# What are good real life Java inheritance (or use of abstract classes) examples ?

Posted on June 9, 2015 by Arulkumaran Kumaraswamipillai

We have already looked at Why favor composition over inheritance? with reasoning and code examples. Inheritance still has its use and let's look at some real life examples.

## #1. Unit Test Classes

```
import javax.inject.Inject;

import org.junit.Assert;
import org.junit.Test;

public class EmailAddressDaoTest extends BaseTes

```

## 600+ Full Stack Java/JEE Interview Q&As ♥Free ♦FAQs

```
    @Inject
    private EmailAddressDao emailAddressDao;

    @Test
    public void testGetEmailAddressForUserId() {
        String emailAddress = emailAddressDao.ge
        Assert.assertEquals("hleibowitz@stateone
    }
}
```

In the base class you can wire up Spring configs, shared logic, etc. For example, you may want to load SQL scripts to test your DAO layer with a HSQLDB embedded database. So, the scripts will get loaded only once into the in memory database and then multiple unit tests can be executed. This is demonstrated in JPA tutorial with unit testing.

```
import static org.junit.Assert.fail;

import java.lang.reflect.Field;
import java.lang.reflect.Modifier;

import org.junit.runner.RunWith;
import org.springframework.test.context.ContextC
import org.springframework.test.context.junit4.S
import org.springframework.test.context.transact
import org.springframework.transaction.annotatio

/**
 * Base class for all unit tests used in this mo
 */
@Transactional(readOnly = false)
@TransactionConfiguration(defaultRollback = fals
@ContextConfiguration(classes = {
    DomainTestConfiguration.class, MyAppServiceT
    initializers = MyAppApplicationContextTestIn
)
@RunWith(SpringJUnit4ClassRunner.class)
public abstract class BaseTest {

     //any common logic shared all test cases go
}
```

## #2. Hibernate Entity Base classes wiring IDs, Version, and Auditing Fields

```
@MappedSuperclass
public abstract class AbstractDomainEntity<ID ex
    private static final long serialVersionUID =

    //for pessimistic concurrency control
    @Version
    @Column(name = "lock_version", nullable = fa
    private Long version;
```

## As a Java Architect

Java architecture & design concepts

```

    @Override
    public abstract ID getId();

    //setters, getts, equals/hashcode/toString m
}
```

Mapping for the database audit fields

```
@MappedSuperclass
@EntityListeners(DomainEntityListener.class)
public abstract class AbstractAuditableDomainEnt
    extends AbstractDomainEntity<ID> {

    private static final long serialVersionUID =

    @Column(name = "created_by", nullable = true
    private String createdBy;

    @Column(name = "created_date", nullable = fa
    @Type(type = "org.joda.time.contrib.hibernat
    private DateTime createdDate;

    @Column(name = "updated_by", nullable = true
    private String lastModifiedBy;

    @Column(name = "updated_date", nullable = fa
    @Type(type = "org.joda.time.contrib.hibernat
    private DateTime lastModifiedDate;

    //setters, getts, equals/hashcode/toString m
}
```

The other entity classes can extend AbstractAuditableDomainEntity class.

```
@Table(name = "account_tbl")
@Entity(name = "Account")
public class Account extends AbstractAuditableDom

  //...
}
```

## #3. Template method and Composite design Patterns

are good examples of using an abstract base class and inheritance. Template and Composite design patterns are covered in Java abstract classes Vs interfaces. **Q11** discusses the template method design pattern — 12 java design patterns interview questions and answers

interview Q&As with diagrams | What should be a typical Java EE architecture?

## Senior Java developers must have a good handle on

open all | close all
- ⊞ Best Practice (6)
- ⊞ Coding (26)
- ⊞ Concurrency (6)
- ⊞ Design Concepts (7)
- ⊞ Design Patterns (11)
- ⊞ Exception Handling (3
- ⊞ Java Debugging (21)
- ⊞ Judging Experience I
- ⊞ Low Latency (7)
- ⊞ Memory Managemen
- ⊞ Performance (13)
- ⊞ QoS (8)
- ⊞ Scalability (4)
- ⊞ SDLC (6)
- ⊞ Security (13)
- ⊞ Transaction Managen

## 80+ step by step Java Tutorials

open all | close all
- ⊞ Setting up Tutorial (6)
- ⊞ Tutorial - Diagnosis (2
- ⊞ Akka Tutorial (9)
- ⊞ Core Java Tutorials (2
- ⊞ Hadoop & Spark Tuto

In Java API, AbstractStringBuilder, AbstractQueue, AbstractAction, AbstractButton, AbstractSet, AbstractMap, etc provide a skeletal implementationto minimize the effort required to implement the concrete classes.

