Gang of Four (GOF) Java Design Patterns Mock Exams

http://www.JavaChamp.com

Open Certification Plattform

Authors: N. Ibrahim, Y. Ibrahim

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

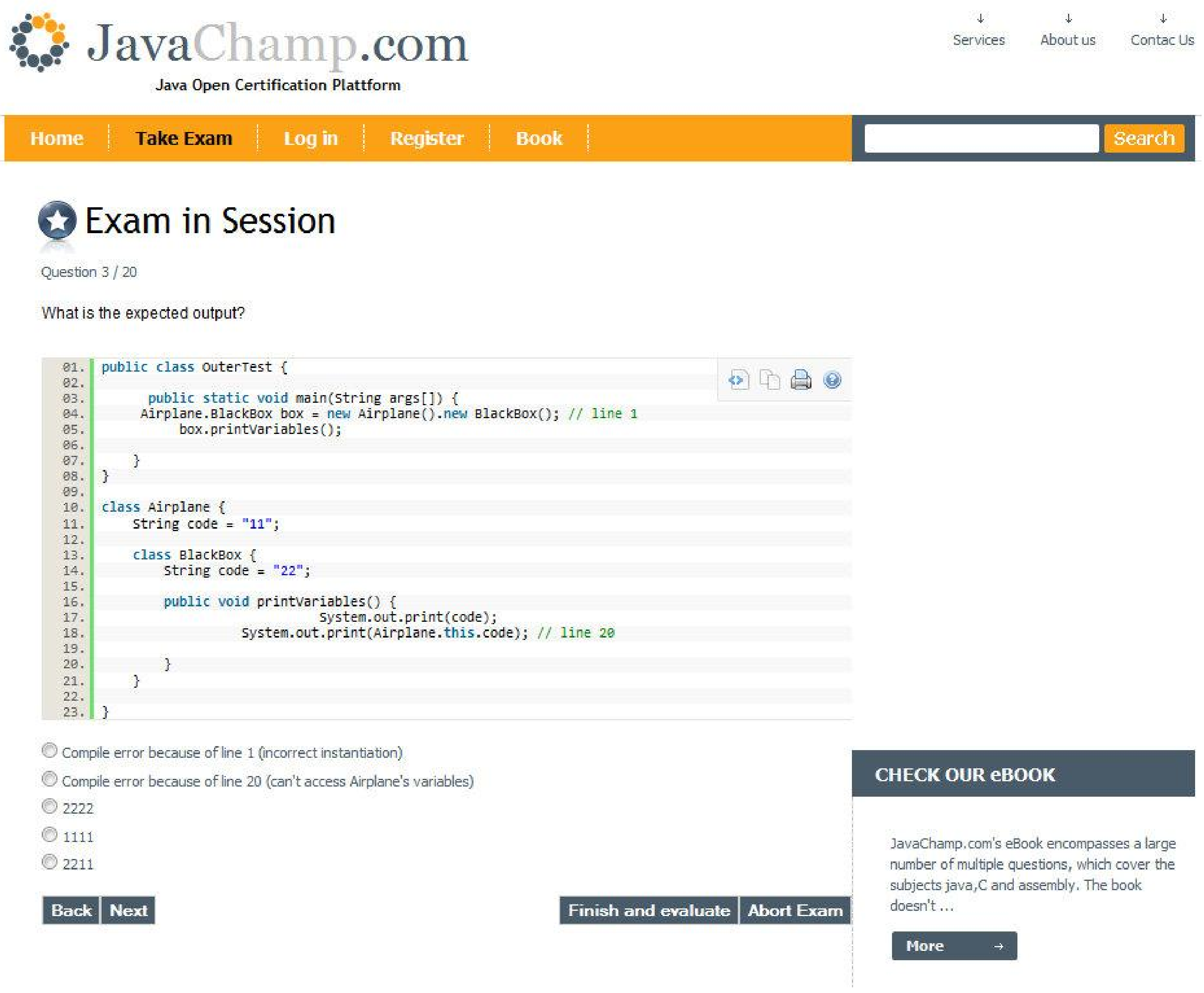
Why to register in our community?

Registeration is optional and free but brings many adavantages

- 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 ...



