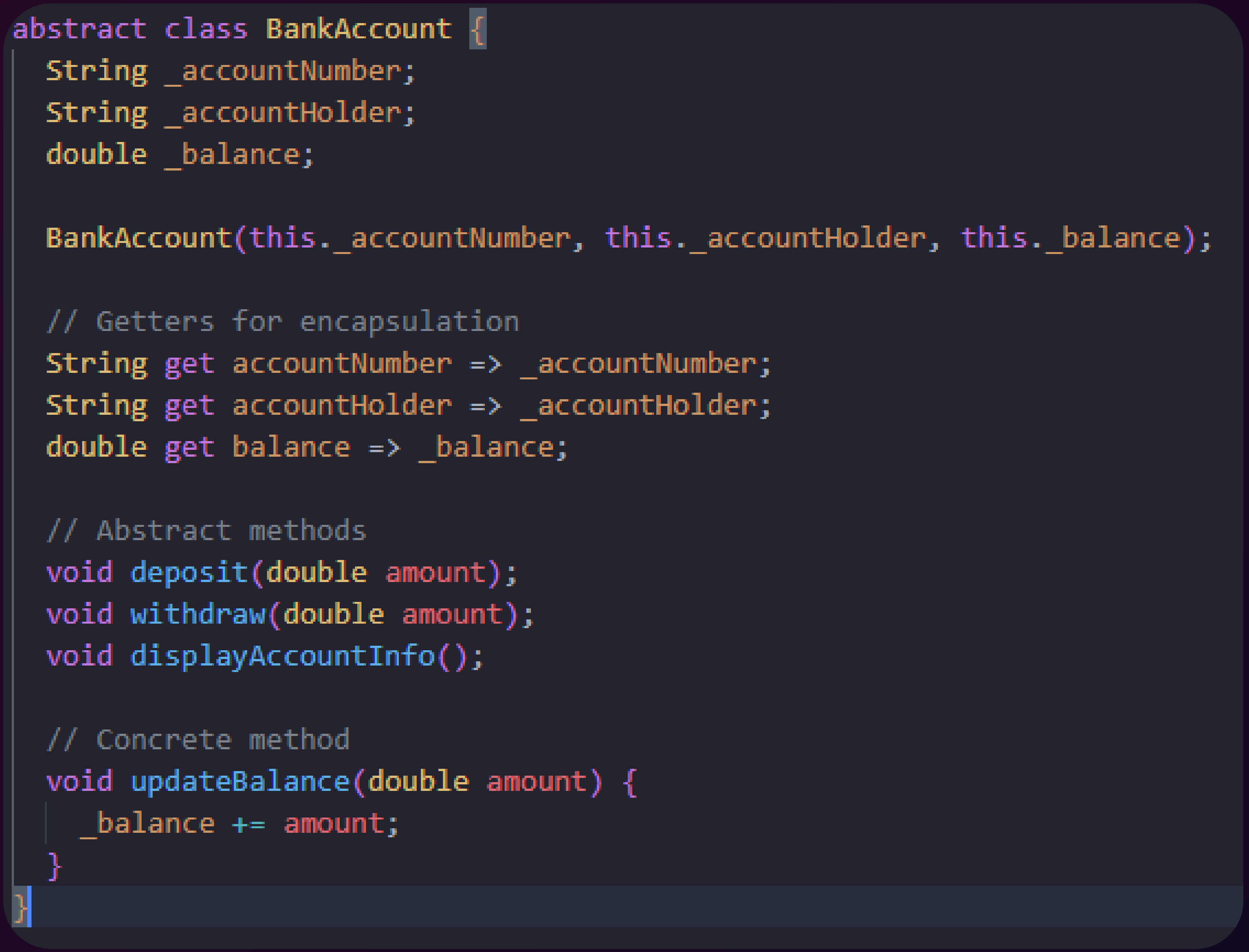
Bank Management System

Salma Elmaghawry

Bank Class

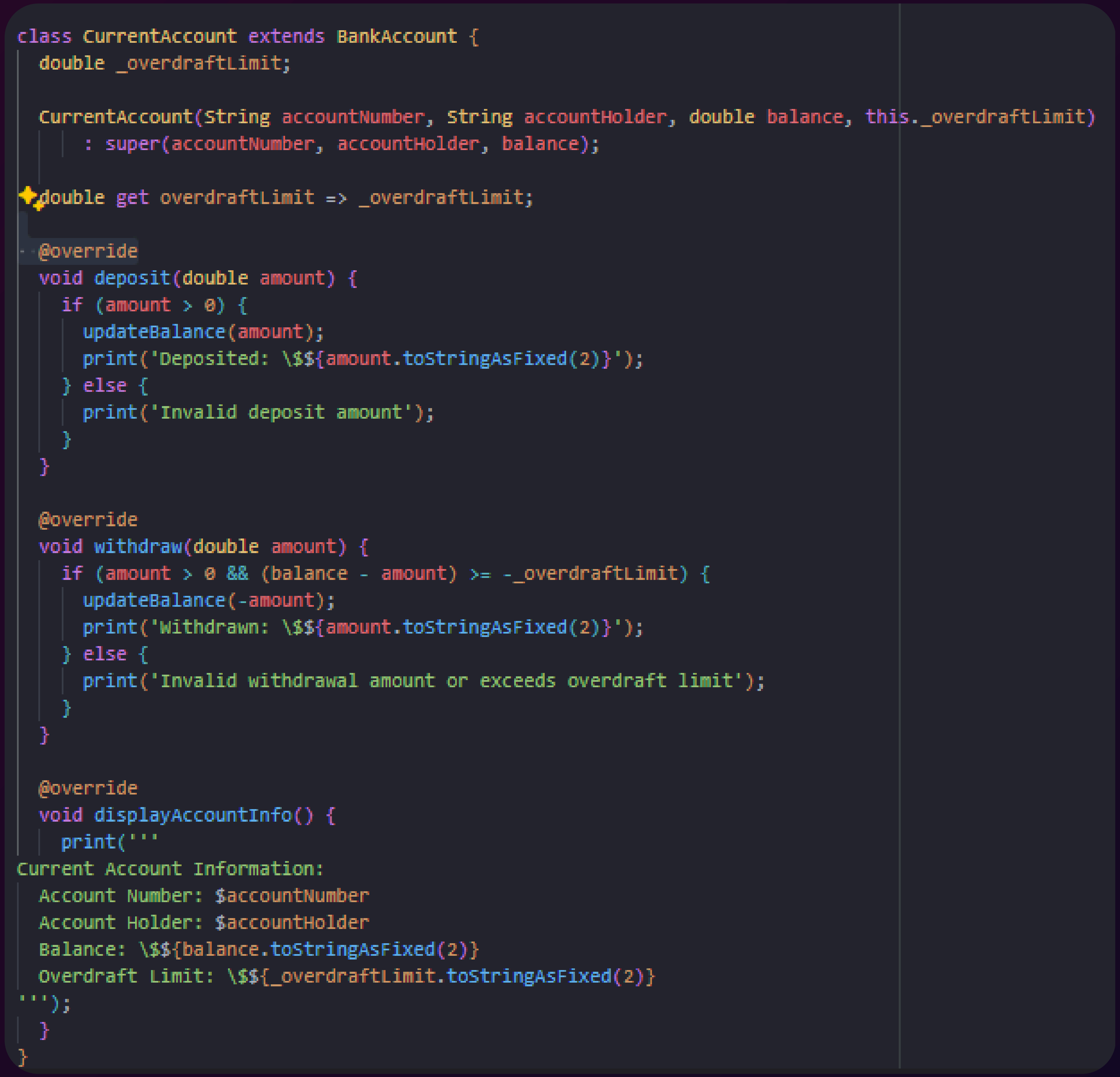
```
class Bank {
  List<BankAccount> _accounts = [];

  void addAccount(BankAccount account) {
    _accounts.add(account);
    print('Account ${account.accountNumber} added successfully.');
  }

  BankAccount? findAccount(String accountNumber) {
    for (var account in _accounts) {
      if (account.accountNumber == accountNumber) {
        return account;
      }
    }
    return null;
  }

  void displayAllAccounts() {
    if (_accounts.isEmpty) {
      print('No accounts in the bank.');
      return;
    }

    print('\nAll Bank Accounts:');
    for (var account in _accounts) {
      account.displayAccountInfo();
    }
  }
}
```