## In Java 8, you can define default methods in an interface. Would this make abstract classes not required any more?

With the advent of default methods in Java 8 functional interfaces, the question arises, which one (i.e. abstract classess or interfaces with default methods?) to use in a given situation. Abstract class or Interfaces with default methods. The abstract classes still have their place as

**1)** Default methods are never final.
**2)** Default methods can not be synchronized.
**3)** default methods can not override Object class's methods.
**4)** Interfaces where default methods are defined cannot hold state. That is, you can't define variables in an interface. Only behvior is allowed in default methods.

## Popular Posts

♦ 11 Spring boot interview questions & answers
**861 views**

♦ Q11-Q23: Top 50+ Core on Java OOP Interview Questions & Answers
**829 views**

18 Java scenarios based interview Questions and Answers
**448 views**

001A: ♦ 7+ Java integration styles & patterns interview questions & answers
**407 views**

♦ 7 Java debugging interview questions & answers
**311 views**

♦ 10 ERD (Entity-Relationship Diagrams) Interview Questions and Answers
**303 views**

⊞ JEE Tutorials (19)
⊞ Scala Tutorials (1)
⊞ Spring & HIbernate Tu
⊞ Tools Tutorials (19)
⊞ Other Tutorials (45)

## Preparing for Java written & coding tests

open all | close all
⊞ ♦ Complete the given
⊞ Can you write code? (
⊞ Converting from A to I
⊞ Designing your classe
⊞ Java Data Structures
⊞ Passing the unit tests
⊞ What is wrong with th
⊞ Writing Code Home A
⊞ Written Test Core Jav
⊞ Written Test JEE (1)

## How good are your…to go places?

open all | close all
⊞ Career Making Know-
⊞ Job Hunting & Resum

01b: ♦ 13 Spring basics Q8 – Q13 interview questions & answers

**294 views**

01: ♦ 15 Ice breaker questions asked 90% of the time in Java job interviews with hints

**288 views**

♦ Q24-Q36: Top 50+ Core on Java classes, interfaces and generics interview questions & answers

**263 views**

8 Git Source control system interview questions & answers

**215 views**

Bio | **Latest Posts**

**Arulkumaran Kumaraswamipillai**

Mechanical Eng to freelance Java developer in 3 yrs. Contracting since 2003, and attended 150+ Java job interviews, and often got 4 - 7 job offers to choose from. It pays to prepare. So, published Java interview Q&A books via Amazon.com in 2005, and sold 35,000+ copies. Books are outdated and replaced with this subscription based site.

**About** Arulkumaran Kumaraswamipillai

Mechanical Eng to freelance Java developer in 3 yrs. Contracting since 2003, and attended 150+ Java job interviews, and often got 4 - 7 job offers to choose from. It pays to prepare. So, published Java interview Q&A books via Amazon.com in 2005, and sold 35,000+ copies. Books are outdated and replaced with this subscription based site.

‹ ♥ Top 6 Spring wiring via JavaConfig [i.e. @Configuration ] examples

♥ Understanding Git terms origin, master, and head ›

**Posted in** member-paid, OOP

**Tags:** TopX

## Empowers you to open more doors, and fast-track

### Technical Know Hows

☀ Java generics in no time ☀ Top 6 tips to transforming your thinking from OOP to FP ☀ How does a HashMap internally work? What is a hashing function?
☀ 10+ Java String class interview Q&As ☀ Java auto un/boxing benefits & caveats ☀ Top 11 slacknesses that can come back and bite you as an experienced Java developer or architect

### Non-Technical Know Hows

☀ 6 Aspects that can motivate you to fast-track your career & go places ☀ Are you reinventing yourself as a Java developer? ☀ 8 tips to safeguard your Java career against offshoring ☀ My top 5 career mistakes

## Prepare to succeed

☀ Turn readers of your Java CV go from "Blah blah" to "Wow"? ☀ How to prepare for Java job interviews? ☀ 16 Technical Key Areas ☀ How to choose from multiple Java job offers?

Select Category ▼

## © Disclaimer