FullStack Café

Unlock 3952 Answers

Sign In

Interview

3952 Full-Stack Interview

Questions & Answers

To Get Your Next Six-Figure Job Offer

Manage, Deploy & Measure Features at Scale with Rollout. Free 14 Day Trial.

ADS VIA CARBON

Check All Answers

Improve Your Dev Resume

.NET Core 51 ● ADO.NET 33 ASP.NET 60
ASP.NET MVC 33 ASP.NET Web API 33 AWS 44
Agile & Scrum 52 Android 113 Angular 120
AngularJS 62 Azure 58 Behavioral 112
Blockchain 42 Bootstrap 38 C# 112
CSS 50 Code Problems 26 Data Structures 30
Design Patterns 45 DevOps 39 Docker 55
Entity Framework 57 Flutter 68 Git 36
Golang 49 GraphQL 24 HTML5 55
Ionic 29 JSON 15 Java 147
JavaScript 116 Kotlin 68 LINQ 35
Laravel 41 MSMQ 16 Microservices 34
MongoDB 81 NoSQL 14 Node.js 95
OOP 53 PHP 82 PWA 22 Python 96
Questions to Ask 39 React 164 React Native 72
Reactive Programming 27 Redux 30 Ruby 84
Ruby on Rails 72 SOA & REST API 66 SQL 56
Software Architecture 38 Software Testing 26

29 Design Patterns Interview Questions (ANSWERED) To Crack in 2020

Design Patterns 45

> *In software engineering, a design pattern is a general repeatable solution to a commonly occurring problem in software design. A design pattern isn't a finished design that can be transformed directly into code. It is a description or template for how to solve a problem that can be used in many different situations. Follow along and refresh your knowledge of 29 Expert Design Patterns Interview Questions (answered and solved) for your next developer/architect tech interview.*

Q1: What is Design Patterns and why anyone should use them?

Design Patterns 45 Entry

Answer

Design patterns are a well-described solution to the most commonly encountered problems which occur during software development.

Design pattern represents the best practices evolved over a period of time by experienced software developers. They promote reusability which leads to a more robust and maintainable code.

Interview Coming Up? Check 45 Design Patterns Interview Questions

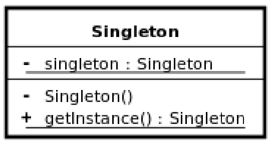
www.educba.com

Q2: What is Singleton pattern?

Design Patterns 45 Entry

Answer

Singleton pattern comes under *creational* patterns category and introduces a single class which is responsible to create an object while making sure that only single object gets created. This class provides a way to access its only object which can be accessed directly without need to instantiate the object of the class.



Interview Coming Up? Check 45 Design Patterns Interview Questions

refactoring.guru

Q3: What is Dependency Injection?

Design Patterns 45 Junior

Answer

Dependency injection makes it easy to create loosely coupled components, which typically means that components consume functionality defined by interfaces without having any first-hand knowledge of which implementation classes are being used.

Dependency injection makes it easier to change the behavior of an application by changing the components that implement the interfaces that define application features. It also results in components that are easier to isolate for unit testing.

Interview Coming Up? Check 45 Design Patterns Interview Questions

Pro ASP.NET Core MVC 2

☞ See All Questions Entity Framework 57

Agile & Scrum 52 LINQ 35 MSMQ 16

Reactive Programming 27 PHP 82

JavaScript 116 Python 96 NoSQL 14

Software Testing 26 Node.js 95

Webpack 32 DevOps 39

Q4: Name types of Design Patterns?

Design Patterns 45 Junior

Answer

Design patterns can be classified in three categories: Creational, Structural and Behavioral patterns.

- Creational Patterns - These design patterns provide a way to create objects while hiding the creation logic, rather than instantiating objects directly using new opreator. This gives program more flexibility in deciding which objects need to be created for a given use case.

