

Le promoteur [cm02] - Exercice

Karine Zampieri, Stéphane Rivière

Table des matières

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

Python - Le promoteur (Solution)



Mots-Clés Classes+ ■

Requis Axiomatique impérative, Classes, Classes (suite) ■

Fichiers dtterrain.txt, Rectangle ■

Difficulté •○○



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].



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 Python @[Terrain.py]

```
from Rectangle import Rectangle
class Terrain:
    def __init__(self,w,h,v):
        self.m_r = Rectangle(w,h)
        self.m_vendu = v

    def afficher(self):
        print("vendu = ",not self.dispo()," surf = ",self.m_r.surface()," ", sep="",end="")
        self.m_r.afficher()

    def surface(self):
        return self.m_r.surface()

    def dispo(self):
        return not self.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 Python @[Promoteur.py]

```
from Terrain import Terrain
class Promoteur:
    def __init__(self):
        self.m_tr = []

    def charger(self,fn):
        f = open(fn,"r")
        for ligne in f:
            s = ligne.strip("\n")
            w,h,ss = s.split(" ")
            v = ss[0]
            b = (v == 'v')
            tr = Terrain(float(w),float(h),bool(b))
            self.m_tr.append(tr)
        f.close()

    def afficher(self):
        for j in range(0,len(self.m_tr)):
            self.m_tr[j].afficher()

    def calculer(self):
        sv, sd = 0.0,0.0
        for j in range(0,len(self.m_tr)):
            if (self.m_tr[j].dispo()):
                sd += self.m_tr[j].surface()
            else:
                sv += self.m_tr[j].surface()
        print("Surface vente",sv)
        print("Surface dispo",sd)
```



Complétez votre programme.



Testez.



Validez votre script avec la solution.

Solution Python @[pgpromoteur.py]

```
from Promoteur import Promoteur
tr = Promoteur()
fn = input("Nom du fichier? ")
tr.charger(fn)
tr.afficher()
tr.calculer()
```

2 Références générales

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