Exam in Session

Question 3 / 20

What is the expected output?

```
public class OuterTest {

      public static void main(String args[]) {
     Airplane.BlackBox box = new Airplane().new BlackBox(); // line 1
          box.printVariables();

     }
}

class Airplane {
    String code = "11";

    class BlackBox {
        String code = "22";

        public void printVariables() {
                            System.out.print(code);
                  System.out.print(Airplane.this.code); // line 20

        }
    }
}
```

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

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 ...

Copyright

Table of Contents

1. Chapter: Gang of Four Design Patterns

Chapter Description and Objectives

1. Creational Patterns

Exam Category Description and Objectives

1.1.1. Factory method design pattern intent

Author: Yasser Ibrahim

Which design pattern you would you use to control the creation of an object based on a established interface, while allowing the concrete implementation to determine the subclass to construct.

Please choose only one answer:

- Singleton design pattern
- Builder Factory design pattern
- Prototype factory design pattern
- Factory method design pattern

Check the answer of this question online on JavaChamp.com: factory method design pattern intent

1.1.2. Prototype design pattern intent

Author: Yasser Ibrahim

Which design pattern you would you use to have a prototypical instance determine the concrete class of object being created?

Please choose only one answer:

- Prototype factory design pattern
- Virtual prototype design pattern
- Abstract prototype design pattern
- Prototype design pattern

Check the answer of this question online on JavaChamp.com: prototype design pattern intent

1.1.3. Prototype design pattern applicability

Author: Yasser Ibrahim

When would you use the GOF Prototype design pattern? (choose all that apply)

Please choose all the answers that apply:

- To abstract steps of construction of complex objects
- To determine the classes to be instantiated at run time
- To avoid the proliferation of the class hierarchy
- To restrict class instantiation to one object

Check the answer of this question online on JavaChamp.com: prototype design pattern applicability

1.1.4. Singleton design pattern intent

Author: Yasser Ibrahim

Which design pattern you would you use to limit the class instantiation to one object?

Please choose only one answer:

- Factory Method Design Pattern
- Builder design pattern
- Prototype design pattern
- Singleton design pattern

Check the answer of this question online on JavaChamp.com: singleton design pattern intent

1.1.5. Builder design pattern applicability

Author: Yasser Ibrahim

When would you use the GOF builder design pattern? (choose all that apply)

Please choose all the answers that apply:

- to abstract steps of construction of complex objects
- to build different representations of complex objects based on the concrete implementations of construction procedure
- to establish an interface for creating an object, but let the concrete implementations decide which subclass to instantiate
- to encapsulate a family of individual factories that have a common theme

Check the answer of this question online on JavaChamp.com: builder design pattern applicability

1.1.6. Singleton design pattern implementation example

Author: Yasser Ibrahim

Which GOF design pattern is applied in the code snippet below?

```
public class PrintSpooler {

private static final PrintSpooler INSTANCE = new PrintSpooler();

private PrintSpooler() {}

public static PrintSpooler getInstance() {

return INSTANCE;

}

}
```

Please choose only one answer:

- PrintSpooler design pattern
- Spooler design pattern
- Singleton design pattern
- Factory design pattern
- Abstract Singleton design pattern

Check the answer of this question online on JavaChamp.com: gof singleton design pattern implementation example

1.1.7. Prototype design pattern consequences

Author: Yasser Ibrahim

What are the consequences of applying the GOF Prototype pattern?

Please choose all the answers that apply:

- each concrete prototype class must implement the clone method
- it makes it easier for a certain family of objects to work together
- it enable the client code to register an new concrete prototype instance at run time
- it reduces of the class hierarchy as compared to the other factory design patterns

Check the answer of this question online on JavaChamp.com: prototype design pattern consequences

1.1.8. Abstract Factory Pattern participants

Author: Yasser Ibrahim

Which of the following are participants in the GOF abstract factory design pattern ?

Please choose all the answers that apply:

- Factory methods
- Factory constructors
- Abstract factory
- Abstract product
- Concrete factory

Check the answer of this question online on JavaChamp.com: participants in the gof abstract factory design pattern

1.1.9. Factory method design pattern java usages

Author: Yasser Ibrahim

Which design pattern is used in the Java Database connectivity JDBC(TM)?

