

Module : POO Python (M103) TP N° 5

Année de Formation 2023/2024

Filière : Développement digital

Groupe : DEV 101 - DEV 102

Niveau : 1ère année

Exercice 1 :

On suppose que vous développez un système de calcul des superficies et des périmètres pour différentes formes géométriques.

Les deux formes géométriques que vous souhaitez prendre en charge sont les cercles et les carrés.

1. Créez une classe abstraite **`FormeGeometrique`** qui définit deux méthodes abstraites :

- **`calculer_superficie()`** : Cette méthode devrait retourner la superficie de la forme géométrique.

- **`calculer_perimetre()`** : Cette méthode devrait retourner le périmètre de la forme géométrique.

2. Créez deux classes concrètes, **`Cercle`** et **`Carre`**, qui héritent de la classe **`FormeGeometrique`**.

3. Dans la classe **`Cercle`**, implémentez les méthodes **`calculer_superficie()`** et **`calculer_perimetre()`** en utilisant les formules appropriées pour un cercle.

4. Dans la classe **`Carre`**, implémentez les méthodes **`calculer_superficie()`** et **`calculer_perimetre()`** en utilisant les formules appropriées pour un carré.

5. Créez des instances de la classe **`Cercle`** et de la classe **`Carre`**, puis appelez les méthodes **`calculer_superficie()`** et **`calculer_perimetre()`** pour chaque instance.

6. Affichez les résultats pour les cercles et les carrés.

7. Assurez-vous que le programme ne génère pas d'erreur lorsque vous appelez les méthodes abstraites.

Exercice 2 :

vous développez un système de gestion de comptes bancaires. Vous avez deux types de comptes : un compte d'épargne et un compte courant. Les deux types de comptes partagent certaines fonctionnalités, mais ils ont également des fonctionnalités spécifiques. Vous devez utiliser une méthode abstraite pour gérer les fonctionnalités communes et une méthode non abstraite pour gérer les fonctionnalités spécifiques à chaque type de compte.

1. Créez une classe abstraite `CompteBancaire` qui définit une méthode abstraite `calculer_interets()`. Cette méthode abstraite doit être utilisée pour calculer les intérêts sur le solde du compte. Vous pouvez utiliser une valeur arbitraire pour les intérêts, par exemple, 1%.
2. Créez deux sous-classes de `CompteBancaire` : `CompteEpargne` et `CompteCourant`.
3. Dans la classe `CompteEpargne`, implémentez la méthode `calculer_interets()` pour calculer les intérêts en multipliant le solde par 1% (0.01).
4. Dans la classe `CompteCourant`, implémentez la méthode `calculer_interets()` pour afficher un message indiquant que les comptes courants n'ont pas d'intérêts.
5. Créez des instances de `CompteEpargne` et `CompteCourant`, déposez de l'argent dans les comptes, puis appelez la méthode `calculer_interets()` pour chaque compte.