

spring-boot Interview Questions and Answer

1- What are stack and heap?

The JVM divides the memory into two parts: stack memory and heap memory.

Stack is used to store the order of method execution and local variables.

Heap memory stores the objects and uses dynamic memory allocation.

Stack stores references of objects that are in heap memory.

2- What is new with Java 8?

(Lambda) Functional programming, new APIs for date-time manipulation, a new streaming API, Optional class.

3- What is a design pattern? What are the design pattern categories?

Design patterns are the solution to common problems software developers face during software development. Creational Patterns, Structural Patterns, and Behavioral Patterns. These categories differ in the level of detail, complexity, and scale of applicability.

4- What are Singleton, Factory, Abstract Factory, Builder, Prototype, Adapter, Decorator, Command, Observer, Strategy, and Template design patterns? How do implement them?

SingletonIt ensures that only one instance of a class is created.It is implemented by making the constructor private. The static method accesses the static field.

FactoryThe Factory Method is a design pattern that provides an interface for creating objects in the superclass but allows subclasses to change that type of object to be created.Implement a Factory class that takes the name of the class to be produced and creates an object accordingly(e.g if the name is A, create an object of type A).

Abstract FactoryThis design pattern provides an interface for the creation of related or dependent objects without specifying concrete classes.In the abstract factory, an object is used; in the factory, a method is used.To implement it is necessary to create a factory class for each subclass. These factory classes to be created must derive from a super factory class whose type is interface or abstract.

BuilderIt provides convenience in assigning fields while objects are being created. It also frees programmers to implement multiple constructors. To implement we maintain a static inner class Builder. We pass the object of this inner class to the constructor. Fields are added step by step. An object of the wanted type gets created only when the build() is called.

A prototype design pattern that allows you to copy an object without your code becoming dependent on its classes.

To implement the Prototype design pattern, first, implement the Clonable interface and then override the clone method.

Adapter The Adapter Design pattern is implemented to reuse an existing class or interface class by adapting it to a different interface class at hand. We do not implement interfaces as they will not be used in our class (contrary to the Interface segregation of SOLID). Therefore, we create an adapter class that injects dependency in our class's constructor and implement it.

decorator is used to dynamically change the method an object inherits. During inheritance, the decorator class is inserted and the inherited method in this class is changed.

strategy defines multiple algorithms and lets the client applications pass the algorithm to be used as a parameter.

the template allows you to define a skeleton of an algorithm in a base class and let subclasses override the steps without changing the overall algorithm's structure.

6- What are checked and unchecked errors? What are their differences?

A checked exception is caught at compile time. For example IOException, and SQLException.

An unchecked exception is caught at runtime. For example Arithmetic, NullPointerException, and OutOfBounds.

7- Encapsulation?

It is the collection and protection of methods and variables in a class under a single unit.

8- The difference between “==” and .equals() ?

“==” checks if both objects point to the same memory location whereas “equals()” evaluates the comparison of values in the objects.

9- What are immutable and mutable? Why do we prefer immutable objects?

Mutable objects can be changed to any value or state without adding a new object.

In the case of immutable objects, whenever we change the state of the object, a new object will be created.

Prefer immutable because Immutable classes are thread-safe.

10- How to make an object immutable?

An immutable object is created by using the final class, private fields, and final mutable objects, and has only getters.

11- What is the final keyword and where it is used?

There are final classes, final methods, and final fields.

No class can inherit from the final class.

The final method cannot be overridden.

The final field's value cannot be changed once its value is given.

12- What are the access modifiers?

Private: Only accessible from the same class.

Default: Only accessible from the same package.

Protected: Only accessible from the same package, also from classes that inherit.

Public: Can be accessed from everywhere.

13- What is the synchronized keyword?

We need to synchronize the shared resources to ensure that at a time only one thread is able to access the shared resource.

14- What are the JPA fetch strategies (LAZY, EAGER)?

FetchType.EAGER: The persistence provider must load the related annotated field or property. This is the default behavior for *@Basic*, *@ManyToOne*, and *@OneToOne* annotated fields.

FetchType.LAZY: The persistence provider should load data when it's first accessed, but can be loaded eagerly.

16- What are the properties of transactions (ACID)?

Atomicity: Entire transaction takes place at once or doesn't happen at all (Abort/commit).

Consistency: Integrity constraints must be maintained so that the database is consistent before and after

the transaction.

Isolation: Transactions occur independently without interference.

Durability: Changes persist even if a system failure occurs.

17- What is the N+1 problem?