BankAccount Class



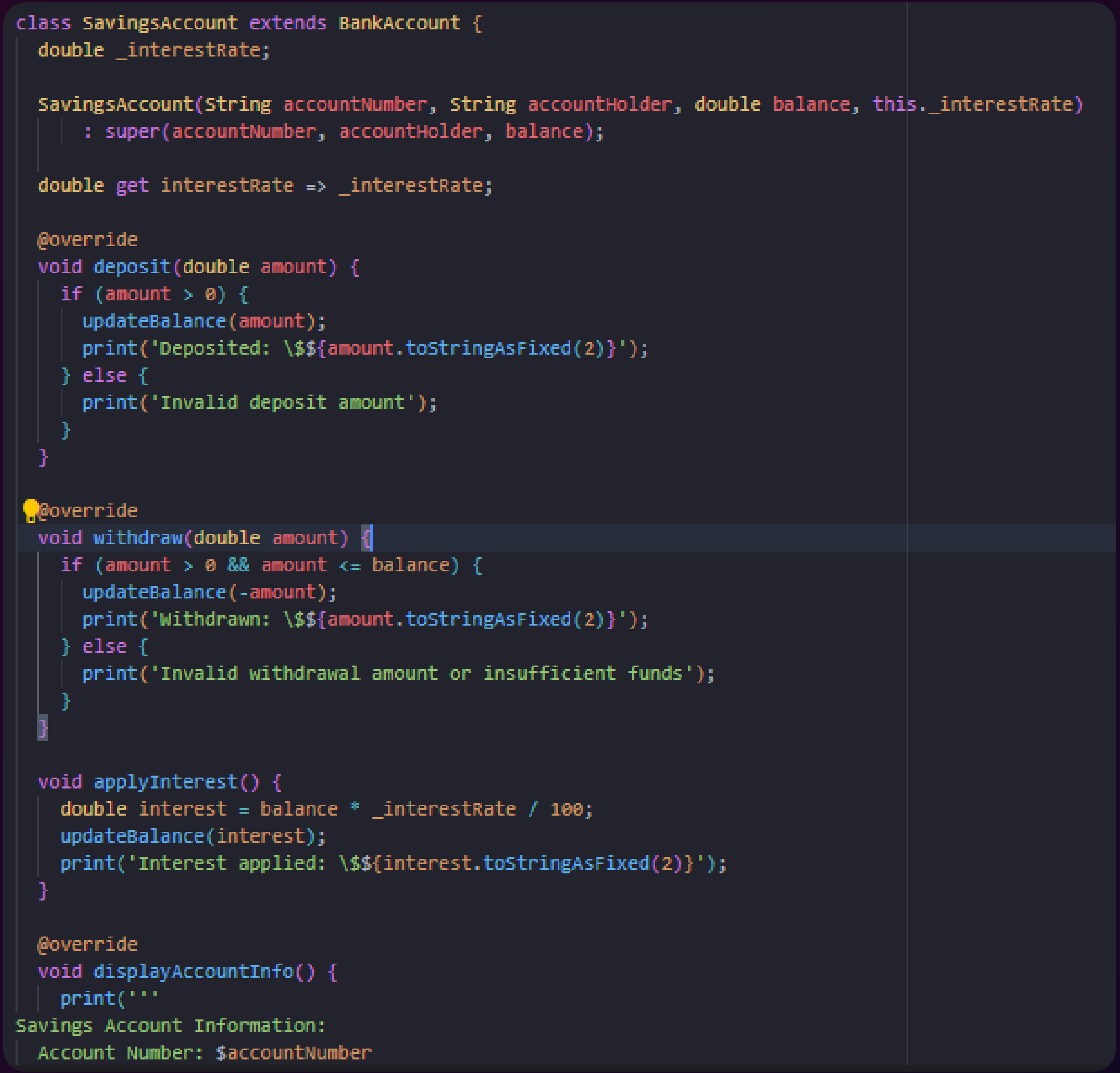
```
abstract class BankAccount {
  String _accountNumber;
  String _accountHolder;
  double _balance;

  BankAccount(this._accountNumber, this._accountHolder, this._balance);

  // Getters for encapsulation
  String get accountNumber => _accountNumber;
  String get accountHolder => _accountHolder;
  double get balance => _balance;

  // Abstract methods
  void deposit(double amount);
  void withdraw(double amount);
  void displayAccountInfo();

  // Concrete method
  void updateBalance(double amount) {
    _balance += amount;
  }
}
```

