

Le promoteur [cm02] - Exercice

Karine Zampieri, Stéphane Rivière

Unisciel  algoprogram  Version 20 mai 2018

Table des matières

1	Le promoteur / pgpromoteur	2
1.1	Classe Terrain	2
1.2	Classe Promoteur	3
1.3	Opérations du Promoteur	4
2	Références générales	6

C++ - Le promoteur (Solution)



Mots-Clés Classes+ ■

Requis Axiomatique impérative, Classes, Classes (suite) ■

Fichiers dtterrain.txt, Rectangle ■

Difficulté ●○○ (XXX) ■



Objectif

Cet exercice réalise la gestion de terrains d'un promoteur.

1 Le promoteur / pgpromoteur

1.1 Classe Terrain

Ce problème définit la classe représentant un terrain.



Soit la classe `Rectangle` modélisant des rectangles du plan. Elle est définie dans l'exercice @[Classes géométriques] du module [Classes, instances, objets].

C++ @[Rectangle.hpp] @[Rectangle.cpp]



Écrivez une classe `Terrain` incluant :

- Un (attribut de) `Rectangle` nommé `r`.
- Un booléen `vendu` qui vaudra `Vrai` si le terrain est vendu, `Faux` sinon.



Écrivez un constructeur à trois paramètres `w` (largeur, réel), `h` (hauteur, réel) et `v` (vendu, booléen) initialisant les attributs.



Écrivez une méthode `afficher` qui affiche les caractéristiques du terrain.



Écrivez une méthode `surface` qui renvoie la surface du terrain.



Écrivez une méthode `dispo` qui renvoie `Vrai` si le terrain est disponible, `Faux` s'il est vendu.



Validez votre classe et vos méthodes avec la solution.

Solution C++ @[Terrain.hpp] @[Terrain.cpp]

```
#ifndef TERRAIN_CLASS
#define TERRAIN_CLASS
/**
 * Terrain compose d'un rectangle
 */
#include "Rectangle.hpp"
class Terrain
{
public:
    Terrain(double w, double h, bool v);
    void afficher() const;
    double surface() const;
    bool dispo() const;
private:
    Rectangle m_r;
    bool m_vendu;
};
```

```

#include "Terrain.cpp"
#endif

#include <iostream>
using namespace std;

/**
 * Constructeur definissant le terrain
 * @param[in] w - largeur du rectangle
 * @param[in] h - hauteur du rectangle
 * @param[in] v - status de la vente
 */
Terrain::Terrain(double w, double h, bool v)
: m_r(w, h), m_vendu(v)
{}

/**
 * Affiche les caracteristiques du terrain
 */
void Terrain::afficher() const
{
    cout<<"vendu = "<<not dispo()<<" surf = "<<m_r.surface()<<" ";
    m_r.afficher();
}

/**
 * Accesseur de la surface
 * @return la surface
 */
double Terrain::surface() const
{
    return m_r.surface();
}

/**
 * Accesseur de la disponibilite
 * @return la disponibilite
 */
bool Terrain::dispo() const
{
    return not m_vendu;
}

```

1.2 Classe Promoteur

Ce problème définit une classe représentant la gestion des terrains.



Écrivez une classe `Promoteur` comprenant :

- Un vecteur (tableau dynamique) `tr` de `Terrain`.



Fournissez un constructeur par défaut.



Écrivez une méthode `charger(fn)` qui charge les données depuis un nom de fichier `fn` (chaîne de caractères). Supposez les données comme suit :

```
largeur longueur carac
```

Le caractère est '`v`' pour terrain vendu et '`d`' pour terrain disponible.



Écrivez une méthode `afficher` qui affiche les caractéristiques des terrains.



Écrivez un début de programme qui :

- Instancie un `Promoteur tr`.
- Demande et saisit le nom du fichier contenant les données.
- Charge les données puis les affiche.



Téléchargez le fichier des données : @[dtterrains.txt]

```
100.0 50.0 v
200.0 70.0 v
200.0 60.0 d
150.0 60.0 d
150.0 40.0 d
250.0 60.0 v
250.0 40.0 v
```



Testez.

1.3 Opérations du Promoteur



Écrivez une méthode `calculer` qui calcule la surface des terrains vendus et celle des terrains disponibles à la construction.



Validez votre classe et vos méthodes avec la solution.

Solution C++ @[Promoteur.hpp] @[Promoteur.cpp]

```
#ifndef TPROMOTEUR_CLASS
#define TPROMOTEUR_CLASS
#include <vector>
#include <string>
using namespace std;

/**
 * Les Terrains
 */
#include "Terrain.hpp"
```

```
class Promoteur
{
public:
    Promoteur();
    void charger(const string& fn);
    void afficher() const;
    void calculer() const;
private:
    vector<Terrain> m_tr;
};
#include "Promoteur.cpp"
#endif

#include <iostream>
#include <fstream>
using namespace std;

/**
    Constructeur par default
*/
Promoteur::Promoteur()
: m_tr()
{}

/**
    Charge les donnees depuis un fichier
    @param[in] fn - nom du fichier des donnees
*/
void Promoteur::charger(const string& fn)
{
    ifstream is(fn.c_str());
    double w, h;
    char v;
    is>>w;
    while (not is.eof())
    {
        is>>h>>v;
        bool b = (v == 'v');
        m_tr.push_back(Terrain(w, h, b));
        is>>w;
    }
    is.close();
}

/**
    Affiche les caracteristiques des terrains
*/
void Promoteur::afficher() const
{
    for (int j = 0; j < m_tr.size(); ++j)
    {
        m_tr[j].afficher();
    }
}

/**
    Calcul des ventes et des disponibilites
*/
```

```

void Promoteur::calculer() const
{
    double sv = 0.0, sd = 0.0;
    for (int j = 0; j < m_tr.size(); ++j)
    {
        if (m_tr[j].dispo())
        {
            sd += m_tr[j].surface();
        }
        else
        {
            sv += m_tr[j].surface();
        }
    }
    cout<<"Surface vente "<<sv<<endl;
    cout<<"Surface dispo "<<sd<<endl;
}

```



Complétez votre programme.



Testez.



Validez votre programme avec la solution.

Solution C++ @[pgpromoteur.cpp]

```

#include <iostream>
using namespace std;

#include "Promoteur.hpp"
int main()
{
    Promoteur tr;
    cout<<"Nom du fichier? ";
    string fn;
    cin>>fn;
    tr.charger(fn);
    tr.afficher();

    tr.calculer();
}

```

2 Références générales

Comprend [Chappelier-CPP1 :c8 :ex45, ex46 ■