If we use an ORM function incorrectly, we might request a query with as many as the rows returned (N) from our query, instead of a single SQL query. For example:

With ORM:

```
SELECT * FROM CARS;
```

```
SELECT * FROM WHEELS WHERE CarId = ?;
```

With SQL command:

```
SELECT * FROM WHEELS;
```

18-The advantages of the Spring framework?

Lightweight(uses POJOs), Flexible(there are libraries), Powerful abstraction(JPA, etc.), Declarative support(e.g transactions), and Loose Coupling(dependency injection).

19- What is Dependency Injection? (Setter, field & constructor injection)

Dependency injection is a programming technique that makes a class independent of its dependencies. It achieves that by decoupling the usage of an object from its creation.

Constructor Injection: Setting private object inside the constructor. Used for mandatory dependencies.

Setter Injection: Calling the setter method on the bean after it has been instantiated with a no-arg constructor. Optional dependencies.

Field Injections: Uses Reflection. Implemented by adding @Autowired on a field. Avoided because it is costlier.

20- An interface has 2 implementations. How does the Spring decide which to implement?

1. way: Using @Qualifier. Shows which one to inject.

2. way: Using @Primary. Annotated the primary implementation.

2. way: Using @Primary. Annotates the primary implementation.

3. way: Make variables name with the implementations name.

Importance order @Qualifier> @Primary > Naming .

21-What is the Application context?

It is Spring's IoC container and is responsible for instantiating, configuring(configuration metadata), and assembling the beans.

22- What are the Spring bean scopes?

Bean scopes define the life cycle and visibility of that bean. There is 6 type of scopes:

- singleton
- prototype
- request
- session
- application
- WebSocket

23- What is the inversion of control (IoC)?

Delegation of control of dependencies from the developer to the framework.

24- What is the spring annotation for transaction management?

@Transactional: Default is "required". There are 7 types of transactions.

25- What are the transaction propagations?

REQUIRED: If there is an active transaction, it is connected to it. Otherwise, it will open new.

SUPPORTS: Checks if there is an active transaction. If there is, it will continue, otherwise, it will continue as a non-transaction.

MANDATORY: Checks if there is an active transaction. If there is, it continues, otherwise, it gives an error.

REQUIRES_NEW: Checks if there is an active transaction. If it exists, it suspends it and creates a new transaction.

NOT_SUPPORTED: Checks if there is an active transaction. If it does, it suspends it and executes non-transactional.

NEVER: Spring throws an error if there is an active transaction.

NESTED: Initiates a nested transaction that belongs to the current transaction. This puts save points between nested calls. That is, the inner transactions can be rolled back independently of the outer ones. JPA dialect, Hibernate does not support this. (NestedTransactionNotSupportedException: JpaDialect does not support savepoints — check your JPA provider's capabilities)

26_ Difference between Equals and hashCode?

Equals are used to compare two objects while hashCode is used in hashing to decide which group an object should be categorized into.

HashCode() returns the hashcode value of an object on calling. A 4-byte integer is used to access quickly the objects or attributes.

27_ What is Kafka?

Kafka stores, reads, and analysis a stream of data. The producer creates messages and the consumer receives messages.

28_ What is Docker?

Docker is a containerization platform. It has a Client-server architecture, and the docker daemon does the building, running, and distribution. On hardware, there are containers running in threads, on top of our os. Applications are downloaded from the repository in containers and run.

29- What are the basic concepts of object-oriented programming?

Objects are tangible or intangible entities that have their own characteristics and exhibit various attitudes by being in a certain situation depending on their characteristics. Class is a set of objects of the same type. Methods are the concept that determines what an object can do. Inheritances are the practice of a class taking certain characteristics and attitudes from a higher class and applying different features and attitudes. Abstraction means that the properties of the object that are required for the application are coded, while the other properties are not coded. Polymorphism allows objects to be represented as different types besides their own type. Encapsulation means hiding information about the work and state of the object and its functional complexity.

30- What is a framework?

The project can be built according to that core structure. The difference between a library and a framework is, that your code calls libraries but frameworks call your code.

31 — What is a compiler?

It is an application that translates source codes in any programming language into machine language.

32 — What is HTTP?

Answer: It is the system that regulates the rules and methods of how information is transferred between servers and end-users on the Internet.

33 — What is Authentication?

Answer: It is to check the user's identity. It is used to log into the system.

34 — What is Authorization?

Answer: To verify the user's access to resources. It is used to operate the system.

35 — What does MVC mean?

Answer: Model is short for View and Controller.

Model; It is the part that represents the data.