Current Account Class



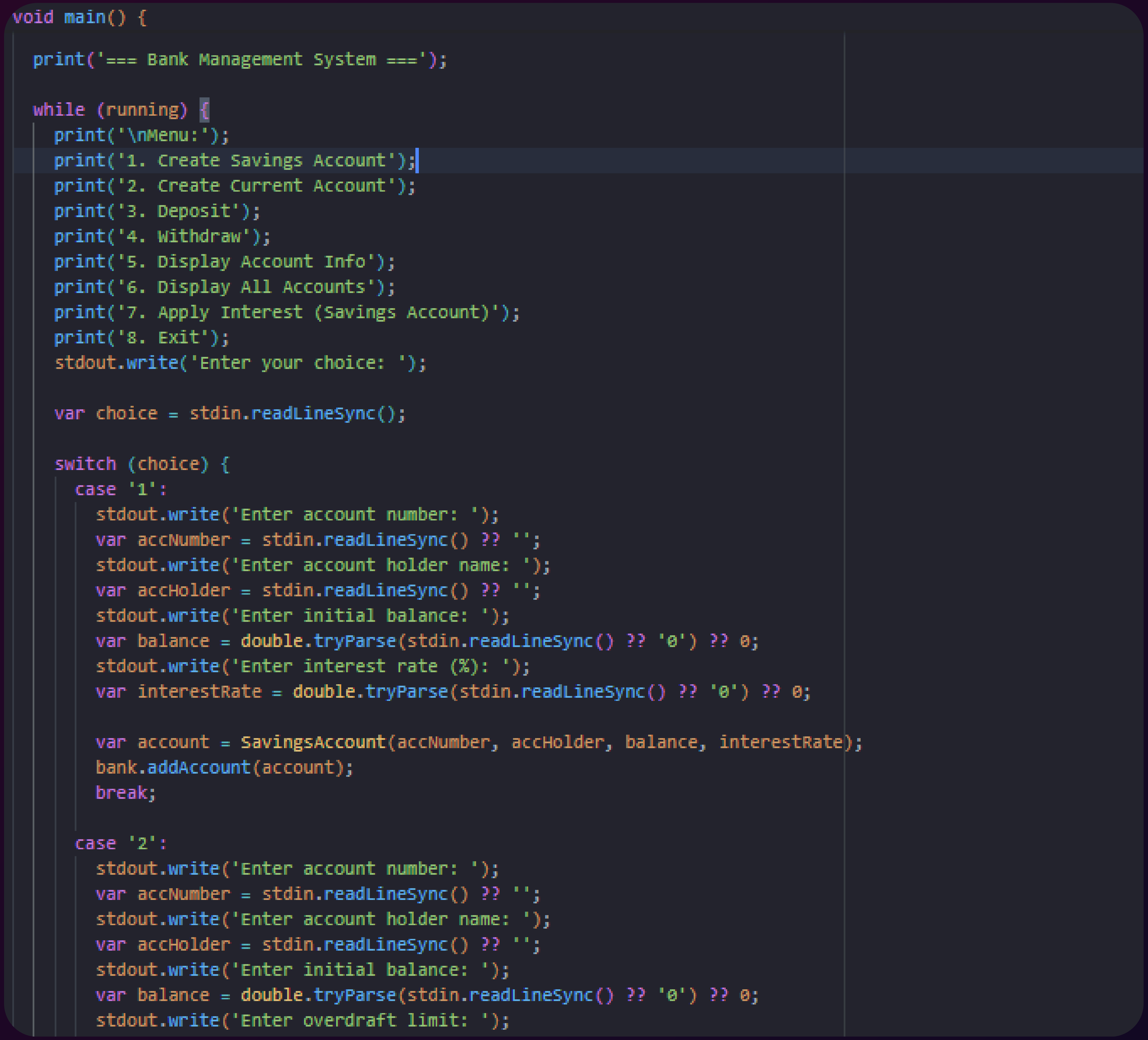
```dart
class CurrentAccount extends BankAccount {
  double _overdraftLimit;

  CurrentAccount(String accountNumber, String accountHolder, double balance, this._overdraftLimit)
      : super(accountNumber, accountHolder, balance);

  double get overdraftLimit => _overdraftLimit;

  @override
  void deposit(double amount) {
    if (amount > 0) {
      updateBalance(amount);
      print('Deposited: \$${amount.toStringAsFixed(2)}');
    } else {
      print('Invalid deposit amount');
    }
  }

  @override
  void withdraw(double amount) {
    if (amount > 0 && (balance - amount) >= -_overdraftLimit) {
      updateBalance(-amount);
      print('Withdrawn: \$${amount.toStringAsFixed(2)}');
    } else {
      print('Invalid withdrawal amount or exceeds overdraft limit');
    }
  }

  @override
  void displayAccountInfo() {
    print('''
Current Account Information:
  Account Number: $accountNumber
  Account Holder: $accountHolder
  Balance: \$${balance.toStringAsFixed(2)}
  Overdraft Limit: \$${_overdraftLimit.toStringAsFixed(2)}
''');
  }
}
```

