

## TD N°2

### EXERCICE 5: Littéraux

On veut définir une classe Angle tel qu'elle est utilisée pour stocker un angle en radian (unité par défaut).

1. écrire la classe Angle.
2. peut-on construire directement un Angle depuis un double (i.e. écrire Angle a = 0.4) ?
3. ajouter une surcharge de l'opérateur << sur le flux de sortie standard afin que son unité soit affichée.
4. modifier la classe permette une conversion direct en double (i.e. sans utiliser de getter),
5. ajouter une méthode permettant d'obtenir l'angle en degré.
6. on veut utiliser les littéraux \_deg et \_rad afin d'initialiser un angle soit à partir de constante double en degré ou en radia. Écrire les fonctions nécessaires.
7. donner un code qui illustre les fonctionnalités décrites ci-dessus.

### Solution:

1. classe Angle :

```
// must be before anything else
const double Pi_d = M_PI;
const float Pi = static_cast<float>(M_PI);

class Angle {
private:
    double value;    // stocké en radian
public:
    Angle(double v) : value(v) {}
    // conversion d'un Angle en double: pas besoin de getter! (question 4)
    operator double() const { return value; }
    // l'angle en degré (question 5)
    double deg() { return value * 180.0 / Pi_d; }
    // surcharge pour l'affichage (question 3)
    friend ostream& operator<<(ostream& os, const Angle& a);
};
```

2. le constructeur le permet directement.
3. surcharge de l'opérateur <<

```
// question 3
ostream& operator<<(ostream& os, const Angle& a) {
    os << a.value << "rad";
    return os;
}
```

4. littéraux

```
// question 6
Angle operator"" _rad(long double x) {
    return Angle(static_cast<double>(x)); }
Angle operator"" _deg(long double x) {
    return Angle(static_cast<double>(x / 180.0 * Pi_d)); }
```

## 5. code d'illustration

```
// question 7
Angle a = Pi_d / 2.0;
cout << "a=" << a << endl;
Angle b = 180.0_deg; // note 180_deg ne fonctionne pas car 180 est un entier
cout << "b=" << b << endl;
Angle c = 180.0_rad;
cout << "b=" << b << endl;
double V1 = a;
double V2 = 3.0 * a; // si conversion en float aussi définie, ceci devient ambigu
double V3 = a.deg() / 6;
cout << "V1=" << V1 << "_V2=" << V2 << "_V3=" << V3 << endl;
```