

Domain Driven Design (DDD) Mock Exams

<http://www.JavaChamp.com>

Open Certification Plattform

Authors: N. Ibrahim, Y. Ibrahim

Copyright (c) 2009-2010

Introducing JavaChamp.com Website

JavaChamp.com is an Open Certification Platform.

What does this mean?

JavaChamp is the best place to learn, share, and certify your professional skills.

We help you develop yourself in the field of computer science and programming

Here are the most significant features offered by JavaChamp:

Online Exams

Start Online Certification Exams in SCJP, SCEA, EJB, JMS, JPA and more...

Top quality mock exams for SCJP, SCEA, EJB, JMS, JPA. Start Express or topic-wise customized exam.

* We offer you unlimited free mock exams

* Exams cover subjects like SCJP, SCEA, EJB, JMS, JPA,..

* You can take as many exams as you want and at any time and for no charges

* Each exam contains 20 multiple choice questions

* You can save the exams taken in your exams history

* Your exams history saves the exams you took, the scores you got, time took you to finish the exam, date of examination and also saves your answers to the questions for later revision

* You can re-take the same exam to monitor your progress

* Your exams history helps the system to offer you variant new questions every time you take a new exam, therefore we encourage you to register and maintain an exams history

Network

Find guidance through the maze, meet Study-Mates, Coaches or Trainees...

Studying together is fun, productive and helps you in building your professional network and collecting leads

Bookshelf

JavaChamp Bookshelf full of PDF eBooks...

Download PDF books with a selected sample of the JavaChamp question bank in SCJP, SCEA, EJB, JMS and more or read it online

JavaChamp Profile

You may publish your profile and connect to your colleagues and friends.

Content Channel

Be an Author and get recognition, leads, and more...

Contributing to the JavaChamp question bank will earn your recognition of your professional skills, expands your network, introduce you to potential leads

Join Us

Join the fast growing JavaChamp Community now.

JavaChamp Community is young and very dynamic, we would be thrilled to welcome you on board :o)



★ About us

JavaChamp.com is a Java Open Certification Plattform!

- It provides unique services to help you develop yourself in the field of computer science and programming
- It provides a web interface for the Java community to learn, interact and certify its acquired java experience.
- JavaChamp.com helps the Java developers to achieve the

★ Why to register in our community?

Registration is optional and free but brings many advantages

- By registering in JavaChamp.com you can save and keep track of your exams for later revision and hence to help you monitor your progress.

CHECK OUR eBook

JavaChamp.com's eBook encompasses a large number of multiple questions, which cover the subjects java, C and assembly. The book doesn't ...

[More](#) →

★ Exam in Session

Question 3 / 20

What is the expected output?

```
01. public class OuterTest {
02.
03.     public static void main(String args[]) {
04.         Airplane.BlackBox box = new Airplane().new BlackBox(); // line 1
05.         box.printVariables();
06.
07.     }
08. }
09.
10. class Airplane {
11.     String code = "11";
12.
13.     class BlackBox {
14.         String code = "22";
15.
16.         public void printVariables() {
17.             System.out.print(code);
18.             System.out.print(Airplane.this.code); // line 20
19.
20.         }
21.     }
22. }
23. }
```

- Compile error because of line 1 (incorrect instantiation)
- Compile error because of line 20 (can't access Airplane's variables)
- 2222
- 1111
- 2211

[Back](#) [Next](#)[Finish and evaluate](#) [Abort Exam](#)

CHECK OUR eBook

JavaChamp.com's eBook encompasses a large number of multiple questions, which cover the subjects java,C and assembly. The book doesn't ...

[More](#) →

Copyright

Copyright 2009 JavaChamp.com

Online version published by JavaChamp.com Germany.

DISCLAIMER

All services and content of JavaChamp.com are provided under JavaChamp.com terms of use on an "as is" basis, without warranty of any kind, either expressed or implied, including, without limitation, warranties that the provided services and content are free of defects, merchantable, fit for a particular purpose or non-infringing. The entire risk as to the quality and performance of the provided services and content is with you. In no event shall JavaChamp.com be liable for any damages whatsoever arising out of or in connection with the use or performance of the services. Should any provided services and content prove defective in any respect, you (not the initial developer, author or any other contributor) assume the cost of any necessary servicing, repair or correction. This disclaimer of warranty constitutes an essential part of these "terms of use". No use of any services and content of JavaChamp.com is authorized hereunder except under this disclaimer.

The detailed "terms of use" of JavaChamp.com can be found under:

<http://www.javachamp.com/public/termsOfUse.xhtml>

This work is licensed under a Creative Commons Attribution-Noncommercial-No Derivative Works 3.0 license.