SavingsAccount Class



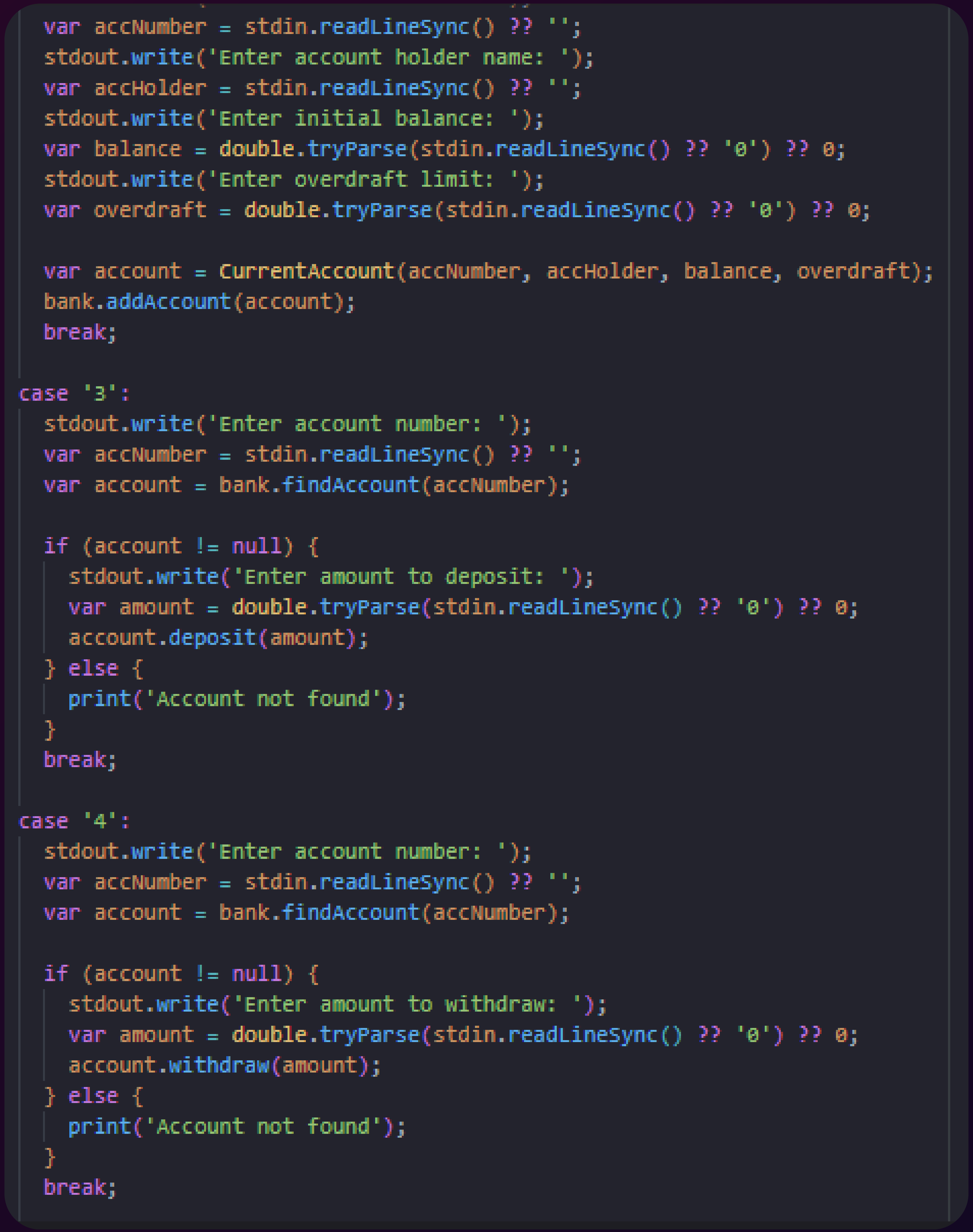
```
class SavingsAccount extends BankAccount {
  double _interestRate;

  SavingsAccount(String accountNumber, String accountHolder, double balance, this._interestRate)
      : super(accountNumber, accountHolder, balance);

  double get interestRate => _interestRate;

  @override
  void deposit(double amount) {
    if (amount > 0) {
      updateBalance(amount);
      print('Deposited: \$${amount.toStringAsFixed(2)}');
    } else {
      print('Invalid deposit amount');
    }
  }

  @override
  void withdraw(double amount) {
    if (amount > 0 && amount <= balance) {
      updateBalance(-amount);
      print('Withdrawn: \$${amount.toStringAsFixed(2)}');
    } else {
      print('Invalid withdrawal amount or insufficient funds');
    }
  }

  void applyInterest() {
    double interest = balance * _interestRate / 100;
    updateBalance(interest);
    print('Interest applied: \$${interest.toStringAsFixed(2)}');
  }

  @override
  void displayAccountInfo() {
    print('''
Savings Account Information:
  Account Number: $accountNumber
```