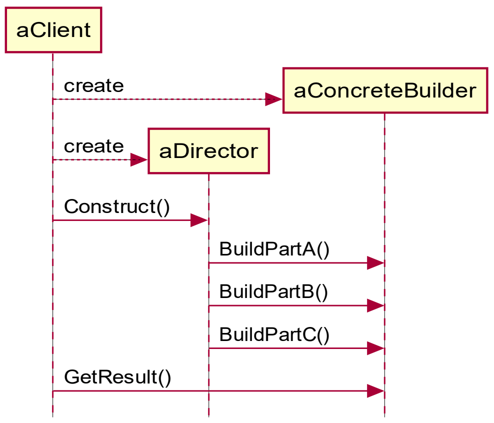
- Structural Patterns - These design patterns concern class and object composition. Concept of inheritance is used to compose interfaces and define ways to compose objects to obtain new functionalities.
- Behavioral Patterns - These design patterns are specifically concerned with communication between objects.

Q5: What is Builder pattern?

Design Patterns 45 Junior

Answer

Builder pattern builds a complex object using simple objects and using a step by step approach. This builder is independent of other objects.



The Director class is optional and is used to make sure that the building steps are executed in the *right order* with the right data by the right builder. It's about validation and delegation.

Builder/Director pattern's steps invocations could be semantically presented by *method chaining* or so called *Fluent Interface* syntax.

```
Pizza pizza = new Pizza.Builder()
                       .cheese(true)
                       .pepperoni(true)
                       .bacon(true)
                       .build();
```

Web Security 58 NoSQL 14

Ruby on Rails 72 UX Design 13

Q6: What is Filter pattern?

Design Patterns 45 Junior

Answer

Filter pattern or **Criteria pattern** is a design pattern that enables developers to filter a set of objects using different criteria and chaining them in a decoupled way through logical operations. This type of design pattern comes under *structural* pattern as this pattern combines multiple criteria to obtain single criteria.

Filter design pattern is useful where you want to add filters dynamically or you are implementing multiple functionalities and most of them require different filter criteria to filter something. In that case instead of hard coding the filters inside the functionalities, you can create filter criteria and re-use it wherever required.

```
List<Laptop> laptops = LaptopFactory.manufactureInBulk();
AndCriteria searchCriteria = new AndCriteria(
  new HardDisk250GBFilter(),
  new MacintoshFilter(),
  new I5ProcessorFilter());
List<Laptop> filteredLaptops = searchCriteria.meets(laptops);
```

Interview Coming Up? Check 45 Design Patterns Interview Questions

tutorialspoint.com

37 ASP.NET Interview Questions You Must Know

#.NET Core #ASP.NET

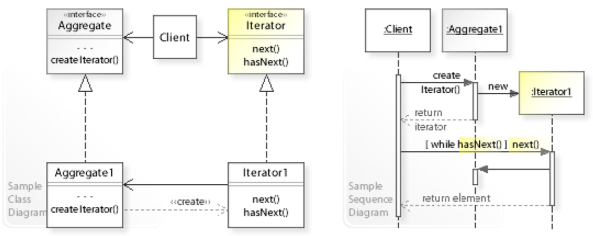
#ASP.NET Web API

Q7: What is Iterator pattern?

Design Patterns 45 Junior

Answer

Iterator pattern is very commonly used design pattern in Java and .Net programming environment. This pattern is used to get a way to access the elements of a collection object in sequential manner without any need to know its underlying representation. Iterator pattern falls under *behavioral* pattern category.



Interview Coming Up? Check 45 Design Patterns Interview Questions

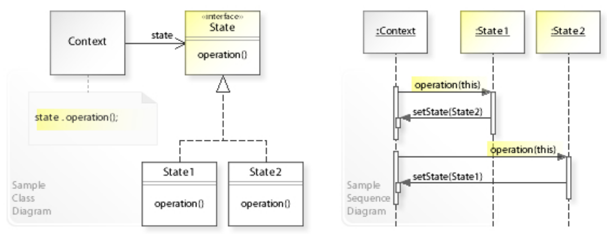
tutorialspoint.com

Q8: What is State pattern?

Design Patterns 45 Junior

Answer

In **State pattern** a class behavior changes based on its state. This type of design pattern comes under *behavior* pattern. In State pattern, we create objects which represent various states and a context object whose behavior varies as its state object changes.