Please choose only one answer:

- Builder design pattern
- Factory method design pattern
- Abstract Factory design Pattern
- Singletone design Pattern

Check the answer of this question online on JavaChamp.com: factory method design pattern java usages

1.1.10. Factory method design pattern other names

Author: Yasser Ibrahim

The factory method design pattern is also known as:

Please choose only one answer:

- Abstract factory
- Abstract Constructor
- Virtual factory
- Virtual Constructor

Check the answer of this question online on JavaChamp.com: factory method design pattern other names

1.1.11. Factory method design pattern consequences

Author: Yasser Ibrahim

What are the consequences of applying the GOF factory method pattern?

Please choose all the answers that apply:

- it decouples the client code from the application specific classes
- it makes the designed product families exchangeable
- it establishes a flexible mechanism for instantiating an object in comparison to the usual java constructor instantiation.

Check the answer of this question online on JavaChamp.com: factory method design pattern consequences

1.1.12. Factory method design pattern participants

Author: Yasser Ibrahim

Which of the following are participants in the GOF builder design pattern?

Please choose all the answers that apply:

- Creator
- Product
- Refined Abstraction
- Abstract factory

Check the answer of this question online on JavaChamp.com: factory method design pattern participants

1.1.13. Factory method design pattern applicability

Author: Yasser Ibrahim

When would you use the GOF Factory method design pattern? (choose all that apply)

Please choose all the answers that apply:

- To ensure that a certain group of related objects are used together
- To control the creation of an object based on a established interface
- To allow the concrete implementation to determine the subclass to construct.
- To abstract steps of construction of complex objects

Check the answer of this question online on JavaChamp.com: factory method design pattern applicability

1.1.14. Singleton design pattern consequences

Author: Yasser Ibrahim

What are the consequences of applying the GOF Singleton pattern?

Please choose all the answers that apply:

- it introduces thread safety issue when the singleton instance is instantiated on demand
- the client code can creates multiple instances at run time
- it reduces of the class hierarchy as compared to the other factory design patterns
- it makes it easier for a certain family of objects to work together

Check the answer of this question online on JavaChamp.com: singleton design pattern consequences

1.1.15. Builder design pattern participants

Author: Yasser Ibrahim

Which of the following are participants in the GOF builder design pattern?

Please choose all the answers that apply:

- Builder interface
- Constructor Interface
- Director Interface
- Concrete Builder class
- Concrete constructor class

Check the answer of this question online on JavaChamp.com: builder design pattern participants

1.1.16. Builder design pattern consequences

Author: Yasser Ibrahim

What are the consequences of applying the Builder design pattern?

Please choose all the answers that apply:

- it makes the designed product hierarchies exchangeable
- it's easier to introduce new product implementation
- it separates the prodcut construction from it's representation
- the director has fine control over the product creation procedure

Check the answer of this question online on JavaChamp.com: builder design pattern consequences

1.1.17. abstract factory pattern consequences

Author: Yasser Ibrahim

What are the consequences of applying the abstract factory patter?

Please choose all the answers that apply:

- it will be much easier to introduce new family of products
- it makes it easier for a certain family of objects to work together
- it makes it easier for the client to deal with tree-structured data
- it makes the designed product families exchangeable

Check the answer of this question online on JavaChamp.com: abstract factory pattern consequences

1.1.18. When to use Singleton Pattern?

Author: Java Champ

You want all the clients using class A to use the same instance of class A, what should you do to achieve this goal?

Please choose only one answer:

- mark class A final
- mark class A abstract
- apply the Singleton pattern to class A
- apply the Memento pattern to class A

Check the answer of this question online on JavaChamp.com: when to use singleton pattern?

1.1.19. Singleton design pattern participants

Author: Yasser Ibrahim

Which of the following are participants in the GOF Singleton design pattern?