Main



```dart
void main() {

  print('=== Bank Management System ===');

  while (running) {
    print('\nMenu:');
    print('1. Create Savings Account');
    print('2. Create Current Account');
    print('3. Deposit');
    print('4. Withdraw');
    print('5. Display Account Info');
    print('6. Display All Accounts');
    print('7. Apply Interest (Savings Account)');
    print('8. Exit');
    stdout.write('Enter your choice: ');

    var choice = stdin.readLineSync();

    switch (choice) {
      case '1':
        stdout.write('Enter account number: ');
        var accNumber = stdin.readLineSync() ?? '';
        stdout.write('Enter account holder name: ');
        var accHolder = stdin.readLineSync() ?? '';
        stdout.write('Enter initial balance: ');
        var balance = double.tryParse(stdin.readLineSync() ?? '0') ?? 0;
        stdout.write('Enter interest rate (%): ');
        var interestRate = double.tryParse(stdin.readLineSync() ?? '0') ?? 0;

        var account = SavingsAccount(accNumber, accHolder, balance, interestRate);
        bank.addAccount(account);
        break;

      case '2':
        stdout.write('Enter account number: ');
        var accNumber = stdin.readLineSync() ?? '';
        stdout.write('Enter account holder name: ');
        var accHolder = stdin.readLineSync() ?? '';
        stdout.write('Enter initial balance: ');
        var balance = double.tryParse(stdin.readLineSync() ?? '0') ?? 0;
        stdout.write('Enter overdraft limit: ');
```

Main

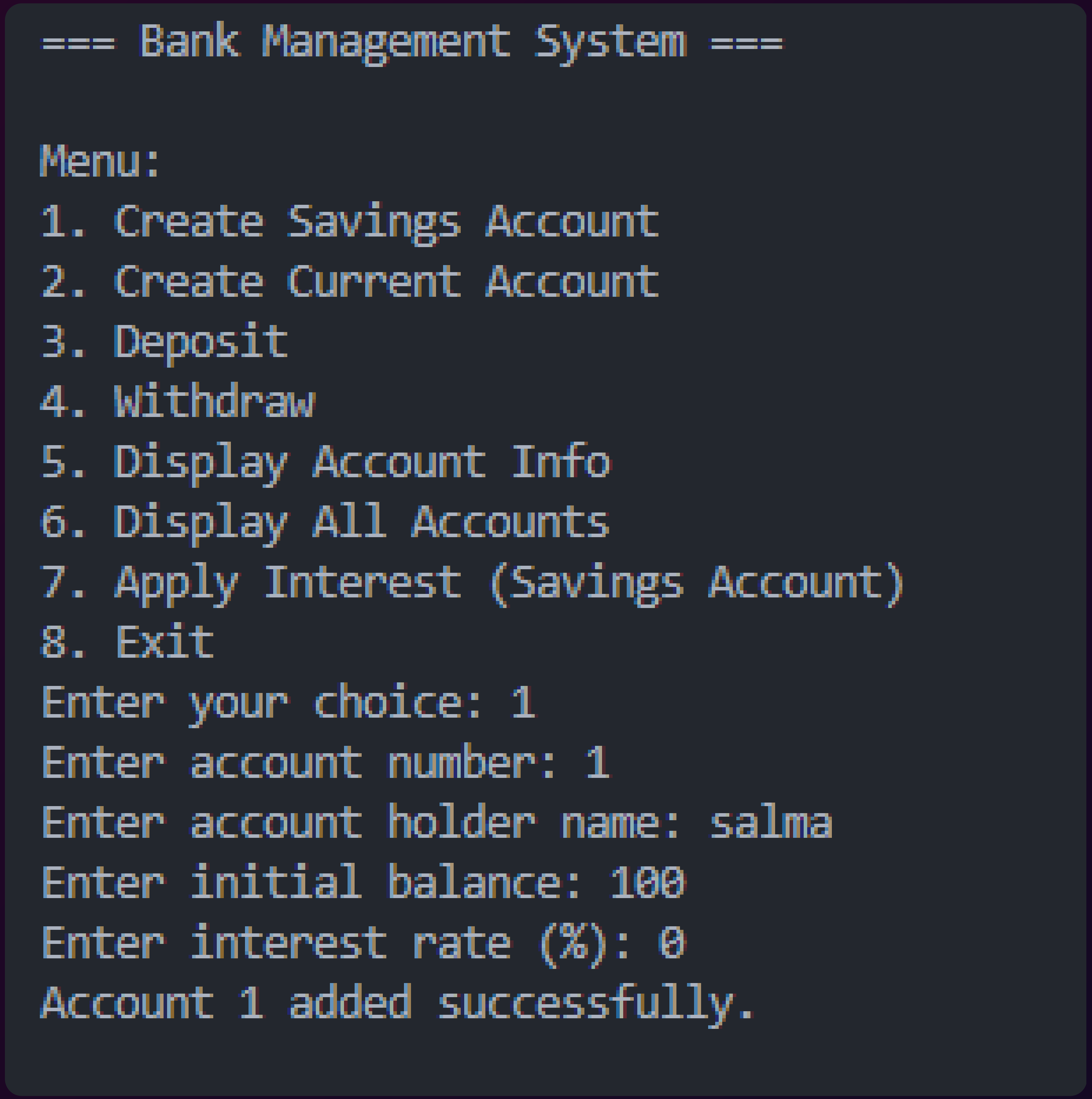


```
  var accNumber = stdin.readLineSync() ?? '';
  stdout.write('Enter account holder name: ');
  var accHolder = stdin.readLineSync() ?? '';
  stdout.write('Enter initial balance: ');
  var balance = double.tryParse(stdin.readLineSync() ?? '0') ?? 0;
  stdout.write('Enter overdraft limit: ');
  var overdraft = double.tryParse(stdin.readLineSync() ?? '0') ?? 0;

  var account = CurrentAccount(accNumber, accHolder, balance, overdraft);
  bank.addAccount(account);
  break;