View; It is the interface that the user sees.

controller; It is the section where everything in the project such as database operations and calculations is controlled.

36 — What is API?

Answer: It is a module created for any application to use certain functions in other applications.

38 — What is REST?

Answer: It is a simple way of exchanging data between client and server. Data can be exchanged in formats such as JSON, XML, and Text. It is sent via HTTP. It works faster than SOAP.

39 — What is Middleware?

Answer: It is the intermediate layer. For example, we are designing a Rest API. We handle authentication with tokens. We've written a few APIs that require tokens. We can create a Middleware to check the validity of the Token in requests to these APIs. Thus, if the incoming request does not pass the control in the Middleware, it is directly interrupted. If he passes, he continues on his way.

40 —What is SSL?

Answer: It is sending encrypted data between the server and the client.

41 — What is static{ }?

Used to create fields that are created as the code compiled. Preferred when we want no nullity from the beginning for a variable.

42 — What are Coupling and Cohesion?

Coupling is the interdependence of classes. It is recommended to minimize Coupling for good software. Cohesion; How much is the purpose of creating a class?

43 — What are the concepts of Index, Trigger, and Stored Procedure in SQL?

Answer: Index; It enables the related table to be processed in order according to the data in the column used as the indexing field. It is used when queries are answered late. trigger; It means to trigger. It is used to automatically perform another operation when a transaction is made in the database. Stored Procedure; It is a piece of code that does a certain job. It does not need to be compiled again after the first compilation.

44 — What does Deadlock mean in SQL?

Answer: Two separate transactions are sent to the database, but both will do their job depending on the result of the other transaction. This deadly deadlock situation must be detected in advance and given priority to one.

45_ What is a microservice architecture?

Application is developed as a collection of services. Every part of the application runs by itself. This creates freedom for technology and development but creates complexity. The complexity comes from the many possibilities for the error scenarios. Let's an application that has 5 services. There are 2^5 different scenarios for working/not working. Should customers see products when comments are down or payments are down etc?

46_What is Agile?

Agile is an iterative approach to software development that helps teams create working products faster for their customers.

47 -Why is Java popular?

Any code written with Java can run on virtually any computing platform.

The code is robust enough to independently run Java programs without any external dependencies.

Java has experienced consistent development.

It's memory management and multithreaded structure.

48 -What is the Difference Between ArrayList and HashSet in Java? (Possible versions with List, Set, Array farkları, Stack, Queue)

ArrayList has an order of items by the insertion order. On the other hand, HashSet has no order and they are indexed by their hash encode.

49_What is the difference between ArrayList and Array?

ArrayList has dynamic memory, the array has a fixed number of elements. ArrayList has more predefined methods than an array.

Arrays can store primitives and objects but ArrayList can only store objects.

You can use an iterator for ArrayLists.

50_ What is the difference between ArrayList and HashList?

ArrayList allows duplicate values while HashSet doesn't allow duplicated values. HashList does not have an order, ArrayList has an order.

51 — HashSet vs TreeSet?

TreeSet is ordered but HashSet is not, hence HashSet is faster.

52 — What is a Java ClassLoader?

Class loaders are responsible for loading Java classes dynamically to the JVM (Java Virtual Machine) during runtime.

53 — Explain the main difference between fail-fast and fail-safe iterators.

A fail-safe iterator doesn't throw any Exception, contrary to a fail-fast Iterator. This is because they work on a clone of Collection instead of the original collection and that's why they are called the fail-safe iterator.

54 — Compare the wait () and sleep () methods in Java.

The major difference is that wait() releases the lock or monitor while sleep() doesn't release the lock or monitor while waiting. wait() is used for inter-thread communication while sleep() is used to introduce a

pause on execution.

55 — What is the static keyword?

The static class is the class inside a class. Static methods and static fields are stored in memory instead of the heap and they belong to the class they are in. They are created at the compile time. They are not accessed by objects but instead directly. Static structures only access other static structures. Static methods are generally preferred for utility classes (e.g Math class).

56 — How can you generate random numbers in Java?

1_Using java. util.Random;

```
Random random = new Random();  
random.nextInt(max); //0 - max random int
```

2_Using Math.random();

```
(int)Math.floor(Math.random()*(max-min+1)+min) //min - max
```

57 — When to use the volatile variable in Java?

when you need to instruct the JVM that a variable can be modified by multiple threads and give hint to JVM that does not cache its value.

58— When to use a transient variable in Java?

when you want to make a variable non-serializable in a class, which implements the Serializable interface. In other words, you can use it for a variable whose value you don't want to save.