Please choose all the answers that apply:

- Abstract Singleton
- Singleton
- Concrete Singleton
- Singleton factory

Check the answer of this question online on JavaChamp.com: singleton design pattern participants

1.1.20. Singleton design pattern applicability

Author: Yasser Ibrahim

When would you use the GOF Singleton design pattern? (choose all that apply)

Please choose all the answers that apply:

- to ensure that a certain group of related objects are used together
- to limit the class instantiation to one object
- to provide global access to once instance across the system
- to abstract steps of construction of complex objects

Check the answer of this question online on JavaChamp.com: singleton design pattern applicability

1.1.21. Builder design pattern intent

Author: Yasser Ibrahim

Which design pattern you would you use to decouple the creation procedure of a complex object from it's concrete instance to be able to apply that procedure on variety of implementations.

Please choose only one answer:

- Factory builder design pattern
- Method Builder design pattern
- Builder design pattern
- Factory method design pattern

Check the answer of this question online on JavaChamp.com: builder design pattern intent

1.1.22. how to implement the Singleton design pattern?

Author: Yasser Ibrahim

how to implement the Singleton design pattern? specify all the need steps that apply.

Please choose all the answers that apply:

- add final modifier to the Class declaration
- add final modifier to the constructor declaration
- add private or protected modifier to the constructor declaration
- introduce a final static of the Singleton class
- introduce a static getter method for the Singleton instance
- The class name must be Singleton

Check the answer of this question online on JavaChamp.com: how to implement the singleton design pattern?

1.1.23. What is Factory Method pattern used for?

Author: Java Champ

It is also known as Virtual Constructor and it is used to define an interface for creating an object but letting the subclass decide which class to instantiate, this pattern is :

Please choose only one answer:

- Builder
- Abstract Factory
- Prototype
- Factory Method

Check the answer of this question online on JavaChamp.com: what is factory method pattern used for?

1.1.24. Singleton design pattern traditional implementation

Author: Yasser Ibrahim

Which of the following code snippet represents a Singleton design pattern implementation?

Please choose only one answer:

-
  ```
  public class PrintSpooler {
    public PrintSpooler INSTANCE = new PrintSpooler();
    public PrintSpooler () {}
    public static PrintSpooler getInstance() {
       return INSTANCE;
    }
  }
  ```

-
  ```
  public class PrintSpooler {
    private PrintSpooler INSTANCE = new PrintSpooler();
    private PrintSpooler () {}
    public static PrintSpooler getInstance() {
       return INSTANCE;
    }
  }
  ```

-
  ```
  public class PrintSpooler {
    private static final PrintSpooler INSTANCE = new PrintSpooler();
    private PrintSpooler () {}
    public static PrintSpooler getInstance() {
       return INSTANCE;
    }
  }
  ```

-
  ```
  public class PrintSpooler {
    private final PrintSpooler INSTANCE = new PrintSpooler();
    private PrintSpooler () {}
    public static PrintSpooler getInstance() {
       return INSTANCE;
    }
  }
  ```

Check the answer of this question online on JavaChamp.com: singleton design pattern traditional implementation

1.1.25. Builder design pattern forces

Author: Yasser Ibrahim

What would lead you to apply the builder design pattern?

Please choose all the answers that apply:

- To abstract steps of construction of objects so that different implementations
- To apply the same object construction procedure on variety of representations
- To translates one interface for a class into a compatible interface
- To restrict instantiation of a class to one object

Check the answer of this question online on JavaChamp.com: builder design pattern forces

1.1.26. Factory Method Design Pattern intent

Author: Java Champ

Which pattern is most appropriate when a decision must be made at the time a class is instantiated?

Please choose only one answer:

- Bridge
- Composite
- Factory Method
- Command

Check the answer of this question online on JavaChamp.com: what are gof creational design patterns good for?

1.1.27. Abstract Factory Pattern applicability

Author: Yasser Ibrahim

When would you use the GOF abstract factory pattern? (choose all that apply)

Please choose all the answers that apply:

- To design a structure which has a uniform interface for both compositions of objects and individual objects.
- To ensure that a certain group of related objects are used together
- The client which uses your design is not dependent on how the object are created or connected together
- To decouple the creation of objects from their usage

Check the answer of this question online on JavaChamp.com: abstract factory pattern applicability

1.1.28. Abstract Factory desing Pattern intent

Author: Java Champ

Given the following scenario:
You want to create families of related objects, to be used interchangeably to configure you application. What is the most appropriate GoF pattern to use?

