

## TP5 : Révision

Nils Beaussé (nils.beausse@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 Rationnel

Créer une classe Rationnel qui a

- deux attributs entiers privés : numerateur et denominateur
- deux méthodes privés pour calculer le pgcd et le ppcm de deux entiers. Ces méthodes vont être utilisées par différentes méthodes publiques
- deux mutateurs (vérifier bien si le dénominateur est différent de 0)
- deux accesseurs
- 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 :

**Rationnel reduce()**

- une méthode qui vérifie si deux fractions sont égales :

**Bool equFract(Rationnel r1)**

- une méthode qui calcule la somme de deux fractions :

**Rationnel somFract(Rationnel r1)**

- une méthode qui calcule le produit de deux fractions :

**Rationnel prodFract(Rationnel 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 (En c++ on peut récupérer un caractère spécial avec `std::cin.get()`; et 10 est le numéro du caractère de saut de ligne.)

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

en cliquant sur "Entrer" on aura l'affichage

```
+-----+
|      |
|  (R)  |
|      |
|  ( )  |
|      |
|  ( )  |
|      |
+-----+
```

en cliquant de nouveau sur "Entrer" on aura l'affichage