Interview Coming Up? Check 45 Design Patterns Interview Questions

🔗 tutorialspoint.com

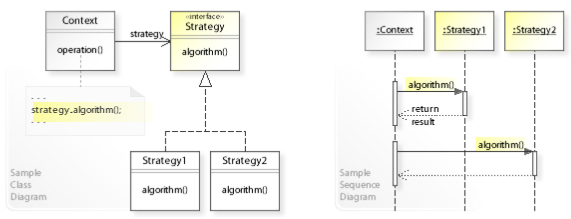
Q9: What is Strategy pattern?

Design Patterns 45 Junior

Answer

In **Strategy pattern**, a class behavior or its algorithm can be changed at run time. This type of design pattern comes under *behavior* pattern.

In Strategy pattern, we create objects which represent various strategies and a context object whose behavior varies as per its strategy object. The strategy object changes the executing algorithm of the context object.



Interview Coming Up? Check 45 Design Patterns Interview Questions

🔗 tutorialspoint.com

Q10: What is Memento pattern?

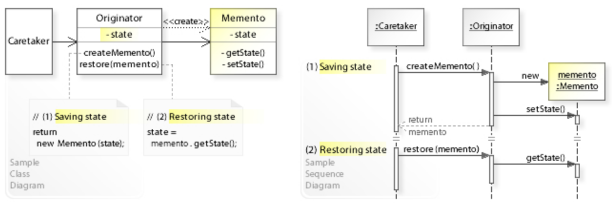
Design Patterns 45 Mid

Answer

Memento pattern is used to restore state of an object to a previous state. Memento pattern falls under *behavioral* pattern category.

Memento pattern uses three actor classes:

- *Memento* contains state of an object to be restored.
- *Originator* creates and stores states in Memento objects and
- *Caretaker* object is responsible to restore object state from Memento.



Interview Coming Up? Check 45 Design Patterns Interview Questions

tutorialspoint.com

See All 45 Design Patterns Q&A

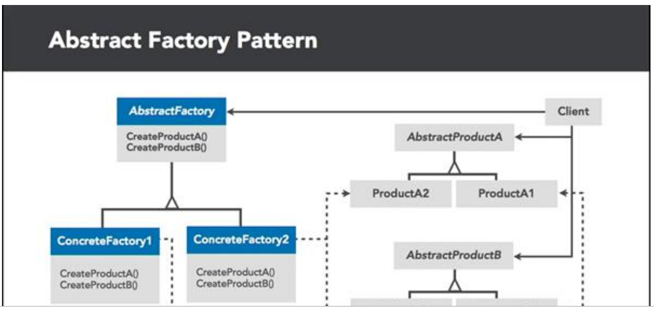
Q11: What is Abstract Factory pattern?

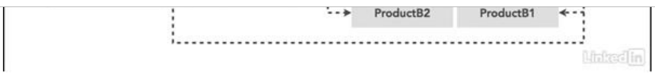
Design Patterns 45 Mid

Answer

Abstract Factory patterns work around a super-factory which creates other factories. This factory is also called as factory of factories. This type of design pattern comes under creational pattern as this pattern provides one of the best ways to create an object.

In Abstract Factory pattern an interface is responsible for creating a factory of related objects without explicitly specifying their classes. Each generated factory can give the objects as per the Factory pattern.





Pros:

- Follows the Open/Closed Principle.
- Allows building families of product objects and guarantees their compatibility.
- Avoids tight coupling between concrete products and code that uses them.
- Divides responsibilities between multiple classes.

Cons:

- Increases overall code complexity by creating multiple additional classes.