Please choose only one answer:

- Chain of Responsibility
- Abstract Factory
- Builder
- Observer

Check the answer of this question online on JavaChamp.com: when to use abstract factory pattern?

1.1.29. Prototype design pattern participants

Author: Yasser Ibrahim

Which of the following are participants in the GOF Prototype design pattern?

Please choose all the answers that apply:

- Prototype
- Abstract Prototype
- Virtual Prototype
- Concrete Prototype

Check the answer of this question online on JavaChamp.com: prototype design pattern participants

2. Structural Patterns

Exam Category Description and Objectives

1.2.1. When to use Decorator Pattern?

Author: Java Champ

It is also known as Wrapper, it is used when subclassing is not possible or practical to add functionality and it is used to add functionality at runtime. This pattern is :

Please choose only one answer:

- Composite
- Adapter
- Decorator
- Proxy

Check the answer of this question online on JavaChamp.com: when to use decorator pattern?

1.2.2. What is the difference between an adapter and a decorator?

Author: Java Champ

What is the difference between an adapter and a decorator? (choose two)

Please choose all the answers that apply:

- The adapter adds no functionalities to the adaptee class, whereas the Decorator extends the functionality of the object
- The adapter is a creational pattern, whereas the decorator is a structural design pattern
- Both introduce a level of indirection between a client class and a class it uses

Check the answer of this question online on JavaChamp.com: what is the difference between an adapter and a decorator?

1.2.3. When to use Flyweight Pattern?

Author: Java Champ

It is a pattern used to minimize memory use by letting similar objects share as much data as possible stored in a sharable object.

Please choose only one answer:

- Flyweight
- Composite
- Iterator
- Momento

Check the answer of this question online on JavaChamp.com: when to use flyweight pattern?

1.2.4. Composite design pattern participants

Author: Yasser Ibrahim

Which of the following are participants in the GOF Composite design pattern? (choose all that apply)

Please choose all the answers that apply:

- Component
- Compound
- Leaf
- Connection
- Composite

Check the answer of this question online on JavaChamp.com: composite design pattern participants

1.2.5. bridge design pattern intent

Author: Yasser Ibrahim

Which design pattern you would you use to decouple an abstraction from its implementation so that the two can vary independently?

Please choose only one answer:

- Adapter design pattern
- bridge design pattern
- Facade design pattern
- Composite bridge design pattern

Check the answer of this question online on JavaChamp.com: gof bridge design pattern

1.2.6. Gang of Four Structural Design Patterns

Author: Java Champ

Which of the following is a Gang of Four (GoF) Structural Design Pattern?

Please choose all the answers that apply:

- Composite
- Flyweight
- Singleton
- Method Factory

Check the answer of this question online on JavaChamp.com: [gang of four (gof) structural design pattern]

1.2.7. Adapter design pattern participants

Author: Yasser Ibrahim

Which of the following are participants in the GOF Adapter design pattern?

Please choose all the answers that apply:

- Adapter
- Adaptee
- Bridge
- Target interface

Check the answer of this question online on JavaChamp.com: adapter design pattern participants

1.2.8. Decorator design pattern consequences

Author: Yasser Ibrahim

What are the consequences of applying the GOF Decorator pattern?

Please choose all the answers that apply:

- the client code can gradually add more functionality as it nests more layers of decorators
- it assigns more functionality to an object without sub-classing it
- it makes it harder to identify the the decorated and the decorator objects
- the client code can traverse tree structures of arbitrary depth recursively

Check the answer of this question online on JavaChamp.com: gof decorator design pattern

1.2.9. Adapter design pattern consequences

Author: Yasser Ibrahim

What are the consequences of applying the GOF Adapter pattern?

Please choose all the answers that apply:

- it will improve the system performance
- it makes the client code easier by interacting with a single uniform interface
- it must modify the adaptee class
- it establishes a flexible mechanism for instantiating an object in comparison to the usual java constructor instantiation.

Check the answer of this question online on JavaChamp.com: adapter design pattern consequences

1.2.10. An example of Flyweight Pattern

Author: Java Champ

Which pattern is utilized in this example ?
Every character object in a word document has data about the graphical representation and the position, but to avoid redundant graphical representation data for objects to the same character, it is advisable to extract out these representation data to one shared object between these objects, and store only the position of each character internally in each object.