The full license legal code can be found under:

<http://creativecommons.org/licenses/by-nc-nd/3.0/legalcode>

And a human-readable summary of the this license can be found under:

<http://creativecommons.org/licenses/by-nc-nd/3.0/>

According to the Creative Commons Attribution-Noncommercial-No Derivative Works 3.0 license You agree to the following:

You are free to share, copy, distribute and transmit the work under the following conditions:

- You must attribute the work to JavaChamp.com with a link to <http://www.javachamp.com>.
- You may not use this work for commercial purposes.
- You may not alter, transform, or build upon this work.

Table of Contents

1. Domain Driven Design
 - Basics

1. Chapter: Domain Driven Design

Chapter Description and Objectives

1. Basics

Exam Category Description and Objectives

1.1.1. intersecting domain context maps

Author: Yasser Ibrahim

which design pattern you would use to create domain context maps when the intersection between domain contexts is high?

Please choose all the answers that apply:

- open host services pattern
- anticorruption layers pattern
- separate ways pattern
- shared kernel pattern
- customer-supplier pattern

Check the answer of this question online on JavaChamp.com: [intersecting domain context maps](#)

1.1.2. repository design pattern

Author: Yasser Ibrahim

which design pattern you would use to encapsulate the logic of retrieving the object references from the backing enterprise data store?

Please choose only one answer:

- value object
- aggregate
- entity
- repository

Check the answer of this question online on JavaChamp.com: [repository model pattern](#)

1.1.3. model driven design Application layer

Author: Yasser Ibrahim

What is the role of the Application layer in model driven design paradigm?

Please choose only one answer:

- renders output to the user and gathers user gestures
- implements entities persistence, and layers communication
- encapsulates the core of the business logic
- harmonize and monitors the application activities and its progress

Check the answer of this question online on JavaChamp.com: [model driven design application layer](#)

1.1.4. aggregate domain pattern

Author: Yasser Ibrahim

which design pattern you would use to enforce the domain constraints and ensure data integrity?

Please choose only one answer:

- value object
- aggregate
- entity
- repository

Check the answer of this question online on JavaChamp.com: [aggregate domain pattern](#)

1.1.5. Communicational vs. functional cohesion

Author: Yasser Ibrahim

what is the difference between communicational and functional cohesion?

Please choose all the answers that apply:

- communicational cohesion modularize subsystems that operate on the same data.
- communicational cohesion modularize subsystems that collaborate together on a single task
- functional cohesion modularize subsystems that operate on the same data.
- functional cohesion modularize subsystems that collaborate together on a single task

Check the answer of this question online on JavaChamp.com: [communicational vs. functional cohesion](#)

1.1.6. factory domain pattern

Author: Yasser Ibrahim

which design pattern you would use to encapsulate the process of complex object creation?

Please choose only one answer:

- value object
- factory
- aggregate
- repository
- entity

Check the answer of this question online on JavaChamp.com: [factory domain pattern](#)

1.1.7. what is an aggregate?

Author: Yasser Ibrahim

A bundle of related objects which are treated from the client prescriptive as a single unit exposing a root object.

Please choose only one answer:

- value object
- aggregate
- entity
- repository

Check the answer of this question online on JavaChamp.com: [what is an aggregate?](#)

1.1.8. model driven design presentation layer

Author: Yasser Ibrahim

What is the role of the presentation layer in model driven design paradigm?

Please choose only one answer:

- implements entities persistence, and layers communication
- encapsulates the core of the business logic
- renders output to the user and gathers user gestures
- harmonize and monitors the application activities and its progress

Check the answer of this question online on JavaChamp.com: [model driven design presentation layer](#)

1.1.9. separate ways pattern

Author: Yasser Ibrahim

When would you use the separate ways pattern?

Please choose only one answer:

- to create domain independent context maps
- to create domain context maps that interact with legacy systems
- to create highly intersecting domain context maps

Check the answer of this question online on JavaChamp.com: [separate ways domain pattern](#)

1.1.10. independent domain context maps

Author: Yasser Ibrahim

which design pattern you would use to create independent domain context maps?

Please choose only one answer:

- open host services pattern
- anticorruption layers pattern
- separate ways pattern
- shared kernel pattern
- customer supplier pattern

Check the answer of this question online on JavaChamp.com: [separate ways pattern](#)

1.1.11. How to implement an Entity?

Author: Yasser Ibrahim

How should you implement an Entity?

Please choose only one answer:

- mutable object
- immutable object
- both are correct

Check the answer of this question online on JavaChamp.com: [how should you implement an entity?](#)

1.1.12. aggregate design pattern

Author: Yasser Ibrahim

Which domain design pattern you would use to state object ownership and boundaries?

Please choose only one answer:

- aggregate design pattern
- factory design pattern
- repository design pattern

Check the answer of this question online on JavaChamp.com: [aggregate design pattern](#)

1.1.13. shared kernel pattern

Author: Yasser Ibrahim

When would you use the shared kernel pattern?

