# Q1

Quel type de méthode serait approprié ? Méthode de classe, d’instance ou statique?

|  |  |
| --- | --- |
|  | Type de méthode |
| Votre méthode doit fonctionner sur ou modifier des données spécifiques à l'instance | D’instance |
| Votre méthode doit accéder à des données spécifiques de la classe | Classe |
| Votre méthode ne dépend pas de l’état de l’instance ni de la classe | Statique |

# Q2

Voici une classe CompteBancaire.

classe CompteBancaire :

def \_\_init\_\_(self, compte\_holder, balance=0) :

self. compte\_holder = compte\_holder

self.balance = balance

Définissez 2 méthodes d’instances :

* **deposer** : permet d’ajouter un montant à la balance du compte d’une personne. Puis imprime qu’un dépôt a été fait, le montant de la transaction et le nouveau solde
* **retirer** : permet de soustraire un montant à la balance du compte d’une personne s’il y a assez d’argent dans le compte. Puis imprime qu’un retrait a été fait, le montant de la transaction et le nouveau solde s’il n’y a pas assez d’argent dans le compte, elle imprime le fait que les fonds sont insuffisants..

# Q3

Si vous voulez ajouter une méthode nommé **aubaine** qui va imprimer un message comme quoi « Épargner 50$ par mois vous donnera un montant de plus de 20 000$ dans 20 ans! », quel type de méthode allez-vous faire?\_instance\_\_\_\_\_\_\_\_\_\_\_\_

Ajoutez cette méthode à la classe CompteBancaire.

Ajoutez une variable de classe listeTransactions qui sera la liste des transactions faites à la classe CompteBancaire. Ajoutez aussi une méthode de classe qui imprime cette liste de transactions. Comme :

12 :45 :12 ‘dépôt de 500$’

12 :52 :64 ‘retrait de 200$’

Une méthode privée \_\_heure\_maintenant() vous ait donné.

Modifiez les méthodes deposer et retirer pour qu’elles ajoutent la transaction à la variable de classe

# Réponse :

## Q1

Méthode de classe, d’instance ou statique?

|  |  |
| --- | --- |
|  | Type de méthode |
| Votre méthode doit fonctionner sur ou modifier des données spécifiques à l'instance | D’instance |
| Votre méthode doit accéder à des données spécifiques de la classe | De classe |
| Votre méthode ne dépend pas de l’état de l’instance ni de la classe | Statique |

## Q2

**Réponse :**

classe CompteBancaire :

def \_\_init\_\_(self, compte\_holder, balance=0) :

self. compte\_holder = compte\_holder

self.balance = balance

def deposer (self, montant) :

self.balance += montant

print(f"${montant} déposé. Nouveau solde : ${self.balance}")

def retirer (self, montant):

if montant <= self.balance :

self.balance -= montant

print(f"Retrait de ${montant}. Nouveau solde : ${self.balance}")

else:

print (f"Fonds insuffisants. Vous essayez de retirer ${montant} mais le solde actuel est de: ${self.balance}")

# Utilisation

compte = CompteBancaire(compte\_holder ="Alice", balance=1000)

compte.deposer(500)

compte.retirer(200)

Dans cet exercice, les méthodes **deposer** et **retirer** sont des méthodes d'instance car elles fonctionnent sur une instance de compte bancaire spécifique. Ils ont accès et peuvent modifier l'équilibre des attributs spécifiques à l'instance.