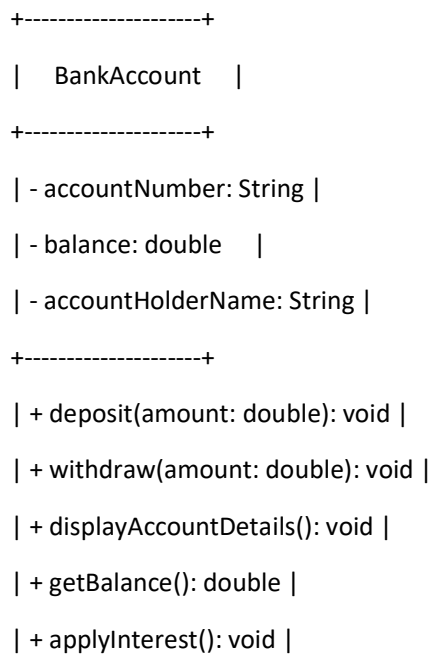
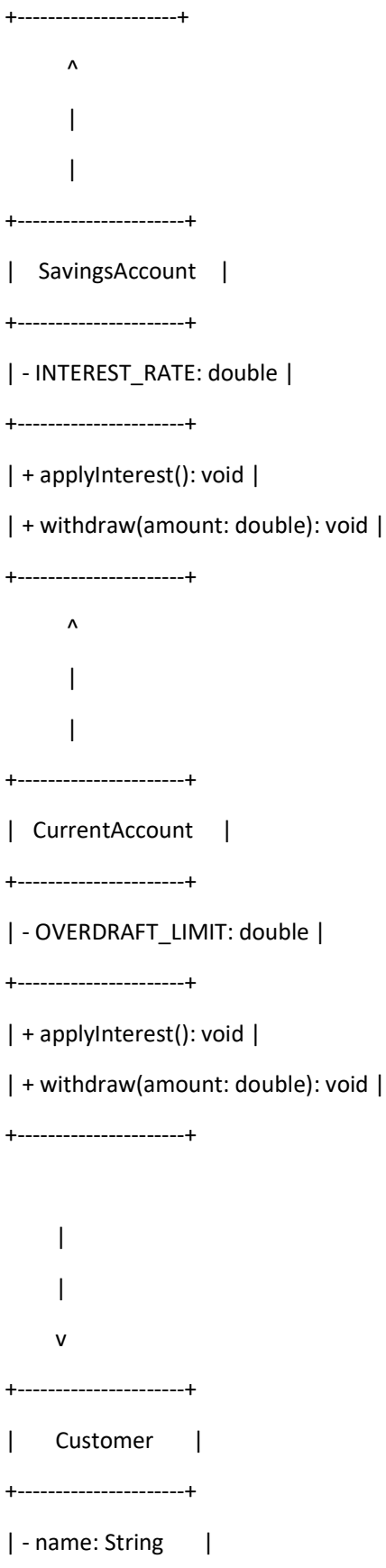
Banking System:---

1. **BankAccount Class** (Base Class):
   - Defines the common properties and methods for all account types.
   - The `applyInterest` method is abstract (in this case, does nothing) and is meant to be overridden by subclasses.
2. **SavingsAccount Class** (Subclass of `BankAccount`):
   - Inherits from `BankAccount` and implements the `applyInterest` method, adding interest based on a fixed rate.
   - Overrides the `withdraw` method to limit withdrawals (e.g., to prevent overdrawing the account).
3. **CurrentAccount Class** (Subclass of `BankAccount`):
   - Inherits from `BankAccount` but provides no interest (as typical for current accounts).
   - Implements the `withdraw` method with an **overdraft facility** allowing withdrawals even if the balance goes below zero, as long as it doesn't exceed the overdraft limit.
4. **Customer Class**:
   - Manages the customer's information and handles the creation of both `SavingsAccount` and `CurrentAccount`.
   - Displays the customer's accounts and allows interaction with each account.
5. **BankingSystem Class** (Main class):
   - Handles user input to create accounts, deposit money, withdraw money, view account details, and apply interest.
   - Supports multiple actions for each type of account.




^
SavingsAccount
- INTEREST_RATE: double
+ applyInterest(): void
+ withdraw(amount: double): void
^
CurrentAccount
- OVERDRAFT_LIMIT: double
+ applyInterest(): void
+ withdraw(amount: double): void
v
Customer
- name: String

| - email: String |

| - accounts: List<BankAccount> |

+----------+

| + createSavingsAccount(accountNumber: String, initialBalance: double): void |

| + createCurrentAccount(accountNumber: String, initialBalance: double): void |

| + displayCustomerAccounts(): void |

| + getAccounts(): List<BankAccount> |

+----------+

Explanation:

1. **BankAccount Class**:
   - This is the **base class** that contains common attributes and methods for all types of bank accounts, such as `accountNumber`, `balance`, and `accountHolderName`.
   - It has methods like `deposit()`, `withdraw()`, `displayAccountDetails()`, `getBalance()`, and an abstract method `applyInterest()` that is meant to be overridden by subclasses.
2. **SavingsAccount Class**:
   - Inherits from the `BankAccount` class.
   - Adds a **static final** attribute `INTEREST_RATE` for the interest rate.
   - Overrides `applyInterest()` to apply interest based on the `INTEREST_RATE`.
   - Overrides `withdraw()` to add specific logic for withdrawing from a savings account (e.g., withdrawal limits).
3. **CurrentAccount Class**:
   - Also inherits from the `BankAccount` class.
   - Adds a **static final** attribute `OVERDRAFT_LIMIT` for the overdraft limit.
   - Overrides `applyInterest()` to provide no interest for current accounts.
   - Overrides `withdraw()` to allow withdrawals up to the overdraft limit.
4. **Customer Class**:
   - Represents a customer who can have multiple accounts (both savings and current).
   - Contains attributes like `name`, `email`, and a list of `BankAccount` objects.
   - Includes methods to create `SavingsAccount` and `CurrentAccount`, display customer accounts, and interact with the list of accounts.