**package** com.springpackage;

**public** **class** BankAccount {

**private** **long** ID;

**private** **double** balance;

**private** String name,type;

**public** BankAccount(**long** iD, **double** balance, String name, String type) {

**super**();

**this**.ID = iD;

**this**.balance = balance;

**this**.name = name;

**this**.type = type;

}

**public** **long** getID() {

**return** ID;

}

**public** **void** setID(**long** iD) {

ID = iD;

}

**public** **double** getBalance() {

**return** balance;

}

**public** **void** setBalance(**double** balance) {

**this**.balance = balance;

}

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

**public** String getType() {

**return** type;

}

**public** **void** setType(String type) {

**this**.type = type;

}

**public** BankAccount() {

**super**();

}

}**package** com.springpackage;

**public** **interface** BankAccountRepositary {

**public** **double** getBalance(**long** Id);

**public** **double** updateBalance(**long** Id, **double** newbalance);

}

**package** com.springpackage;

**import** java.util.ArrayList;

**import** java.util.List;

**import** javax.swing.text.html.HTMLDocument.Iterator;

**public** **class** BankAccountRepositaryImpl **implements** BankAccountRepositary {

List<BankAccount> idList=**null**;

**public** BankAccountRepositaryImpl() {

**super**();

idList=**new** ArrayList<BankAccount>();

idList.add(**new** BankAccount(1l,20000.00,"Tom","savings"));

}

@Override

**public** **double** getBalance(**long** Id) {

java.util.Iterator<BankAccount> itr=idList.iterator();

**double** balance=0;

**while**(itr.hasNext()){

BankAccount ba=(BankAccount)itr.next();

**if**(Id==ba.getID()) {

balance=ba.getBalance();

**break**;

}

}

**return** balance;

}

@Override

**public** **double** updateBalance(**long** Id, **double** newbalance) {

java.util.Iterator<BankAccount> itr=idList.iterator();

**double** balance=0;

**while**(itr.hasNext()){

BankAccount ba=(BankAccount)itr.next();

**if**(Id==ba.getID()) {

balance=ba.getBalance()+newbalance;

**break**;

}

}

**return** balance;

}

}

**package** com.springpackage;

**public** **interface** BankAccountService {

**public** **double** withdraw(**long** Id , **double** balance);

**public** **double** deposit(**long** Id , **double** balance);

**public** **double** getbalance(**long** Id);

**public** **boolean** fundTransfer(**long** fromAccount, **long** toAccount, **double** amount);

}

**package** com.springpackage;

**import** java.util.ArrayList;

**import** java.util.List;

**public** **class** BankAccountServiceImpl **implements** BankAccountService {

List<BankAccount> idList=**null**;

**public** BankAccountServiceImpl() {

**super**();

idList=**new** ArrayList<BankAccount>();

idList.add(**new** BankAccount(1l,20000.00,"Tom","savings"));

}

@Override

**public** **double** withdraw(**long** Id, **double** balance) {

java.util.Iterator<BankAccount> itr=idList.iterator();

**double** balance1=0;

**while**(itr.hasNext()){

BankAccount ba=(BankAccount)itr.next();

**if**(Id==ba.getID()) {

balance1=ba.getBalance()-balance;

**break**;

}

}

**return** balance1;

}

@Override

**public** **double** deposit(**long** Id, **double** balance) {

java.util.Iterator<BankAccount> itr=idList.iterator();

**double** balance1=0;

**while**(itr.hasNext()){

BankAccount ba=(BankAccount)itr.next();

**if**(Id==ba.getID()) {

balance1=ba.getBalance()+balance;

**break**;

}

}

**return** balance1;

}

@Override

**public** **double** getbalance(**long** Id) {

java.util.Iterator<BankAccount> itr=idList.iterator();

**double** balance1=0;

**while**(itr.hasNext()){

BankAccount ba=(BankAccount)itr.next();

**if**(Id==ba.getID()) {

balance1=ba.getBalance();

**break**;

}

}

**return** balance1;

}

@Override

**public** **boolean** fundTransfer(**long** fromAccount, **long** toAccount, **double** amount) {

java.util.Iterator<BankAccount> itr=idList.iterator();

**double** balance1=0;

**while**(itr.hasNext()){

BankAccount ba=(BankAccount)itr.next();

**if**(fromAccount==ba.getID()) {

balance1=ba.getBalance();

**break**;

}

}

**return** **false**;

}

}

**package** com.springpackage;

**public** **class** BankAccountController {

BankAccountServiceImpl bankAccount;

**public** **double** withdraw(**long** Id , **double** balance) {

**return** **this**.bankAccount.withdraw(Id,balance);

}

**public** **double** deposit(**long** Id , **double** balance) {

**return** **this**.bankAccount.deposit(Id,balance);

}

**public** **double** getbalance(**long** Id) {

**return** **this**.bankAccount.getbalance(Id);

}

**public** **boolean** fundTransfer(**long** fromAccount, **long** toAccount, **double** amount) {

**return** **false**;

}

}

**package** com.springpackage;

**import** org.springframework.context.ApplicationContext;

**import** org.springframework.context.support.ClassPathXmlApplicationContext;

**public** **class** Test {

**public** **static** **void** main(String[] args) {

ApplicationContext context = **new** ClassPathXmlApplicationContext("Beans.xml");

BankAccountController obj = (BankAccountController) context.getBean("bankAccountController");

obj.withdraw(1L,2000.0);

}

}

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<beans xmlns=*"http://www.springframework.org/schema/beans"*

xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xsi:schemaLocation=*"*

*http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans.xsd"*>

<bean id=*"helloworld"* class=*"com.springpackage.Helloworld"*>

</bean >

<bean id=*"bankAccountController"* class=*"com.springpackage.BankAccountController"*>

</bean>

<bean >

</bean>

</beans>