**Observer Pattern Example: Archiver**

**Interface: Subject**

**package** archiver;

**public** **interface** Subject {

**public** **void** registerObserver(Observer o);

**public** **void** removeObserver(Observer o);

**public** **void** notifyObserver();

}

**Class: Database**

**package** archiver;

**import** java.util.\*;

**public** **class** Database **implements** Subject{

**private** Vector observers;

**private** String operation;

**private** String record;

**public** Database() {

observers = **new** Vector();

}

@Override

**public** **void** registerObserver(Observer o) {

observers.add(o);

}

@Override

**public** **void** removeObserver(Observer o) {

observers.remove(o);

}

**public** **void** editRecord(String operation, String record) {

**this**.operation = operation;

**this**.record = record;

notifyObserver();

}

**public** **void** notifyObserver() {

**for** (**int** loopIndex = 0; loopIndex < observers.size(); loopIndex++) {

Observer observer = (Observer)observers.get(loopIndex);

observer.update(operation, record);

}

}

}

**Interface: Observer**

**package** archiver;

**public** **interface** Observer {

**public** **void** update(String operation, String record);

}

**Class: Archiver**

**package** archiver;

**public** **class** Archiver **implements** Observer{

**public** Archiver() {

}

**public** **void** update(String operation, String record) {

System.***out***.println("This archiver says a " + operation +

" operation was performed on " + record);

}

}

**Class: Client**

**package** archiver;

**public** **class** Client **implements** Observer {

**public** Client() {

}

**public** **void** update(String operation, String record) {

System.***out***.println("This client says a " + operation

+ " operation was performed on " + record);

}

}

**Class: Boss**

**package** archiver;

**public** **class** Boss **implements** Observer {

**public** Boss() {

}

**public** **void** update(String operation, String record) {

System.***out***.println("This Boss says a " + operation +

" operation was performed on " + record);

}

}

**Class: TestObserver**

**package** archiver;

**public** **class** TestObserver {

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

Database database = **new** Database();

Archiver archiver = **new** Archiver();

Client client = **new** Client();

Boss boss = **new** Boss();

database.registerObserver(archiver);

database.registerObserver(client);

database.registerObserver(boss);

database.editRecord("delete", "record 1");

}

}

Result:

This archiver says a delete operation was performed on record 1

This client says a delete operation was performed on record 1

This Boss says a delete operation was performed on record 1