

# Mini-projet La banque

Christophe Viroulaud

Terminale - NSI

**Lang 04**

La Banque des Périgourdins (BdP) est une institution nouvellement implantée en Dordogne. Outre un compte courant classique, elle propose également des comptes rémunérés à différents taux. À une date fixée, la banque calcule les intérêts et crédite les comptes rémunérés des clients. Elle vous demande de créer un logiciel pour administrer correctement sa clientèle.

modularite.zip sur site

La *Banque des Périgourdins (BdP)* est une institution nouvellement implantée en Dordogne. Outre un compte courant classique, elle propose également des comptes rémunérés à différents taux. À une date fixée, la banque calcule les intérêts et crédite les comptes rémunérés des clients. Elle vous demande de créer un logiciel pour administrer correctement sa clientèle.

# Sommaire

1. Cahier des charges

2. Programmation modulaire

3. Modélisation

4. Implémentation

**À retenir**

Pour pouvoir modéliser le projet il faut s'appuyer sur un *cahier des charges*. Celui-ci doit être le plus précis possible.

# Cahier des charges

**À retenir**

Pour pouvoir modéliser le projet il faut s'appuyer sur un *cahier des charges*. Celui-ci doit être le plus précis possible.

► Un **compte** possède une somme d'argent, initialisée à 0 à l'ouverture. Ce compte peut être rémunéré ou non. Ce compte peut être crédité ou débité. En cas de débit supérieur à la somme présente sur le compte, le retrait est refusé.

- Un **compte** possède une somme d'argent, initialisée à 0 à l'ouverture. Ce compte peut être rémunéré ou non. Ce compte peut être crédité ou débité. En cas de débit supérieur à la somme présente sur le compte, le retrait est refusé.

- Un **compte** possède une somme d'argent, initialisée à 0 à l'ouverture. Ce compte peut être rémunéré ou non. Ce compte peut être crédité ou débité. En cas de débit supérieur à la somme présente sur le compte, le retrait est refusé.
- Un **client** possède un ou plusieurs **comptes** rémunérés ou non. Il peut alimenter ou utiliser l'argent sur son compte. Il peut également décider d'ouvrir un nouveau compte en demandant un nouveau numéro à la **banque**.

- Un **compte** possède une somme d'argent, initialisée à 0 à l'ouverture. Ce compte peut être rémunéré ou non. Ce compte peut être crédité ou débité. En cas de débit supérieur à la somme présente sur le compte, le retrait est refusé.
- Un **client** possède un ou plusieurs **comptes** rémunérés ou non. Il peut alimenter ou utiliser l'argent sur son compte. Il peut également décider d'ouvrir un nouveau compte en demandant un nouveau numéro à la **banque**.

- Un **compte** possède une somme d'argent, initialisée à 0 à l'ouverture. Ce compte peut être rémunéré ou non. Ce compte peut être crédité ou débité. En cas de débit supérieur à la somme présente sur le compte, le retrait est refusé.
- Un **client** possède un ou plusieurs **comptes** rémunérés ou non. Il peut alimenter ou utiliser l'argent sur son compte. Il peut également décider d'ouvrir un nouveau compte en demandant un nouveau numéro à la **banque**.
- La **banque** possède une liste de tous ses clients. Elle peut enregistrer un nouveau **client**. Dans ce cas elle lui ouvre automatiquement un **compte** non rémunéré. Enfin elle peut également appliquer un taux de rémunération de 3% à tous les comptes rémunérés.

- Un **compte** possède une somme d'argent, initialisée à 0 à l'ouverture. Ce compte peut être rémunéré ou non. Ce compte peut être crédité ou débité. En cas de débit supérieur à la somme présente sur le compte, le retrait est refusé.
- Un **client** possède un ou plusieurs **comptes** rémunérés ou non. Il peut alimenter ou utiliser l'argent sur son compte. Il peut également décider d'ouvrir un nouveau compte en demandant un nouveau numéro à la **banque**.
- La **banque** possède une liste de tous ses clients. Elle peut enregistrer un nouveau **client**. Dans ce cas elle lui ouvre automatiquement un **compte** non rémunéré. Enfin elle peut également appliquer un taux de rémunération de 3% à tous les comptes rémunérés.

# Sommaire

1. Cahier des charges

2. **Programmation modulaire**

3. Modélisation

4. Implémentation



# Programmation modulaire

- Un projet important implique plusieurs programmeurs.  
On le découpe en plusieurs objets.

- Un projet important implique plusieurs programmeurs. On le découpe en plusieurs objets.
- Chaque personne n'a pas besoin de connaître l'implémentation de chaque objet. Ce dernier lui fournit par contre une **interface** qui définit ce que peut faire l'objet.

