Java Inheritance – Account

Purpose

Practice coding parent and child classes that illustrate polymorphism.

NoSufficientFundsException.java

Write a NoSufficientFundsException class which is a derived class from Exception. The class only has two constructors. The parameterized/initialization constructor will pass the string parameter to the parent class.

NoSufficientFundsException
+NoSufficientFundsException () +NoSufficientFundsException (string: String)

IllegalAmountException.java

Write an IllegalAmountException class which is a derived class from Exception. The class only has two constructors. The parameterized/initialization constructor will pass the string parameter to the parent class.

IllegalAmountException
+IllegalAmountException () +IllegalAmountException (string: String)

Account.java

Write a parent/base abstract class called Account. An abstract class cannot be instantiated and has zero or more abstract methods. The Account class below has 1 abstract method named withdraw. You must designate both the class and method with the non-access modifier *abstract*. The withdraw method will have no curly braces or code and ends in a semi-colon. The first account created will have an id of 10_000_000_000. All subsequent accounts will have an id of one more. The deposit method will throw an IllegalAmountException if a user attempts a negative deposit.

Account
-id: long -balance: double = 0.
+Account() +Account(balance: double) +getBalance(): double +setBalance(balance: double): void +deposit(amt: double) throws IllegalAmountException: void *+withdraw(amt: double) throws NoSufficientFundsException, IllegalAmountException: void* +toString() : String

The class name and method withdraw are in italic which indicate that they are abstract

CheckingAccount.java

Write a CheckingAccount class which is a child or derived class from Account. Implement the withdraw method. A checking account must have a balance greater than zero and the withdraw method should throw an InsufficientFundsException if the account cannot process the withdraw. The withdraw method should throw an IllegalAmountException if a user attempts to process a negative withdraw. Call the parameterized constructor passing a String with the reason and the withdraw amount. This class has no instance variables.

CheckingAccount
+CheckingAccount () +CheckingAccount (balance: double) +withdraw(amt: double) throws NoSufficientFundsException, IllegalAmountException: void +toString() : String

SavingsAccount.java

Write a SavingsAccount class that is a child of Account. A savings account must have a balance of 500(default). The addInterest method will update the account balance based on the interest rate (rate in the form of 0.0625 for 6.25%). This class has 1 instance variable.

SavingsAccount
-interestRate: double = 0.01
+SavingsAccount() +SavingsAccountt(id: long, balance: double, interestRate: double) +setInterestRate(interestRate: double): void +getInterestRate(): double +addInterest(): void +withdraw(amt: double) throws NoSufficientFundsException, IllegalAmountException: void +toString() : String

Sample Execution

```
---------------------------------------------------------------------------------------------------------------------------------
Testing No-Arg Constructors and toString()

Parent: Account{id=10,000,000,000; balance=$0.00}, Child: Checking Account
Parent: Account{id=10,000,000,001; balance=$0.00}, Child: Checking Account
Parent: Account{id=10,000,000,002; balance=$500.00}, Child: SavingsAccount{interestRate=01.00%}
Parent: Account{id=10,000,000,003; balance=$500.00}, Child: SavingsAccount{interestRate=01.00%}
```

```
---------------------------------------------------------------------------------------------------------------------------------------
Making deposits of 300, 400, 1000, 2000 respectively

Parent: Account{id=10,000,000,000; balance=$300.00}, Child: Checking Account
Parent: Account{id=10,000,000,001; balance=$400.00}, Child: Checking Account
Parent: Account{id=10,000,000,002; balance=$1,500.00}, Child: SavingsAccount{interestRate=01.00%}
Parent: Account{id=10,000,000,003; balance=$2,500.00}, Child: SavingsAccount{interestRate=01.00%}
---------------------------------------------------------------------------------------------------------------------------------------
Making withdraws of 400(can't), 400, 500, 2500.01(can't)

Insufficient funds
Balance: $300.00
Withdraw amount: $400.00

Insufficient funds. Account must be >= $500
Balance: $2500.00
Withdraw amount: $2500.01

Parent: Account{id=10,000,000,000; balance=$300.00}, Child: Checking Account
Parent: Account{id=10,000,000,001; balance=$0.00}, Child: Checking Account
Parent: Account{id=10,000,000,002; balance=$1,000.00}, Child: SavingsAccount{interestRate=01.00%}
Parent: Account{id=10,000,000,003; balance=$2,500.00}, Child: SavingsAccount{interestRate=01.00%}
---------------------------------------------------------------------------------------------------------------------------------------
Making withdraws of negatives

Amount must be positive: $-0.01
Amount must be positive: $-5.01
Amount must be positive: $-10.01
Amount must be positive: $-15.01

Making a negative deposit

Amount must be positive: $-10.00
---------------------------------------------------------------------------------------------------------------------------------------
Setting Interest w/ 10% and 6.25% rates

Parent: Account{id=10,000,000,002; balance=$1,000.00}, Child: SavingsAccount{interestRate=10.00%}
Parent: Account{id=10,000,000,003; balance=$2,500.00}, Child: SavingsAccount{interestRate=06.25%}

Add Interest(Expecting $1,100.00 and $2,656.25)

Parent: Account{id=10,000,000,002; balance=$1,100.00}, Child: SavingsAccount{interestRate=10.00%}
Parent: Account{id=10,000,000,003; balance=$2,656.25}, Child: SavingsAccount{interestRate=06.25%}
```