Please choose only one answer:

- to create domain independent context maps
- to create domain context maps that interact with legacy systems
- to create highly intersecting domain context maps

Check the answer of this question online on JavaChamp.com: [shared kernel domain pattern](#)

1.1.14. domain driven design patterns

Author: Yasser Ibrahim

which of the following are considered domain driven design patterns? (choose 2)

Please choose all the answers that apply:

- aggregates
- layered architecture
- model view controller (MVC)
- Data Access Object (DAO)

Check the answer of this question online on JavaChamp.com: [domain driven design patterns](#)

1.1.15. how would you implement a bank account object?

Author: Yasser Ibrahim

how would you implement a bank account object?

Please choose only one answer:

- as transfer object
- as a value object
- as an entity

Check the answer of this question online on JavaChamp.com: [how would you implement a bank account object?](#)

1.1.16. model driven design Domain layer

Author: Yasser Ibrahim

What is the role of the Domain layer in model driven design paradigm?

Please choose only one answer:

- harmonize and monitors the application activities and its progress
- renders output to the user and gathers user gestures
- encapsulates the core of the business logic
- implements entities persistence, and layers communication

Check the answer of this question online on JavaChamp.com: [model driven design domain layer](#)

1.1.17. anticorruption layers pattern

Author: Yasser Ibrahim

When would you use the anticorruption layers pattern?

Please choose only one answer:

- to create domain independent context maps
- to create domain context maps that interact with legacy systems
- to create highly intersecting domain context maps

Check the answer of this question online on JavaChamp.com: [anticorruption layer pattern](#)

1.1.18. Domain Driven Design programming language

Author: Yasser Ibrahim

What is the most suitable programming language concept for implementing the model of Domain Driven Design paradigm?

Please choose only one answer:

- Procedural languages such as C, Pascal
- Object-oriented programming such as Java, C++
- Aspect oriented programming such as AspectJ

Check the answer of this question online on JavaChamp.com: [domain driven design programming language](#)

1.1.19. object creation and storage design pattern

Author: Yasser Ibrahim

Which domain design pattern handles object creation and storage?

Please choose all the answers that apply:

- aggregate design pattern
- factory design pattern
- repository design pattern

Check the answer of this question online on JavaChamp.com: [object creation and storage design pattern](#)

1.1.20. How to implement a value object?

Author: Yasser Ibrahim

How should you implement a value object?

Please choose only one answer:

- mutable object
- immutable object
- both are correct

Check the answer of this question online on JavaChamp.com: [how to implement a value object?](#)

1.1.21. domain context maps with legacy systems

Author: Yasser Ibrahim

which design pattern you would use to create domain context maps that interact with legacy systems?

Please choose all the answers that apply:

- open host services pattern
- anticorruption layers pattern
- separate ways pattern
- shared kernel pattern
- customer supplier pattern

Check the answer of this question online on JavaChamp.com: [open host services pattern](#)

1.1.22. model driven design Infrastructure layer

Author: Yasser Ibrahim

What is the role of the Infrastructure layer in model driven design paradigm?

Please choose only one answer:

- encapsulates the core of the business logic
- implements entities persistence, and layers communication
- renders output to the user and gathers user gestures
- harmonize and monitors the application activities and its progress

Check the answer of this question online on JavaChamp.com: [model driven design infrastructure layer](#)

1.1.23. what is true about domain services?

Author: Yasser Ibrahim

what is true about domain services?

Please choose only one answer:

- the operations performed by a domain service are typically stateful
- the operations performed by a domain service are typically comprehensive and span more than one object in the domain
- a domain service is usually persisted in the application backing data store

Check the answer of this question online on JavaChamp.com: [what is true about domain services?](#)

1.1.24. how would you implement the address object?

Author: Yasser Ibrahim

For an e-Shop web application, the customer will provide his personal information for shipment of the ordered products. You would like to attach residence, invoice, delivery addresses to a person object.

how would you implement the address object?

Please choose only one answer:

- as an entity
- as a value object
- as a transfer object

Check the answer of this question online on JavaChamp.com: [how would you implement the address object?](#)

1.1.25. open host services pattern

Author: Yasser Ibrahim

When would you use the open host services pattern?

Please choose only one answer:

- to create domain independent context maps
- to create domain context maps that interact with legacy systems
- to create highly intersecting domain context maps

Check the answer of this question online on JavaChamp.com: [domain driven design open host services pattern](#)

1.1.26. entities vs. value objects

Author: Yasser Ibrahim

what are the differences between entities and value objects?

Please choose all the answers that apply:

- entities are immutable objects while value objects are mutable objects
- an entity has a unique identity while a value object doesn't
- sharing value object is thread safe as it is immutable
- sharing an entity is thread safe as it has a unique identity

Check the answer of this question online on JavaChamp.com: [entities vs. value objects](#)

1.1.27. customer supplier pattern

Author: Yasser Ibrahim

When would you use the customer supplier pattern?

Please choose only one answer:

- to create domain independent context maps
- to create domain context maps that interact with legacy systems
- to create highly intersecting domain context maps

Check the answer of this question online on JavaChamp.com: [ddd customer supplier pattern](#)

