

```
/*
-----
Laboratoire : 04
Fichier      : produit.cpp
Auteur(s)    : Thibaud Franchetti, Sacha Perdrizet
Date         : 08.04.2019

But          : Implémente la classe Produit déclarée dans produit.h

Remarque(s) : - Il aurait été préférable d'initialiser la donnée membre privée 'no'
                après le possible lancement d'une exception, malheureusement
                ceci est impossible.

Compilateur  : GCC-g++ 7.3.0
                GCC-g++ 8.2.0
-----
*/

#include "produit.h"
#include "exceptions.h"
#include <iomanip>
#include <sstream>

using namespace std;

const double Produit::PRIX_MINIMAL = 0.05;

// Génère le message d'erreur à afficher à l'aide du paramètre nomFonction
string Produit::prixNonValide(const char* nomFonction){
    const double PRIX_EN_CENTIMES = PRIX_MINIMAL * 100.0;
    stringstream os;

    os << "Erreur dans Produit::" << nomFonction << " : " << endl << "le prix doit etre >= " <<
    PRIX_EN_CENTIMES << " cts !";

    return os.str();
}

Produit::Produit(unsigned no, const string& libelle, double prix) : no(no) {
    if (prix < PRIX_MINIMAL) {
        throw PrixNonValide(prixNonValide(__func__));
    }
    (unsigned&) this->no = no;
    this->libelle = libelle;
    this->prix = prix;
}

Produit& Produit::operator=(const Produit& produit) {
    (unsigned&) this->no = produit.no;
    this->libelle = produit.libelle;
    this->prix = produit.prix;
    return *this;
}

void Produit::setPrix(double prix) {
    if (prix < PRIX_MINIMAL) {
        throw PrixNonValide(prixNonValide(__func__));
    }
    this->prix = prix;
}

double Produit::getPrix() const {
    return prix;
}

ostream& operator<<(ostream& os, const Produit& produit) {
    return os << '(' << produit.no << ", \"" << produit.libelle << "\", "
    << fixed << setprecision(2) << produit.prix << defaultfloat
    << setprecision(6) << ')';
}

bool operator==(const Produit& lhs, const Produit& rhs) {
    return lhs.no == rhs.no;
}
```