Please choose only one answer:

- Abstract Factory
- Method Factory
- Builder
- Flyweight

Check the answer of this question online on JavaChamp.com: an example of flyweight pattern

1.2.11. Bridge design pattern applicability

Author: Yasser Ibrahim

When would you use the GOF Bridge design pattern? (choose all that apply)

Please choose all the answers that apply:

- to implement lazy loading of expensive objects
- to improve system reliability and scalability
- to decouple an abstraction from its implementation so that the two can vary independently
- to hide the implementation details from the client code of your design

Check the answer of this question online on JavaChamp.com: gof bridge design pattern applicability

1.2.12. Composite design pattern java usages

Author: Yasser Ibrahim

Which design pattern is used in the Java AWT Abstract Window Toolkit?

Please choose only one answer:

- Adapter design pattern
- Composite design pattern
- Bridge design pattern
- Proxy design pattern

Check the answer of this question online on JavaChamp.com: composite design pattern java usages

1.2.13. Decorator design pattern participants

Author: Yasser Ibrahim

Which of the following are participants in the GOF Decorator design pattern?

Please choose all the answers that apply:

- Abstract Decorator
- Virtual Decorator
- Decorator
- Concrete Decorator
- Component

Check the answer of this question online on JavaChamp.com: decorator design pattern participants

1.2.14. Lazy Loading and Proxy Pattern?

Author: Java Champ

Which design pattern can be used to implement Lazy Loading?

Please choose only one answer:

- Adapter
- Mediator
- Proxy
- Chain of Responsibility

Check the answer of this question online on JavaChamp.com: lazy loading and proxy pattern

1.2.15. The Surrogate Pattern

Author: Java Champ

A pattern known as Surrogate is :

Please choose only one answer:

- Adapter
- Proxy
- Facade
- Method Factory

Check the answer of this question online on JavaChamp.com: the surrogate pattern

1.2.16. Composite design pattern applicability

Author: Yasser Ibrahim

When would you use the GOF Composite design pattern? (choose all that apply)

Please choose all the answers that apply:

- to decouple an abstraction from its implementation so that the two can vary independentl
- to translates an existing class interface into a compatible target interface
- to arrange object hierarchies such that the client code can access both the individual objects and compositions in a uniform manner
- to improve the system overall performance

Check the answer of this question online on JavaChamp.com: composite design pattern applicability

1.2.17. How to implement the GOF Adapter pattern?

Author: Yasser Ibrahim

What are the common implementation strategies of the GOF Adapter pattern?

Please choose all the answers that apply:

- Factory Adapter
- Object Adapter
- Class Adapter
- Interface Adapter

Check the answer of this question online on JavaChamp.com: how to implement the gof adapter pattern?

1.2.18. Decorator design pattern applicability

Author: Yasser Ibrahim

When would you use the GOF Decorator design pattern? (choose all that apply)

Please choose all the answers that apply:

- to translates an existing class interface into a compatible target interface
- to assign more functionality to an object without sub-classing it
- to decouple an abstraction from its implementation so that the two can vary independently
- to nest layers of decorators to add more functionality

Check the answer of this question online on JavaChamp.com: decorator design pattern applicability

1.2.19. bridge design pattern participants

Author: Yasser Ibrahim

Which of the following are participants in the GOF bridge design pattern ?

Please choose all the answers that apply:

- Abstraction
- Bridge
- RefinedAbstraction
- Class Generalization
- Implementor

Check the answer of this question online on JavaChamp.com: gof bridge design pattern participants

1.2.20. When to use the Composite Design Pattern?

Author: Java Champ

Which best defines the use of the Composite Design Pattern?

Please choose only one answer:

- the Composite design pattern allows the clients to build a complex object of smaller different ones
- the Composite design pattern allows adding and removing functionality dynamically
- the Composite design pattern allows the clients to treat individual objects and compositions uniformly

Check the answer of this question online on JavaChamp.com: when to use the composite design pattern?