Interview Coming Up? Check 45 Design Patterns Interview Questions

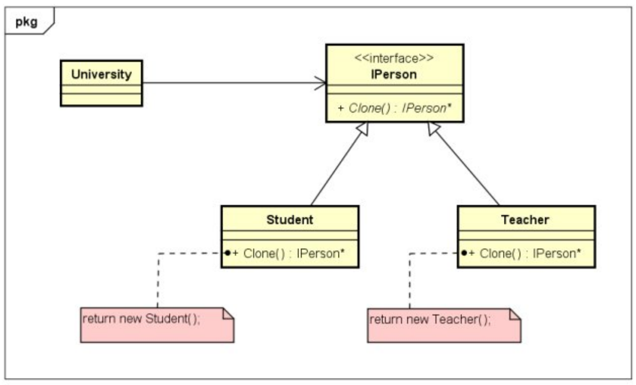
tutorialspoint.com

Q12: What is Prototype pattern?

Design Patterns 45 Mid

Answer

Prototype pattern refers to creating duplicate object while keeping performance in mind. This pattern involves implementing a prototype interface which tells to create a clone of the current object.



The Prototype pattern is used when creation of object directly is costly. For example, an object is to be created after a costly database operation. We can cache the object, returns its clone on next request and update the database as and when needed thus reducing database calls.

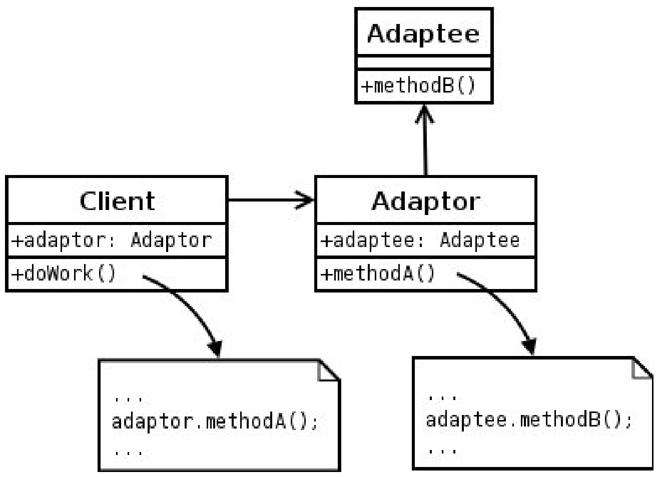
Interview Coming Up? Check 45 Design Patterns Interview Questions 　tutorialspoint.com

Q13: What is Adapter Pattern?

Design Patterns 45 　Mid

Answer

Adapter pattern works as a bridge between two incompatible interfaces. This pattern involves a single class which is responsible to join functionalities of independent or incompatible interfaces (adaptees).



A real life example could be a case of card reader which acts as an adapter between memory card and a laptop. You plugin the memory card into card reader and card reader into the laptop so that memory card can be read via laptop.

Interview Coming Up? Check 45 Design Patterns Interview Questions 　tutorialspoint.com

See All 45 Design Patterns Q&A

See All Questions 　Software Testing 26

Webpack 32 　Golang 49 　GraphQL 24

SQL 56 　Python 96 　Docker 55

Laravel 41 　MongoDB 81 　JavaScript 116

jQuery 51 　AWS 44 　JSON 15

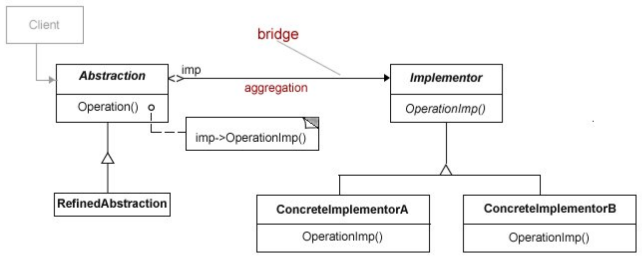
Q14: What is Bridge pattern?

Design Patterns 45 　Mid

Answer

Bridge pattern is used when we need to decouple an abstraction from its implementation so that the two can vary independently. This type of design pattern comes under *structural* pattern as this pattern decouples implementation class and abstract class by providing a bridge structure between them.

The bridge pattern is useful when both the class and what it does vary often. The class itself can be thought of as the abstraction and what the class can do as the implementation. The bridge pattern can also be thought of as two layers of abstraction.



