Design pattern interview questions for Senior and experienced level

design pattern and software design question asked in interviewThese are questions which not only relates to design patterns but also related to software design. These questions requires some amount of thinking and experience to answer. In most of the cases interviewer is not looking for absolute answers but looking for your approach, how do you think about a problem, do you able to think through, do you able to bring out things which are not told to you. This is where experience come in picture, What are things you consider while solving a problem etc. overall these design questions kicks off your thought process. Some time interviewer ask you to write code as well so be prepare for that. you can excel in these questions if you know the concept, example and application of your programming and design skill. You can take help from Head First design pattern to learn more about about design pattern and object oriented analysis and design.

1. Give an example where you prefer abstract class over interface? (answer)

This is common but yet tricky design interview question. both interface and abstract class follow "writing code for interface than implementation" design principle which adds flexibility in code, quite important to tackle with changing requirement. here are some pointers which help you to answer this question:

1. In Java you can only extend one class but implement multiple interface. So if you extend a class you lost your chance of extending another class.

2. Interface are used to represent adjective or behavior e.g. Runnable, Clonable, Serializable etc, so if you use an abstract class to represent behavior your class can not be Runnable and Clonable at same time because you can not extend two class in Java but if you use interface your class can have multiple behavior at same time.

3. On time critical application prefer abstract class is slightly faster than interface.

4. If there is a genuine common behavior across the inheritance hierarchy which can be coded better at one place than abstract class is preferred choice. Some time interface and abstract class can work together also where defining function in interface and default functionality on abstract class.

To learn more about interface in Java check my post 10 things to know about Java interfaces

2. Design a Vending Machine which can accept different coins, deliver different products?

This is an open design question which you can use as exercise, try producing design document, code and Junit test rather just solving the problem and check how much time it take you to come to solution and produce require artifacts, Ideally this question should be solve in 3 hours, at least a working version.

3. You have a Smartphone class and will have derived classes like IPhone, AndroidPhone,WindowsMobilePhone

can be even phone names with brand, how would you design this system of Classes.

This is another design pattern exercise where you need to apply your object oriented design skill to come with a design which is flexible enough to support future products and stable enough to support changes in existing model.

4. When do you overload a method in Java and when do you override it? (answer)

Rather a simple question for experienced designer in Java. if you see different implementation of a class has different way of doing certain thing than overriding is the way to go while overloading is doing same thing but with different input. method signature varies in case of overloading but not in case of overriding in java.

5. Design ATM Machine ?

We all use ATM (Automated Teller Machine) , Just think how will you design an ATM ? for designing financial system one must requirement is that they should work as expected in all situation. so no matter whether its power outage ATM should maintain correct state (transactions), think about locking, transaction, error condition, boundary condition etc. even if you not able to come up exact design but if you be able to point out non functional requirement, raise some question , think about boundary condition will be good progress.

6. You are writing classes to provide Market Data and you know that you can switch to different vendors overtime like Reuters, wombat and may be even to direct exchange feed , how do you design your Market Data system.

This is very interesting design interview question and actually asked in one of big investment bank and rather common scenario if you have been writing code in Java. Key point is you will have a MarketData interface which will have methods required by client e.g. getBid(), getPrice(), getLevel() etc and MarketData should be composed with a MarketDataProvider by using dependency injection. So when you change your MarketData provider Client won't get affected because they access method form MarketData interface or class.

7. Why is access to non-static variables not allowed from static methods in Java

You can not access non-static data from static context in Java simply because non-static variables are associated with a particular instance of object while Static is not associated with any instance. You can also see my post why non static variable are not accessible in static context for more detailed discussion.

8. Design a Concurrent Rule pipeline in Java?

Concurrent programming or concurrent design is very hot now days to leverage power of ever increasing cores in

advanced processor and Java being a multi-threaded language has benefit over others. Do design a concurrent system key point to note is thread-safety, immutability, local variables and avoid using static or instance variables. you just to think that one class can be executed by multiple thread a same time, So best approach is that every thread work on its own data, doesn't interfere on other data and have minimal synchronization preferred at start of pipeline. This question can lead from initial discussion to full coding of classes and interface but if you remember key points and issues around concurrency e.g. race condition, deadlock, memory interference, atomicity, ThreadLocal variables etc you can get around it.

good design pattern and software design interview questions answers

Design pattern interview questions for Beginners

These software design and design pattern questions are mostly asked at beginners level and just informative purpose that how much candidate is familiar with design patterns like does he know what is a design pattern or what does a particular design pattern do ? These questions can easily be answered by memorizing the concept but still has value in terms of information and knowledge.

1. What is design patterns ? Have you used any design pattern in your code ?

Design patterns are tried and tested way to solve particular design issues by various programmers in the world. Design patterns are extension of code reuse.

2. Can you name few design patterns used in standard JDK library?

Decorator design pattern which is used in various Java IO classes, Singleton pattern which is used in Runtime , Calendar and various other classes, Factory pattern which is used along with various Immutable classes likes Boolean e.g. Boolean.valueOf and Observer pattern which is used in Swing and many event listener frameworks.

3. What is Singleton design pattern in Java ? write code for thread-safe singleton in Java

Singleton pattern focus on sharing of expensive object in whole system. Only one instance of a particular class is maintained in whole application which is shared by all modules. Java.lang.Runtime is a classical example of Singleton design pattern. You can also see my post 10 questions on Singleton pattern in Java for more questions and discussion. From Java 5 onwards you can use enum to thread-safe singleton.

4. What is main benefit of using factory pattern ? Where do you use it?

Factory pattern’s main benefit is increased level of encapsulation while creating objects. If you use Factory to create object you can later replace original implementation of Products or classes with more advanced and high performance implementation without any change on client layer. See my post on Factory pattern for more detailed explanation and benefits.

5. What is observer design pattern in Java

Observer design pattern is based on communicating changes in state of object to observers so that they can take there action. Simple example is a weather system where change in weather must be reflected in Views to show to public. Here weather object is Subject while different views are Observers. Look on this article for complete example of Observer pattern in Java.

6. Give example of decorator design pattern in Java ? Does it operate on object level or class level ?

Decorator pattern enhances capability of individual object. Java IO uses decorator pattern extensively and classical example is Buffered classes like BufferedReader and BufferedWriter which enhances Reader and Writer objects to perform Buffer level reading and writing for improved performance. Read more on Decorator design pattern and Java

7. What is MVC design pattern ? Give one example of MVC design pattern ?

8. What is FrontController design pattern in Java ? Give an example of front controller pattern ?

9. What is Chain of Responsibility design pattern ?

10.What is Adapter design pattern ? Give examples of adapter design pattern in Java?

These are left for your exercise, try finding out answers of these design pattern questions as part of your preparation.

Read more: http://javarevisited.blogspot.com/2012/06/20-design-pattern-and-software-design.html#ixzz4fFP2WDUa