1.2.21. bridge design pattern scenario

Author: Yasser Ibrahim

You design an application for an advertis-ement company, which produces different sort of publications such as books, articles, leaflets, etc.
The company publishes those products in many media formats such as printed material, CD, DVD, online websites, etc.

How you would model the company products hierarchy (Publications, Media)?

Please choose only one answer:

- Use composite design pattern to enable the publication and media hierarchies to be treated in the same way as a single instance of an object.
- Use the adapter design pattern to translates publication interface into a compatible media interface.
- Use bridge design pattern to build separate hierarchies for the publications and the media, to decouple the abstraction from its implementation so that the two can vary independently

Check the answer of this question online on JavaChamp.com: bridge design pattern

1.2.22. Decorator design pattern java usages

Author: Yasser Ibrahim

Which design pattern is used in the Java InputStream, OutputStream, Reader, Writer hierarchies?

Please choose only one answer:

- Adapter design pattern
- Decorator design pattern
- Composite design pattern
- Bridge design pattern

Check the answer of this question online on JavaChamp.com: decorator design pattern java usages

1.2.23. Bridge design pattern consequences

Author: Yasser Ibrahim

What are the consequences of applying the GOF Bridge pattern?

Please choose all the answers that apply:

- it makes implementing the client code easier by interacting with a single uniform interface
- it decouples an abstraction from its implementation so that the two can vary independently
- it will decrease the system extesibility
- the client code can traverse tree structures of arbitrary depth recursively

Check the answer of this question online on JavaChamp.com: bridge design pattern consequences

1.2.24. When to use the Adapter Design Pattern?

Author: Java Champ

You are trying to add a class already written in another application to serve clients, beside other classes, in your system. All other classes have the same interface, the incoming class has a totally different interface than the clients expect, but contains all required functionalities.
What kind of refactoring is needed to make this class fit in with minimum changes in your system?

Please choose only one answer:

- apply the Proxy Pattern
- apply the Adapter Pattern
- create a new class which implements the expected interface and copy and paste the code from the class in the other application to this new class

Check the answer of this question online on JavaChamp.com: when to use the adapter design pattern?

1.2.25. Adapter design pattern applicability

Author: Yasser Ibrahim

When would you use the GOF Adapter design pattern? (choose all that apply)

Please choose all the answers that apply:

- to translates an existing class interface into a compatible target interface
- to improve your system performance
- to transforming the client code data into appropriate format expected by the target interface
- to allow classes with incompatible interfaces to work together

Check the answer of this question online on JavaChamp.com: adapter design pattern applicability

1.2.26. Decorator design pattern forces

Author: Yasser Ibrahim

Which design pattern you would you use to assign more functionality to an object without sub-classing it?

Please choose only one answer:

- Bridge design pattern
- Adapter design pattern
- Composite design pattern
- Decorator design pattern

Check the answer of this question online on JavaChamp.com: decorator design pattern forces

1.2.27. virtual proxy pattern

Author: Java Champ

You are building an online makeup website, which provides beside text articles, makeup tutorials as videos files.

Which pattern is recommended to use in such a website to deliver these media resources?

Please choose only one answer:

- value list handler
- virtual proxy
- intercepting filter
- composite view

Check the answer of this question online on JavaChamp.com: virtual proxy pattern

1.2.28. Composite design pattern forces

Author: Yasser Ibrahim

Which design pattern you would you use to arrange object hierarchies such that the client code can access both the individual objects and compositions in a uniform manner ?

Please choose only one answer:

- Adapter design pattern
- Composite design pattern
- Composite adapter design pattern
- Composite factory design pattern

Check the answer of this question online on JavaChamp.com: composite design pattern forces

1.2.29. Composite design pattern consequences

Author: Yasser Ibrahim

What are the consequences of applying the GOF Composite pattern?

Please choose all the answers that apply:

- the client code can traverse tree structures of arbitrary depth recursively
- increases the coupling between the client code and the composite structre
- it will improve the system performance
- the client code can access both the individual objects and compositions in a uniform manner

Check the answer of this question online on JavaChamp.com: composite design pattern consequences

1.2.30. When to use Facade Pattern?

Author: Java Champ

Big portion of clients are remotely invoking methods in your system, the surface area exposed to clients includes fine-grained methods, which of course causes network overhead.
Which pattern would improve your system performance in this case?

