

Programmation Orientée Objets L3 Miage – Info M1 IMDS

Exercices sur le cours 4

Boîte

On a déjà la classe boîte suivante représentant une boîte :

```
class boîte {
public:
    boîte(int largeur, int profondeur, int hauteur);
    boîte(int taille); //boîte cubique
    int largeur() const;
    int profondeur() const;
    int hauteur() const;
private:
    int d_largeur, d_profondeur, d_hauteur;
};

boîte::boîte(int largeur, int profondeur, int hauteur):
    d_largeur{largeur}, d_profondeur{profondeur},
    d_hauteur{hauteur}
{}

boîte::boîte(int taille): d_largeur{taille},
    d_profondeur{taille}, d_hauteur{taille}
{}

int boîte::largeur() const { return d_largeur; }

int boîte::profondeur() const { return d_profondeur; }

int boîte::hauteur() const { return d_hauteur; }
```

On veut créer une boîte qui en plus indique le matériau avec lequel elle est fabriquée.

Écrivez la classe boîteEnMatériau :

- c’est une boîte qui en plus stocke son matériau de fabrication `d_materiau` (une chaîne)
- une boîte en matériau est construite à partir de ses dimensions et de son matériau et à partir d’une taille et de son matériau (boîte cubique) ;

- on veut connaître sa matériau de fabrication : méthode `materiau()`

On a la procédure `afficheBoîte` suivante qui affiche une boîte sous la forme :

boîte de largeur *largeur* de profondeur *profondeur* et de hauteur *hauteur*

```
void afficheBoîte(const boîte& b)
{
    cout<<"boîte de largeur "<<b.largeur();
    cout<<" de profondeur "<<b.profondeur();
    cout<<" et de hauteur "<<b.hauteur()<<endl;
}
```

Écrivez la procédure `afficheBoîteMatériau` qui affiche une boîte en matériau sous la forme suivante :

boîte de largeur *largeur* de profondeur *profondeur* et de hauteur *hauteur*
faite en *matériau*

Écrivez la procédure de test `test` qui crée la boîte en matériau de largeur 10, de profondeur 20, de hauteur 5 et de matériau carton et l’affiche. Compilez et testez.

Solution boite

Déclaration boiteEnMateriau :

```
class boiteEnMateriau : public boite {
public:
    boiteEnMateriau(int largeur, int profondeur, int hauteur,
const std::string& materiau);
    boiteEnMateriau(int taille, const std::string& materiau);
    std::string materiau() const;
private:
    std::string d_materiau;
};
```

Définition classe :

```
boiteEnMateriau::boiteEnMateriau(int largeur, int profondeur,
int hauteur, const std::string& materiau) :
    boite{largeur, profondeur, hauteur},
    d_materiau{materiau}
{}

boiteEnMateriau::boiteEnMateriau(int taille,
    const std::string& materiau) :
    boite{taille}, d_materiau{materiau}
{}

std::string boiteEnMateriau::materiau() const
{
    return d_materiau;
}
```

Affichage et programme de test

```
void afficheBoiteMateriau(const boiteEnMateriau& b)
{
    afficheBoite(b);
    cout<<"faite en "<<b.materiau()<<endl;
}

void test()
{
    boiteEnMateriau b{10,20,5, "carton"};
    afficheBoiteMateriau(b);
}
```