|  |  |
| --- | --- |
| Course | Advanced Software Design – CS525 |
| Assignment | Lab 9 |
| Week | 09 |
| Due | Apr 13, 2020 |
| Student name | Quan Hong Doan |
| Student ID | 986956 |

Online version:

**Problem**:

Problem is the question of bank code from the PDF file.

**Answer**:

The source code on my GitHub

**Factory**

**DaoFactory**

**public** **class** DaoFactory {

**private** BankFactory factory;

**public** DaoFactory() {

**try** (InputStream in = **new** FileInputStream(**new** File("").getAbsolutePath() + "/config.properties")) {

Properties prop = **new** Properties();

prop.load(in);

String env = prop.getProperty("db.password");

**if** (env.equals("test")) {

factory = **new** MockFactory();

} **else** **if** (env.equals("production")) {

factory = **new** ProductionFactory();

} **else** {

System.***out***.println("Check the environment config");

}

} **catch** (IOException ex) {

ex.printStackTrace();

}

}

**public** BankFactory getFactory() {

**return** factory;

}

}

**BankFactory**

**public** **interface** BankFactory {

IAccountDAO getAccountDAO();

EmailDAO getEmailDAO();

}

**ProductionFactory**

**public** **class** ProductionFactory **implements** BankFactory {

@Override

**public** IAccountDAO getAccountDAO() {

**return** **new** AccountDAOLogger();

}

@Override

**public** EmailDAO getEmailDAO() {

**return** EmailSender.*getInstance*();

}

}

**MockFactory**

**public** **class** MockFactory **implements** BankFactory {

@Override

**public** IAccountDAO getAccountDAO() {

**return** **new** MockAccountDAO();

}

@Override

**public** EmailDAO getEmailDAO() {

**return** **new** MockEmailSender();

}

}

**EmailDAO**

**public** **interface** EmailDAO {

**void** sendEmail();

}

**MockAccountDAO**

**public** **class** MockAccountDAO **implements** IAccountDAO {

@Override

**public** **void** saveAccount(Account account) {

System.***out***.println("MockAccountDAO: saveAccount");

}

@Override

**public** **void** updateAccount(Account account) {

System.***out***.println("MockAccountDAO: updateAccount");

}

@Override

**public** Account loadAccount(**long** accountnumber) {

System.***out***.println("MockAccountDAO: loadAccount");

**return** **null**;

}

@Override

**public** Collection<Account> getAccounts() {

System.***out***.println("MockAccountDAO: getAccounts");

**return** **null**;

}

}

**MockEmailSender**

**public** **class** MockEmailSender **implements** EmailDAO {

@Override

**public** **void** sendEmail() {

System.***out***.println("MockEmailSender");

}

}

**Singleton**

**public** **class** EmailSender **implements** EmailDAO {

**private** **static** EmailDAO *sender*;

**private** EmailSender() {

**if** (*sender* != **null**) {

**throw** **new** RuntimeException("Already initiated, use getInstance() to get the instance instead.");

}

}

**public** **static** EmailDAO getInstance() {

**if** (*sender* == **null**) {

**synchronized** (EmailSender.**class**) {

**if** (*sender* == **null**) {

*sender* = **new** EmailSender();

}

}

}

**return** *sender*;

}

@Override

**public** **void** sendEmail() {

System.***out***.println("EmailSender");

}

}

**Builder**

**public** AccountEntry(Builder builder) {

**this**.date = builder.date;

**this**.amount = builder.amount;

**this**.description = builder.description;

**this**.fromAccountNumber = builder.fromAccountNumber;

**this**.fromPersonName = builder.fromPersonName;

}

**public** **static** **class** Builder {

Date date;

**double** amount;

String description;

String fromAccountNumber;

String fromPersonName;

**public** Builder withDate(Date d) {

**this**.date = d;

**return** **this**;

}

**public** Builder withAmount(**double** a) {

**this**.amount = a;

**return** **this**;

}

**public** Builder withDescription(String desc) {

**this**.description = desc;

**return** **this**;

}

**public** Builder withFromAccount(String fromAccount) {

**this**.fromAccountNumber = fromAccount;

**return** **this**;

}

**public** Builder withFromName(String fromName) {

**this**.fromPersonName = fromName;

**return** **this**;

}

**public** AccountEntry build() {

**return** **new** AccountEntry(**this**);

}

}

**Decorator**

**public** **class** AccountDAOLogger **extends** AccountDAO {

**public** AccountDAOLogger() {

**super**();

}

**public** **void** saveAccount(Account account) {

System.***out***.println("AccountDAOLogger: saveAccount");

**super**.saveAccount(account);

}

**public** **void** updateAccount(Account account) {

System.***out***.println("AccountDAOLogger: updateAccount");

**super**.updateAccount(account);

}

**public** Account loadAccount(**long** accountnumber) {

System.***out***.println("AccountDAOLogger: loadAccount");

**return** **super**.loadAccount(accountnumber);

}

**public** Collection<Account> getAccounts() {

System.***out***.println("AccountDAOLogger: getAccounts");

**return** **super**.getAccounts();

}

}

And we need to change in the AccountService class

**public** AccountService() {

accountDAO = **new** AccountDAOLogger();

}