# Programmation modulaire

- Un projet important implique plusieurs programmeurs. On le découpe en plusieurs objets.
- Chaque personne n'a pas besoin de connaître l'implémentation de chaque objet. Ce dernier lui fournit par contre une **interface** qui définit ce que peut faire l'objet.

- Un projet important implique plusieurs programmeurs. On le découpe en plusieurs objets.
- Chaque personne n'a pas besoin de connaître l'implémentation de chaque objet. Ce dernier lui fournit par contre une **interface** qui définit ce que peut faire l'objet.
- La **classe** garantit ce que va faire l'objet (mais pas comment le programmeur va s'en servir).

## Programmation modulaire

- Un projet important implique plusieurs programmeurs. On le découpe en plusieurs objets.
- Chaque personne n'a pas besoin de connaître l'implémentation de chaque objet. Ce dernier lui fournit par contre une **interface** qui définit ce que peut faire l'objet.
- La **classe** garantit ce que va faire l'objet (mais pas comment le programmeur va s'en servir).

**À retenir**

On crée une classe dans un fichier séparé. La classe, ainsi que chacune de ses méthodes est complétée d'une `docstring`. Elle servira d'**interface**.

**Activité 1 :**

- ▶ Télécharger le dossier compressé `modularite.zip` et extraire les fichiers.
- ▶ Lancer le fichier `modularite_poo.py`.
- ▶ Observer l'aide affichée.

**À retenir**

On crée une classe dans un fichier séparé. La classe, ainsi que chacune de ses méthodes est complétée d'une `docstring`. Elle servira d'**interface**.

**Activité 1 :**

- ▶ Télécharger le dossier compressé `modularite.zip` et extraire les fichiers.
- ▶ Lancer le fichier `modularite_poo.py`.
- ▶ Observer l'aide affichée.

- 1. Cahier des charges
- 2. Programmation modulaire
- 3. **Modélisation**
- 4. Implémentation

# Sommaire

1. Cahier des charges

2. Programmation modulaire

3. **Modélisation**

4. Implémentation

**Activité 2** : Réfléchir en commun pour modéliser le concept de la banque. Définir les objets nécessaires et les actions possibles (cahier des charges page suivante).

# Modélisation

**Activité 2** : Réfléchir en commun pour modéliser le concept de la banque. Définir les objets nécessaires et les actions possibles (cahier des charges page suivante).

- ▶ Un **compte** possède une somme d'argent, initialisée à 0 à l'ouverture. Ce compte peut être rémunéré ou non. Ce compte peut être crédité ou débité. En cas de débit supérieur à la somme présente sur le compte, le retrait est refusé.
- ▶ Un **client** possède un ou plusieurs **comptes** rémunérés ou non. Il peut alimenter ou utiliser l'argent sur son compte. Il peut également décider d'ouvrir un nouveau compte en demandant un nouveau numéro à la **banque**.
- ▶ La **banque** possède une liste de tous ses clients. Elle peut enregistrer un nouveau **client**. Dans ce cas elle lui ouvre automatiquement un **compte** non rémunéré. Enfin elle peut également appliquer un taux de rémunération de 3% à tous les comptes rémunérés.

- ▶ Un **compte** possède une somme d'argent, initialisée à 0 à l'ouverture. Ce compte peut être rémunéré ou non. Ce compte peut être crédité ou débité. En cas de débit supérieur à la somme présente sur le compte, le retrait est refusé.
- ▶ Un **client** possède un ou plusieurs **comptes** rémunérés ou non. Il peut alimenter ou utiliser l'argent sur son compte. Il peut également décider d'ouvrir un nouveau compte en demandant un nouveau numéro à la **banque**.
- ▶ La **banque** possède une liste de tous ses clients. Elle peut enregistrer un nouveau **client**. Dans ce cas elle lui ouvre automatiquement un **compte** non rémunéré. Enfin elle peut également appliquer un taux de rémunération de 3% à tous les comptes rémunérés.

- 1. Cahier des charges
- 2. Programmation modulaire
- 3. Modélisation
- 4. Implémentation

# Sommaire

1. Cahier des charges

2. Programmation modulaire

3. Modélisation

4. Implémentation



**Activité 3 :**

- ▶ Partager les tâches à effectuer et construire les différentes classes.
- ▶ Construire un programme qui crée la banque et plusieurs clients.

# Implémentation

**Activité 3 :**

- ▶ Partager les tâches à effectuer et construire les différentes classes.
- ▶ Construire un programme qui crée la banque et plusieurs clients.