This pattern involves an interface which acts as a bridge which makes the functionality of concrete classes independent from interface implementer classes. Both types of classes can be altered structurally without affecting each other.

The example of bridge pattern implementation is when:

```
                    ----Shape---
                   /            \
         Rectangle              Circle
        /         \            /      \
BlueRectangle  RedRectangle BlueCircle RedCircle
```

refactored to:

```
           ----Shape---                        Color
          /            \                       /   \
Rectangle(Color)   Circle(Color)            Blue    Red
```

or in general when:

```
          A
       /     \
      Aa      Ab
     / \     /  \
   Aa1 Aa2  Ab1 Ab2
```

refactored to:

```
      A         N
   /     \     / \
Aa(N) Ab(N)   1   2
```

Interview Coming Up? Check 45 Design Patterns Interview Questions

tutorialspoint.com

Q15: When should I use composite design pattern?

Design Patterns 45 Mid

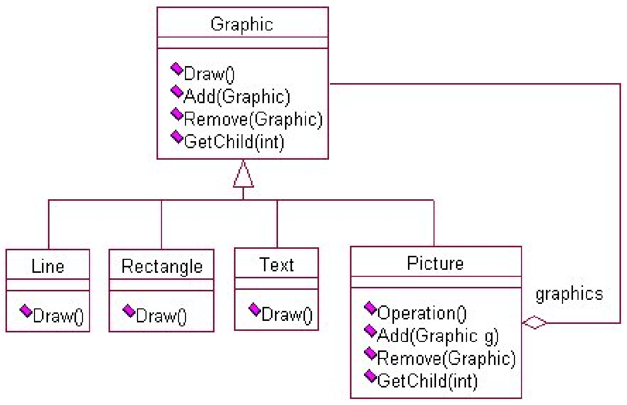
Answer

Use the **Composite Pattern** when

- you want to represent part-whole hierarchies of objects.
- you want clients to be able to ignore the difference between compositions of objects and individual objects. Clients will treat

all objects in the composite structure uniformly.

A common usage is a display system of graphic windows which can contain other windows and graphic elements such as images, text. The composite can be composed at run-time, and the client code can manipulate all the elements without concern for which type it is for common operations such as drawing. Another example is directories contain entries, each of which could be a directory.



Interview Coming Up? Check 45 Design Patterns Interview Questions

tutorialspoint.com

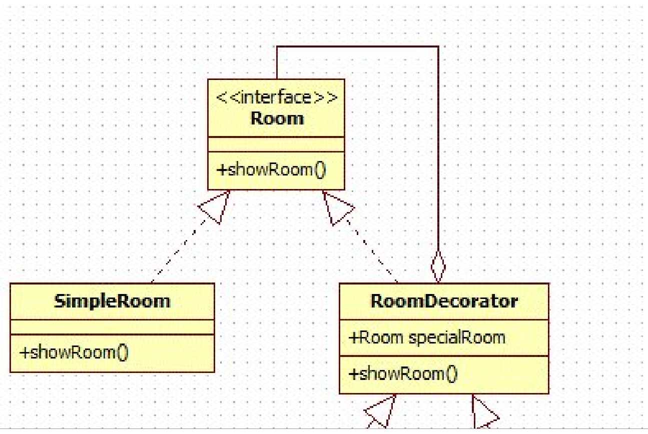
See All Questions Git 36 SQL 56 Reactive Programming 27 SOA & REST API 66 ADO.NET 33 Behavioral 112 Agile & Scrum 52 Entity Framework 57 Design Patterns 45 Software Testing 26 UX Design 13 JavaScript 116 WCF 33 Webpack 32 Blockchain 42

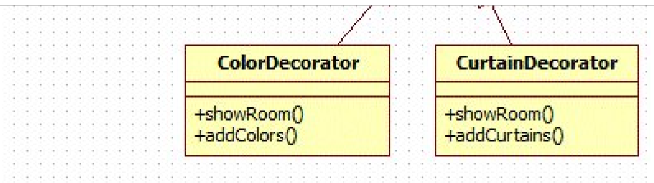
Q16: What is Decorator pattern?

