

Méthodes constantes

- Imprimante

Le code suivant ne compilera pas. Dites pourquoi et remédiez à la situation. Vous devez privilégier une solution sécuritaire.

```
class CartoucheEncre
{
public :
    CartoucheEncre(float niveau)
        { niveau_ = niveau; }
    float getNiveau()
        {return niveau_;}
    void afficher()
        { cout << "    " << getNiveau() << " % encre" << endl;}
private :
    float niveau_; // %
};

class BacFeuilles
{
public :
    BacFeuilles(int nbFeuilles)
        { nbFeuilles_ = nbFeuilles }
    int getNbFeuille()
        {return nbFeuilles_;}
    void afficher()
        { cout << "    " << getNbFeuille() << " feuilles" << endl;}
private :
    int nbFeuilles_;
};

class Imprimante
{
public :
    Imprimante(float niveauEncre, int nbFeuilles)
        : encreNoir_(niveauEncre), bacFeuilles_(nbFeuilles) {}

    void afficher() const
    {
        cout << "Etat de l'imprimante :" << endl;
        encreNoir_.afficher();
        bacFeuilles_.afficher();
    }
private :
    CartoucheEncre encreNoir_;
    BacFeuilles bacFeuilles_;
};
```

Solution à la page suivante

Solution

La méthode void Imprimante::afficher() const demandait à ce que toutes les fonctions appelées ne peuvent modifier la classe Imprimante ainsi que ses attributs. Il y a quatre **const** à ajouter.

```
class CartoucheEncre
{
public :
    CartoucheEncre(float niveau)
        { niveau_ = niveau; }
    float getNiveau() const
        {return niveau_;}
    void afficher() const
        { cout << "    " << getNiveau() << " % encre" << endl;}
private :
    float niveau_; // %
};

class BacFeuilles
{
public :
    BacFeuilles(int nbFeuilles)
        { nbFeuilles_ = nbFeuilles }
    int getNbFeuille() const
        {return nbFeuilles_;}
    void afficher() const
        { cout << "    " << getNbFeuille() << " feuilles" << endl;}
private :
    int nbFeuilles_;
};

class Imprimante
{
public :
    Imprimante(float niveauEncre, int nbFeuilles)
        : encreNoir_(niveauEncre), bacFeuilles_(nbFeuilles) {}

    void afficher() const
    {
        cout << "Etat de l'imprimante :" << endl;
        encreNoir_.afficher();
        bacFeuilles_.afficher();
    }
private :
    CartoucheEncre encreNoir_;
    BacFeuilles bacFeuilles_;
};
```