59 — Difference between Serializable and Externalizable in Java?

Externalizable gives you more control over the Serialization process.

60— Difference between ArrayList and Vector in Java

Many, but most important `ArrayList` are non-synchronized and fast while `Vector` is synchronized and slow. It's also legacy class like `Hashtable`.

61— Difference between PATH and Classpath in Java?

`PATH` is used by the operating system while `Classpath` is used by JVM to locate Java binary, e.g. JAR files or Class files.

62 — Can we override the static method in Java?

No, because overriding resolves at runtime while a static method call is resolved at compile time.

63 — Difference between Iterator and Enumeration in Java?

Iterator also gives you the ability to remove an element while iterating while Enumeration doesn't allow that.

64— What is the race condition and deadlock in Java?

A race condition occurs when multiple concurrently executing processes access a shared data item and the result of execution depends on the order in which execution takes place.

Deadlock means coming to a point where no progress can be made because of fundamental disagreement.

65 — What are the basic principles of the OOPs concept

Abstraction, Polymorphism, Encapsulation, Inheritance.

66 — StringBuilder vs StringBuffer?

StringBuffer is synchronized and preferred for concurrent programs. On the other hand, for single-thread applications, StringBuilder is faster.

67 — What are the different ways to create threads in java?

There are two ways to create threads in java. a. Implement a Runnable interface. By extending the Thread class

68 — What is synchronization in java?

Synchronization is a technique to control access to a method with multiple threads at the same time. If we declare a method synchronized, then only one thread can use this method at a time. This is basically used for Thread safety.

69 — Shallow Copy vs Deep Copy?

In shallow copy, only fields of the primitive data type are copied while the object references are not copied. Deep copy involves the copy of primitive data types as well as object references.

70 — What is Spring IoC Container?

The Core of the Spring Framework creates the objects, configures and assembles their dependencies, and manages their entire life cycle. The Container uses Dependency Injection(DI) to manage the components that make up the application.

71 —How does configuration metadata provided to the Spring container?

Either XML-based, annotation-based, or java based configurations exist. The most popular way is annotation-based.

72 — Autowiring in Spring?

Used for searching a matching bean definition in the configuration file.

73 — What are Thread, Multithread, Mutex, and Semaphore?

Threads are independent computation sequences. Multithread applications compute different operations at the same time to gain time but are complex to successfully implement.

In multithreading, a mutex is an object used for locking sources whereas a semaphore is an integer used for signaling.

Semaphore supports wait and signal operations modification, whereas Mutex is only modified by the process that may request or release a resource.

Semaphore value is modified using wait () and signal () operations, on the other hand, Mutex operations are locked or unlocked.

74 — What is big O notation?

Show the worst-case complexity of an algorithm (learn to find the worst-time complexity of an algorithm).

75--What is the advantage and disadvantage of the normalization of a database?

The Advantage of normalization: Normalization reduces data duplication, groups data logically, and enforces referential integrity.

The disadvantage of normalization: Normalization slows database performance, complex to successfully normalize a database.

76- What is the difference between an abstract class and an interface?

Explaining abstract classes vs interfaces with example

77-How to compare two objects' equality?

All classes extend object class and object class has multiple methods such as equals(), hashCode(), to String(), etc. To compare two objects' equality we override the equals() method and define the compare rules.

78- Why encapsulation is done?

It prevents outer classes from accessing and changing fields and methods of a class and also helps to create setter and getter rules on data fields.

79-What is the purpose of “having” and “group by” in SQL?

“Group by” organizes similar data into groups another way of saying it gets summary data. The GROUP BY statement is often used with aggregate functions (COUNT(), MAX(), MIN(), SUM(), AVG()) to group the result set by one or more columns.

```
SELECT Shippers.ShipperName, COUNT(Orders.OrderID) AS NumberOfOrders FROM Orders
LEFT JOIN Shippers ON Orders.ShipperID = Shippers.ShipperID
GROUP BY ShipperName;
```

“Having” exists because “where” cannot be used on aggregations. A HAVING clause in SQL specifies that a “select” statement must only return rows where aggregate values meet the specified conditions.

```
SELECT Employees.LastName, COUNT(Orders.OrderID) AS NumberOfOrders
FROM (Orders
INNER JOIN Employees ON Orders.EmployeeID = Employees.EmployeeID)
GROUP BY LastName
HAVING COUNT(Orders.OrderID) > 10;
```

80-What happens when you try to put an already existing key element in the HashMap?

Does not give an error, just updates that key's value.

