

Object-Oriented Programming in Python: Inheritance and Overriding

Introduction

In this tutorial, you will be designing and implementing a simplified banking system using object-oriented programming in Python. This system will consist of classes that represent different types of bank accounts. Through this exercise, you will practice applying concepts such as classes, inheritance, method overriding, and the use of `super()`.

Project Requirements

Your task is to create a Python program that models a banking system. The system should allow for the creation of different types of accounts, processing deposits and withdrawals, and applying interest where applicable.

BankAccount Class

Start by designing a base class named `BankAccount`. This class should include the following:

- Attributes for the account holder's name, account number, and balance.
- A constructor (`__init__` method) that initializes the account holder's name, account number, and balance (defaulting to 0).
- Methods for depositing and withdrawing funds.
- A method to display the account details.

SavingsAccount and CheckingAccount Classes

Next, extend the `BankAccount` class to create two subclasses: `SavingsAccount` and `CheckingAccount`.

SavingsAccount

The `SavingsAccount` class should:

- Inherit from `BankAccount`.
- Include an additional attribute for the interest rate.
- Override the constructor to include the interest rate.
- Include a method to apply interest to the balance.

CheckingAccount

The `CheckingAccount` class should:

- Inherit from `BankAccount`.
- Include an additional attribute for the minimum balance required.
- Override the constructor to include the minimum balance.
- Override the withdrawal method to check that the withdrawal will not drop the balance below the minimum balance required. If it does, the withdrawal should be denied.

Additional Challenges

For those looking for more advanced features to implement, consider adding the following:

- Transaction history tracking for each account.
- A method to transfer funds between accounts.

Conclusion

This project will help solidify your understanding of object-oriented programming concepts in a practical, real-world application. By designing and implementing a banking system, you will gain hands-on experience with class inheritance, method overriding, and encapsulation. Good luck, and have fun with this coding exercise!