Design Patterns 45 Mid

Answer

Decorator pattern allows a user to add new functionality to an existing object without altering its structure. This type of design pattern comes under *structural pattern* as this pattern acts as a wrapper to existing class.





This pattern creates a decorator class which wraps the original class and provides additional functionality keeping class methods signature intact.

Interview Coming Up? Check 45 Design Patterns Interview Questions tutorialspoint.com

See All 45 Design Patterns Q&A

Top 29 AngularJS Interview Questions (ANSWERED) You Will Be Asked Tomorrow

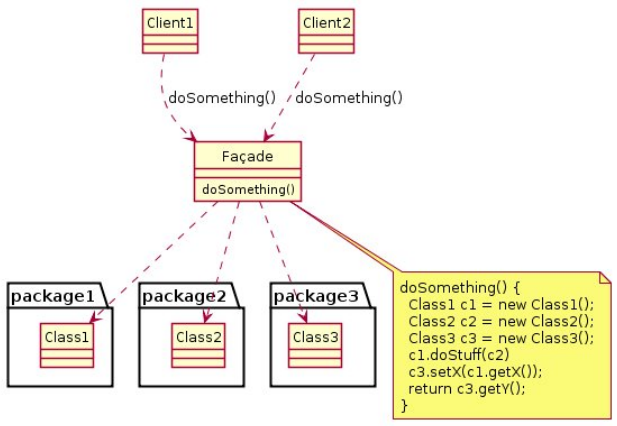
#AngularJS #JavaScript

Q17: What is Facade pattern?

Design Patterns 45 Mid

Answer

Facade pattern hides the complexities of the system and provides an interface to the client using which the client can access the system. This type of design pattern comes under *structural pattern* as this pattern adds an interface to existing system to hide its complexities.



```
doSomething() {
  Class1 c1 = new Class1();
  Class2 c2 = new Class2();
  Class3 c3 = new Class3();
  c1.doStuff(c2)
  c3.setX(c1.getX());
  return c3.getY();
}
```

This pattern involves a single class which provides simplified methods required by client and delegates calls to methods of existing system classes.

Interview Coming Up? Check 45 Design Patterns Interview Questions tutorialspoint.com

See All Questions Laravel 41 LINQ 35 Vue.js 41 Ionic 29 JavaScript 116 Docker 55 Web Security 58 XML & XSLT 41 Software Testing 26

Android 113 PHP 82 Ruby 84

Ruby on Rails 72

Q18: Can you give any good explanation what is the difference between Proxy and Decorator?

Design Patterns 45 Mid

Answer

Decorator Pattern focuses on dynamically adding functions to an object, while **Proxy Pattern** focuses on controlling access to an object.

Interview Coming Up? Check 45 Design Patterns Interview Questions

stackoverflow.com

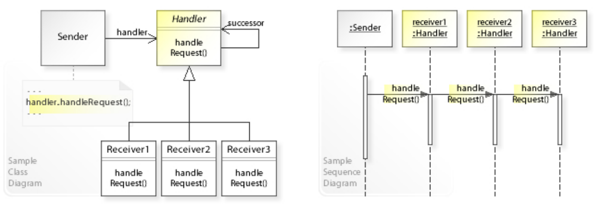
Q19: What is the Chain of Responsibility pattern?

Design Patterns 45 Mid

Answer

As the name suggests, the **chain of responsibility** pattern creates a chain of receiver objects for a request. This pattern decouples sender and receiver of a request based on type of request. This pattern comes under *behavioural* patterns.

In this pattern, normally each receiver contains reference to another receiver. If one object cannot handle the request then it passes the same to the next receiver and so on. The chain of responsibility is an object oriented version of the `if ... else if ... else if ....... else ... endif` idiom, with the benefit that the condition–action blocks can be dynamically rearranged and reconfigured at runtime.



