Exercise 13.2 – Bank Application AOP

The Setup:

In this exercise, we will be extending the bank application to use AOP. Create a copy of
exercise12.2 and call it **exercise13.2**. Be sure to update the pom.xml file dependencies for AOP.

The Exercise:

Use AOP to:
   a) Log every call to any method in the bank.dao package (using the Logger).
   b) Use the Spring StopWatch functionality to measure the duration of all service level
      methods (any method in the bank.service package) and output the results to the console.
   c) Log every JMS message that is sent (using the Logger)
   d) Remove all the other logger calls from AccountService (doing so will make it easier to
      see whether your advice is running or not.

bank.service

<<interface>>
IAccountService
+createAccount()
+deposit()
+withdraw()
+depostEuros()
+withdrawEuros()
+getAccount()
+getAllAccounts()
+transferFunds()

AccountService
+createAccount()
+deposit()
+withdraw()
+depostEuros()
+withdrawEuros()
+getAccount()
+getAllAccounts()
+transferFunds()

bank

Application
+main()

DI

<<interface>>
ICurrencyConverter
+euroToDollars()
+dollarsToEuros()

CurrencyConverter
+euroToDollars()
+dollarsToEuros()

bank.domain

<<entity>>
Account
+accountnumber
+deposit()
+withdraw()
+getBalance()
+transferFunds()

<<entity>>
Customer
+name
1

<<entity>>
AccountEntry
+date
+amount
+description
+fromAccountNumber
+fromPersonName

DI

bank.dao

<<interface>>
IAccountDAO
+saveAccount()
+updateAccount()
+loadAccount()
+getAccounts()

AccountDAO
+saveAccount()
+updateAccount()
+loadAccount()
+getAccounts()

AOP

AOP

bank.service.aop

StopWatchAdvice
+methodTimer()

JMSLogAdvice
+logJMSMessage()

DAOLogAdvice
+logDAOCall()

bank.jms

<<interface>>
IJMSSender
+sendJMSMessage()

JMSSender
+sendJMSMessage()

AOP

DI

bank.logging

<<interface>>
ILogger
+log()

Logger
+log()