case '3':
  stdout.write('Enter account number: ');
  var accNumber = stdin.readLineSync() ?? '';
  var account = bank.findAccount(accNumber);

  if (account != null) {
    stdout.write('Enter amount to deposit: ');
    var amount = double.tryParse(stdin.readLineSync() ?? '0') ?? 0;
    account.deposit(amount);
  } else {
    print('Account not found');
  }
  break;

case '4':
  stdout.write('Enter account number: ');
  var accNumber = stdin.readLineSync() ?? '';
  var account = bank.findAccount(accNumber);

  if (account != null) {
    stdout.write('Enter amount to withdraw: ');
    var amount = double.tryParse(stdin.readLineSync() ?? '0') ?? 0;
    account.withdraw(amount);
  } else {
    print('Account not found');
  }
  break;
```

Main

```dart
  case '5':
    stdout.write('Enter account number: ');
    var accNumber = stdin.readLineSync() ?? '';
    var account = bank.findAccount(accNumber);

    if (account != null) {
      account.displayAccountInfo();
    } else {
      print('Account not found');
    }
    break;

  case '6':
    bank.displayAllAccounts();
    break;

  case '7':
    stdout.write('Enter account number: ');
    var accNumber = stdin.readLineSync() ?? '';
    var account = bank.findAccount(accNumber);

    if (account is SavingsAccount) {
      account.applyInterest();
    } else if (account != null) {
      print('This is not a savings account.');
    } else {
      print('Account not found');
    }
    break;

  case '8':
    running = false;
    print('Exiting Bank Management System. Goodbye!');
    break;

  default:
    print('Invalid choice. Please try again.');
}
```

Output



```
=== Bank Management System ===

Menu:
1. Create Savings Account
2. Create Current Account
3. Deposit
4. Withdraw
5. Display Account Info
6. Display All Accounts
7. Apply Interest (Savings Account)
8. Exit
Enter your choice: 1
Enter account number: 1
Enter account holder name: salma
Enter initial balance: 100
Enter interest rate (%): 0
Account 1 added successfully.
```

Thank You