. Pour accéder à une couleur on utilise
    SignalColor::Red
    */
    Trafficlight(SignalColor initial_color);

    // change la couleur courante (voir l'exemple ci-dessous
```

```
void change();  
SignalColor get_color();  
// pour l'affichage, voir l'exemple ci-dessous  
void print();  
};
```

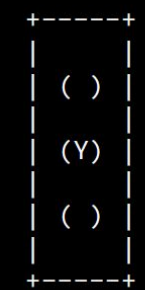
Exemple d'affichage :



```
+-----+  
| ( ) |  
| ( ) |  
| (G) |  
+-----+
```

en cliquant sur "Entrer" on aura l'affichage (sa traduction en c++ est :

`std::cin.get();`)



```
+-----+  
| ( ) |  
| (Y) |  
| ( ) |  
+-----+
```

en cliquant sur "Entrer" on aura l'affichage

(R)
( )
( )

en cliquant sur "Entrer" on aura l'affichage

( )
( )
(G)