



```
# -*- coding: utf-8 -*-  
"""
```

```
Created on Tue Jan 3 15:16:32 2023
```

```
@author:
```

```
    Groupe16 :
```

```
        LABULU IBAM Danny
```

```
        ONKETU ANTEMBA Beni
```

```
        KABANGU MWATA Olivier
```

```
"""
```

```
""""RESOLUTION 2""""
```

```
#classe Rectangle qui herite de la classe mère Geo_Form
```

```
class Rectangle(Geo_Form):
```

```
    try: #gestion des exceptions
```

```
        def __init__(self, nomF, longueur, largeur): #initialisation de la classe
```

```
Rectangle avec nom, long, largeur
```

```
        #initialisation des variables internes de la classe
```

```
        self.nomF = nomF
```

```
        self.longueur = longueur
```

```
        self.largeur = largeur
```

```
        #methode de calcul du perimetre
```

```
        def perimetre(self):
```

```
            return 2*self.longueur + 2*self.largeur
```

```
        #methode de calcul de la surface
```

```
        def surface(self):
```

```
            return self.longueur*self.largeur
```

```
    except: #gestion des exceptions
```

```
        print("Parametres non pris en charge")
```

```
#classe Cercle qui herite de la classe mère Geometrie_Forme
```

```
class Cercle(Geo_Form):
```

```
    try:
```

```
        def __init__(self, nomF, rayon):
```

```
            self.nomF = nomF
```

```
            self.rayon = rayon
```

```
        def perimetre(self):
```

```
            return 2*pi *self.rayon
```

```
        def surface(self):
```

```
            return pi *(self.rayon**2)
```

```
    except:
```

```
        print("Parametres non pris en charge")
```

```
#classe Triangle qui herite de la classe mère Geo_Form
```

```
class Triangle(Geo_Form):
```

```
    try:
```

```
        def __init__(self, nomF, CA, CB, CC):
```

```
            self.nomF = nomF
```

```
            self.CB = CB
```

```
            self.CA = CA
```

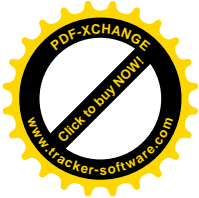
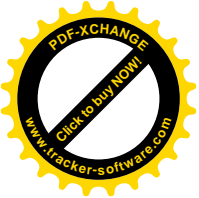
```
            self.CC = CC
```

```
        def perimetre(self):
```

```
            return self.CB + self.CA + self.CC
```

```
        def surface(self):
```

```
            p = self.perimetre()/2
```



```
aire = sqrt(p*(p - self.CA)*(p - self.CB)*(p - self.CC))
aire = aire.real
return aire
except:
    print("Parametres non pris en charge")
```