

Spring AOP Assignments

- 1) Write a program to demonstrate Proxy design pattern.
- 2) Example on Before advice, After advice, throws advice, After returning advice and around advice, loggers and @Transactional

Design and Develop a Banking Application as follows:

- a. Create a BankAccount class with following attributes: accountId, accountHolderName, accountType, accountBalance
- b. Create an interface BankAccountRepository with following methods:
public double getBalance(long accountId)
public double updateBalance(long accountId, double newBalance):
Note: Above method returns updated balance.
- c. Create a class BankAccountRepositoryImpl that implements BankAccountRepository interface.
You can use database or any collection object as persistence store.
- d. Create an interface BankAccountService with following methods:
public double withdraw(long accountId, double balance) throws LowBalanceException
public double deposit(long accountId, double balance)
public double getBalance(long accountId)
public boolean fundTransfer(long fromAccount, long toAccount, double amount)
throws BankAccountNotFoundException, LowBalanceException
- e. Create a class BankAccountServiceImpl that implements BankAccountService interface.
- f. Create a class BankAccount controller with following operations:
public double withdraw(long accountId, double balance)
public double deposit(long accountId, double balance)
public double getBalance(long accountId)
public boolean fundTransfer(long fromAccount, long toAccount, double amount)
- g. Create a Test class with main() method, get BankAccountController bean object from ApplicationContext and perform all the operations.

h. Also write the JUnit Test cases for above program.

- Apply the Logger aspect for above application using Spring AOP.
- Use throws advice when either LowBalanceException or BankAccountNotFoundException occurs during Transaction.
(Using Logger aspect, log the exceptions)
- Use appropriate transaction manager for Transactions like withdraw(), deposit() and fundTransfer()
- Use both XML and annotations based configurations.
- Use JoinPoints and PointCuts.