

Partie 1 : Gestion des Banques

Objectif :

Concevoir un système de gestion bancaire complet qui applique les principes de la programmation orientée objet (POO) afin de gérer les comptes et les opérations des clients.

Principales fonctionnalités :

- **Comptes Bancaires (Compte.py) :**
La classe de base gère le solde, la limite quotidienne de retrait et le type de compte (Bronze, Or, Affaires) selon la limite estimée. Elle offre des méthodes pour déposer, retirer et transférer des fonds tout en enregistrant l'historique des transactions.
- **Cartes Bancaires (Carte.py) :**
Chaque client peut se voir attribuer une carte bancaire générée aléatoirement. La classe gère la définition et la vérification du code PIN. En cas de trois tentatives erronées, la carte est bloquée, garantissant ainsi la sécurité de l'opération.
- **Clients Utilisateurs (ClientUtilisateur.py) :**
Ce module représente les clients particuliers. Il utilise la composition en associant un compte bancaire et une ou plusieurs cartes. Il inclut les opérations de dépôt, retrait (avec vérification du PIN) et transfert vers d'autres comptes.
- **Clients Entreprises (ClientEntreprise.py) :**
Ici, le compte bancaire est partagé entre plusieurs employés d'une entreprise. Le PDG ouvre le compte, puis des employés peuvent y accéder. En plus des opérations classiques, la classe permet de demander des prêts et gère une liste d'employés liés au compte.
- **Employé Bancaire (EmployeBancaire.py) :**
L'employé de la banque se connecte via un identifiant et peut consulter la liste des comptes clients, approuver ou rejeter des demandes de compte, consulter l'historique des transactions et, si nécessaire, geler ou fermer un compte.
- **Transactions (Transaction.py) :**
Chaque opération (dépôt, retrait, transfert) est encapsulée dans un objet Transaction, qui enregistre notamment la date, le montant, le type de transaction et, en cas de transfert, l'identifiant du compte destinataire.

Structure et fonctionnement :

Le fichier principal `main.py` dans le dossier `banque/` simule des opérations bancaires en créant des instances de clients, en effectuant des dépôts, des retraits et des transferts, tout en illustrant le rôle de l'employé bancaire. Les données (clients, entreprises, transactions, cartes) sont prévues pour être stockées dans des fichiers textes dans le dossier `data/`.

Partie 2 : Jeu du Morpion

Objectif :

Implémenter le classique jeu du morpion (tic-tac-toe) en appliquant les principes de la POO, en mettant l'accent sur l'abstraction, l'encapsulation et la gestion des interactions entre objets.

Principales fonctionnalités :

- **Grille de Jeu (Board.py) :**
La classe gère la grille 3x3 du morpion, permettant de placer les symboles, d'afficher la grille, de vérifier si la grille est pleine ou si un joueur a gagné.
- **Joueurs (Player.py) :**
Chaque joueur est défini par un nom et un symbole (X ou O). La méthode pour obtenir le coup du joueur se base sur une saisie console, garantissant une interaction simple et directe.
- **Gestion de la Partie (Game.py) :**
Cette classe orchestre le déroulement du jeu. Elle initialise la partie avec deux joueurs, gère les tours de jeu, vérifie les conditions de victoire ou d'égalité et bascule entre les joueurs après chaque coup.

Structure et fonctionnement :

Le fichier principal `main.py` du dossier `morpion/` invite les joueurs à entrer leurs noms, crée les instances de `Player` et lance la partie via la classe `Game`. Le jeu se déroule en boucle jusqu'à ce qu'un joueur gagne ou que la grille soit entièrement remplie.

Conclusion Personnalisée

Ce projet combine deux applications distinctes pour démontrer la maîtrise de la POO : un système bancaire complet et un jeu interactif du morpion. Chaque partie est conçue pour mettre en évidence des concepts clés tels que l'héritage, la composition, la gestion des erreurs et la sécurité (notamment avec le système de vérification du PIN pour les transactions bancaires). De plus, la structure modulaire facilite l'évolution du projet, que ce soit par l'ajout de nouvelles fonctionnalités ou par l'amélioration de l'interface utilisateur.

Ce dépôt GitHub, organisé en deux dossiers principaux (`banque/` et `morpion/`), permet de maintenir une séparation claire entre les deux applications tout en partageant des concepts communs de bonne conception en programmation orientée objet. Ce projet offre ainsi une base solide pour développer des systèmes complexes tout en assurant une maintenance et une évolutivité futures.