Interview Coming Up? Check 45 Design Patterns Interview Questions

tutorialspoint.com

See All 45 Design Patterns Q&A

See All Questions Xamarin 83 Java 147

Laravel 41 Android 113 OOP 53

Design Patterns 45 Vue.js 41

Agile & Scrum 52 T-SQL 36 MSMQ 16

Data Structures 30 AngularJS 62

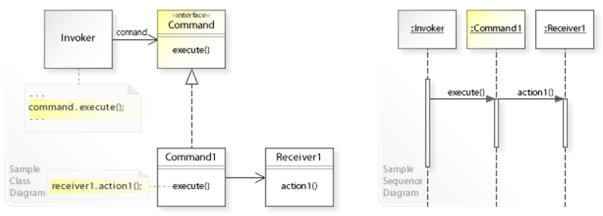
Node.js 95

Q20: What is Command pattern?

Design Patterns 45 Mid

Answer

Command pattern is a data driven design pattern and falls under *behavioural* pattern category. A request is wrapped under an object as *command* and passed to *invoker* object. *Invoker* object looks for the appropriate object which can handle this command and passes the command to the corresponding *receiver* which executes the command.



Interview Coming Up? Check 45 Design Patterns Interview Questions

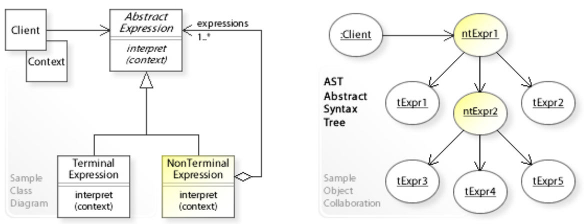
tutorialspoint.com

Q21: What is Interpreter pattern?

Design Patterns 45 Mid

Answer

Interpreter pattern provides a way to evaluate language grammar or expression. This type of pattern comes under *behavioral* pattern. This pattern involves implementing an expression interface which tells to interpret a particular context.



This class diagram means that an `AbstractExpression` is either a `TerminalExpression` or a `NonTerminalExpression`. If its a `NonTerminalExpression`, it is itself an aggregation of one or several `AbstractExpression`.

Any mechanism for interpreting formal languages suites this pattern perfectly, it can be anything: from a simple calculator to a C# parser.

Interview Coming Up? Check 45 Design Patterns Interview Questions

tutorialspoint.com

See All Questions T-SQL 36 LINQ 35

GraphQL 24 aws AWS 44 MongoDB 81
Golang 49 ADO.NET 33 Webpack 32
Java 147 Microservices 34 jQuery 51
Flutter 68 Blockchain 42

Q22: What is Observer pattern?

Design Patterns 45 **Mid**

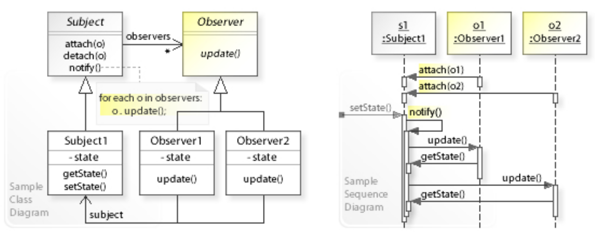
Answer

Observer pattern (also known as *Publish-Subscribe Pattern*) is used when there is one-to-many relationship between objects such as if one object is modified, its dependent objects are to be notified automatically. Observer pattern falls under *behavioral* pattern category.

An object with a one-to-many relationship with other objects who are interested in its state is called the *subject* or *publisher*. The *observers* are notified whenever the state of the *subject* changes and can act accordingly. The *subject* can have any number of dependent *observers* which it notifies, and any number of *observers* can subscribe to the *subject* to receive such notifications.

Observer pattern uses two actor classes:

- The Observer (os Subscriber) abstract class provides an `update()` method which will be called by the subject to notify it of the subject’s state change.
- The Subject (or Publisher) class is also an abstract class and defines four primary methods: `attach()`, `detach()`, `setState()`, and `notify()`



Interview Coming Up? Check 45 Design Patterns Interview Questions

sitepoint.com

See All 45 Design Patterns Q&A

Q23: How is Bridge pattern is different from Adapter pattern?

