PDF-XCHANGE Click to buy NOW! www.tracker-software.com
PDF-XCHANGE Click to buy NOW! www.tracker-software.com

```
# -*- coding: utf-8 -*-
"""
Created on Tue Jan  3 15:16:32 2023
@author:
    Groupe16 :
        LABULU IBAM Danny
        ONKETU ANTEMBA Beni
       KABANGU MWATA Olivier
"""
"""RESOLUTION8"""
from abc import ABCMeta, abstractmethod
from math import pi, sqrt

class Geo_Form(metaclass = ABCMeta):
    @abstractmethod
    def perimetre():
        pass

    @abstractmethod
    def surface():
        pass

class Rectangle(Geo_Form):
    try:
        def __init__(self,nomF, longueur, largeur):
            self.nomF = nomF
            self.longueur = longueur
            self.largeur = largeur
        def perimetre(self):
            return 2*self.longueur + 2*self.largeur

        def surface(self):
            return self.longueur*self.largeur
    except:
        print("Parametres non pris en charge")

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")

class Triangle(Geo_Form):
    try:
        def __init__(self,nomF, CA,CB,CC):
            self.nomF = nomF
            self.CB = CB
            self.CA = CA
            self.CC = CC
```

PDF-XCHANGE Click to buy NOW! www.tracker-software.com
PDF-XCHANGE Click to buy NOW! www.tracker-software.com

```
        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")
class Carre(Rectangle):
    try:
        def __init__(self,nomF, cote):
            Rectangle.__init__(self, nomF, cote, cote)
    except:
        print("Parametres non pris en charge ")

class TriangleRectangle(Triangle):
    try:
        def __init__(self,nomF, base, hauteur):
            hyp = sqrt(base**2+hauteur**2)
            Triangle.__init__(self, nomF, base, hauteur, hyp)
    except:
        print("Parametres non pris en charge ")

class GeoFig():
    try:
        def __init__(self):
            self.KGeo_rep = []
        def add(self, fig):
            self.KGeo_rep.append(fig)

    except:
        print("Parametres non pris en charge")
#utilisation du polymorphisme
def tout_perimetre(obj):
    return obj.perimetre()

def tout_superficie(obj):
    return obj.surface()
```