Exercise 3 - Inheritance

1. Create a base class called `BankAccount` with attributes like `accountNumber`, `accountHolderName`, and `balance`. Implement methods for depositing and withdrawing money. Then, create two subclasses, `SavingsAccount` and `CheckingAccount`, which inherit from the `BankAccount` class. Add specific attributes and methods to each subclass. For example, `SavingsAccount` might have an `interestRate` attribute, and `CheckingAccount` could have a `monthlyFee` attribute. Demonstrate how you can create instances of these subclasses and perform transactions.

2. Create a base class called `Shape` with an abstract method `calculateArea()`. Define subclasses like `Circle`, `Rectangle`, and `Triangle`, which inherit from the `Shape` class. Implement the `calculateArea()` method in each subclass to calculate the area of the respective shape. Demonstrate polymorphism by creating an array of `Shape` objects, including instances of each subclass, and calculate and display their areas.

3. Create a base class called `Employee` with attributes like `name`, `employeeId`, and `salary`. Implement a method called `calculateSalary()` in the `Employee` class that calculates the annual salary. Then, create subclasses like `Manager` and `Engineer`, which inherit from the `Employee` class. Override the `calculateSalary()` method in each subclass to include specific calculations. For example, `Manager` might include a bonus calculation, and `Engineer` could have an overtime pay calculation. Demonstrate how you can create instances of these subclasses and calculate their salaries.