Design Patterns 45 **Senior**

Answer

Join FSC to see **Answer** to this Interview Question...

Join To Unlock Answer

See All Questions jQuery 51 AngularJS 62
SQL 56 Code Problems 26 LINQ 35

ADO.NET 33 Java 147 Ruby on Rails 72
Kotlin 68 UX Design 13 Bootstrap 38
JSON 15 Spring 87 CSS 50

Q24: When would you use the Builder Pattern? Why not just use a Factory Pattern?

Design Patterns 45 Senior

Answer

Join FSC to see **Answer** to this Interview Question...

Join To Unlock Answer

Q25: Why would I ever use a Chain of Responsibility over a Decorator?

Design Patterns 45 Senior

Answer

Join FSC to see **Answer** to this Interview Question...

Join To Unlock Answer

See All 45 Design Patterns Q&A

☞ See All Questions Reactive Programming 27
OOP 53 CSS 50 Web Security 58
Angular 120 NoSQL 14 AWS 44
Redux 30 Xamarin 83 .NET Core 51
Software Testing 26 Docker 55

Q26: What is the difference between Strategy design pattern and State design pattern?

Design Patterns 45 Senior

Answer

Join FSC to see **Answer** to this Interview Question...

Join To Unlock Answer

31 Redux Interview Questions (ANSWERED) React Devs Must Know in 2020

#React #Redux

Q27: Explain difference between the Facade, Proxy, Adapter and Decorator design patterns?

Design Patterns 45 Expert

Answer

Share this post to Unlock **Answer** to Expert Interview Question...

Share This Post To Unlock Expert Answers!

☞ See All Questions Python 96 Git 36 Xamarin 83 Code Problems 26 React 164 CSS 50 Java 147 WCF 33 Angular 120 Microservices 34 C# 112 Docker 55 Design Patterns 45

Q28: What's the difference between the Dependency Injection and Service Locator patterns?

Design Patterns 45 Expert

Answer

Share this post to Unlock **Answer** to Expert Interview Question...

Share This Post To Unlock Expert Answers!

See All 45 Design Patterns Q&A

Q29: What is the difference between the template patterns and the strategy pattern?

Design Patterns 45 Expert

Answer

Share this post to Unlock **Answer** to Expert Interview Question...

Share This Post To Unlock Expert Answers!

☞ See All Questions Agile & Scrum 52 PHP 82 Azure 58 Spring 87 AngularJS 62 Ionic 29 Blockchain 42 TypeScript 39 Xamarin 83 Software Testing 26 WCF 33

ASP.NET MVC 33 Vue.js 41 JSON 15

Unlock 3952 Answers Now

29 Web Services Interview Questions To Prepare For In 2020

#Microservices #SOA & REST API #Web Security

29 Web Services Interview Questions To Prepare For In 2020

A Web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable...

25 NoSQL Interview Questions (ANSWERED) You Must Know In 2020

#MongoDB #NoSQL #SQL

25 NoSQL Interview Questions (ANSWERED) You Must Know In 2020

Paradoxically the main reason behind the popularity of NoSQL data stores is the fact that their lack of ability to do advanced queries (joins, groupings, ranking and anal...

20 SOAP Interview Questions (ANSWERED) Developers Must Kill In 2020

#SOA & REST API

20 SOAP Interview Questions (ANSWERED) Developers Must Kill In 2020

SOAP stands for Simple Object Access Protocol. It defines the protocol used in communication between client and server. It is based on HTML, uses XML as a format and comm...

29+ Advanced XML Interview Questions (ANSWERED) Web Devs Must Know in 2020

#XML & XSLT

22 UX Design Interview Questions (ANSWERED) UX Designers Must Know [2020 UPDATE]

#Bootstrap #CSS #UX Design

29 Advanced Android Interview Questions (ANSWERED) For Senior App Developers (2020)

#Android

Guest Voice: Top 10 NodeJS Frameworks For Developers in 2020

#Node.js

35 Entity Framework Interview Questions (ANSWERED + .NET Core Updates) For 2020

#Entity Framework

See All FREE Preparation Plans