Please choose only one answer:

- Chain of Responsibility Pattern
- Facade Pattern
- Command Pattern
- Adapter Pattern

Check the answer of this question online on JavaChamp.com: when to use facade pattern?

1.2.31. Decorator design pattern consequences

Author: Yasser Ibrahim

What are the consequences of applying the GOF Decorator pattern?

Please choose all the answers that apply:

- it makes the client code easier by interacting with a single uniform interface
- the client code can traverse tree structures of arbitrary depth recursively
- it will will not alter the class inheritance hierarchy to accommodate for the additional functionality
- it allows to nest layers of decorators to add more functionality

Check the answer of this question online on JavaChamp.com: decorator design pattern consequences

1.2.32. When to use Proxy Pattern?

Author: Java Champ

A Proxy Pattern is best used to : (choose three)

Please choose all the answers that apply:

- control access to a remote object
- assemble complex objects
- fetch a resource-intensive object when requested by a client
- store common redundant data between large number of objects
- extract out from the calling client the access or connection logic needed to call an object

Check the answer of this question online on JavaChamp.com: when to use proxy pattern?

1.2.33. Adapter design pattern forces

Author: Yasser Ibrahim

Which design pattern you would you use to translates an existing class interface into a compatible target interface?

Please choose only one answer:

- Proxy design pattern
- Adapter design pattern
- Facade design pattern
- Adapter factory design pattern

Check the answer of this question online on JavaChamp.com: java adapter

3. Behavioral Patterns

Exam Category Description and Objectives

1.3.1. Which pattern is applied in java.util.Enumeration?

Author: Java Champ

java.util.Enumeration is an example of which pattern?

Please choose only one answer:

- Iterator
- Command
- Observer
- Strategy

Check the answer of this question online on JavaChamp.com: which pattern is applied in java.util.enumeration?

1.3.2. When to use Strategy Pattern?

Author: Java Champ

A pattern that is intended to provide a means to define a family of algorithms and encapsulate each one as an object for interchangeable use:

Please choose only one answer:

- Strategy Pattern
- Abstract Factory Pattern
- Visitor Pattern
- State Pattern

Check the answer of this question online on JavaChamp.com: when to use strategy pattern?

1.3.3. An example of Chain of Responsibility Pattern

Author: Java Champ

Which is considered an example of Chain of Responsibility Pattern? (choose two)

Please choose all the answers that apply:

- java.awt.event.ComponentAdapter
- The Java Servlet filter framework
- javax.jms.QueueConnectionFactory
- java.awt.Toolkit
- Java exception handling

Check the answer of this question online on JavaChamp.com: an example of chain of responsibility pattern

1.3.4. When to use Chain of Responsibility Pattern?

Author: Java Champ

Which pattern to use when more than one object can handle a request, and the handler is unknown?

Please choose only one answer:

- Chain of Responsibility
- Command
- Strategy
- Observer

Check the answer of this question online on JavaChamp.com: when to use chain of responsibility pattern?

1.3.5. Publish-Subscribe and the Observer Pattern

Author: Java Champ

In the Publish-Subscribe messaging model, the subscribers register themselves in a topic and are notified when new messages arrive to the topic. Which pattern does most describe this model?

Please choose only one answer:

- Adapter
- Notifier
- Observer
- State

Check the answer of this question online on JavaChamp.com: publish subscribe and the observer pattern

1.3.6. Problem Forces to select Strategy Pattern

Author: Java Champ

Which are considered forces to select a Strategy Pattern? (choose two)

Please choose all the answers that apply:

- A client needs to use a family of related objects
- A change to an object requires changing other objects
- A client needs to choose from multiple algorithms
- Multiple classes are the same but differ only in their behaviors

Check the answer of this question online on JavaChamp.com: problem forces to select strategy pattern

1.3.7. What is Command Pattern?

Author: Java Champ

It is known as Action or Transaction and is used to encapsulate a request as an object to support rollback, logging, or transaction functionality

Please choose only one answer:

- Chain of Responsibility Pattern
- Command Pattern
- Observer Pattern
- Strategy Pattern

Check the answer of this question online on JavaChamp.com: what is command pattern?