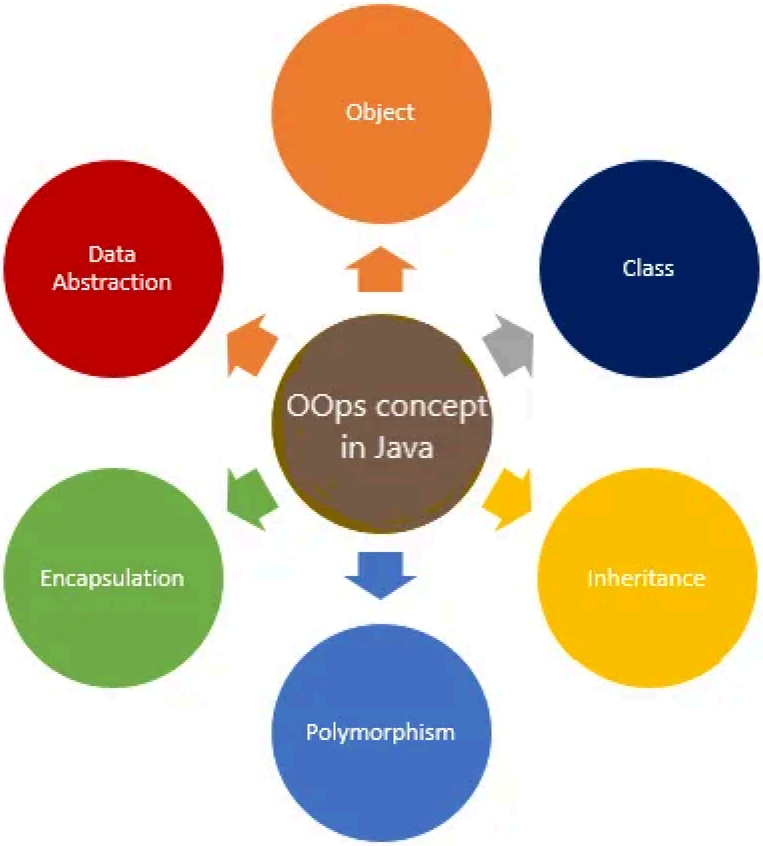
Java OOPs Interview Questions and Answers

Akshata Kanaje Follow 8 min read · Apr 4, 2024



1. **What is OOPs?**

Ans: OOPs stands for Object-Oriented Programming Systems. It's a type of programming that is based on objects and classes rather than functions and stored procedures.

- Object is an instance of a class consisting of properties that makes data useful whereas classes are a representation of an object describing its details and used to organize the data
- OOPs main aim is to implement real-world entities like inheritance, hiding, polymorphism etc. into programming.

2. Why do we need OOPs?

Ans:There are various reasons for defining the need for OOPs, but the important ones are:-

- OOPs make the development and maintenance of projects easier.
- OOPs provide the feature of data hiding that is good for security concern.
- We can provide the solution to real-world problems if we are using OOPs.
- Code can be reused through inheritance thereby reducing redundancy.
- With the help of encapsulation data and code can be bound together thus increasing security.
- The concept of Polymorphism gives flexibility to the program by allowing the entities to have multiple forms.

3. What are the features of OOPs?

Ans: The OOPs comprises of four main features that are:

- **Inheritance —** Inheritance is the concept in which one class inherits the property of the other class.
- **Encapsulation —** It wraps up the data and the code working together in a single unit.
- **Polymorphism —** It's the ability to exist in multiple forms. It lets users perform a single task in various ways.
- **Data Abstraction —** It helps hide the important information from the users and shows only necessary details.

4. What are the limitations of OOPs?

- Not suitable for solving small problems.
- Without the proper class documentation, it is hard to understand the code.
- Requires a lot of time to solve problems.
- Required intensive testing.
- The programmer should think of solving a problems in terms of objects.

5. State the differences between OOPs and Structural Programming?

- OOPs are based on objects and classes rather than functions and stored procedures where as Structural Programming is divided into functions and provides a logical structure to the program.
- A OOPs is more secure as data hiding is available where as a Structural Programming less secure as data hiding is not available.

6. What is class in java?

Ans: A class in Java is a blueprint from which objects are created. It defines the structure and behavior of objects that belong to the same type. Classes consist of fields (variables) and methods (functions). Fields represent the state or attributes of the object, while methods represent the actions that the object can perform.

Syntax:-

class < class_name > {

field;

method;

}

Example of a simple class:

```
public class Dog {
    // Fields (attributes)
    private String name;
    private int age;

    // Constructor
    public Dog(String name, int age) {
        this.name = name;
        this.age = age;
    }
}
```

Note: The class name should begin with the initial letter capitalized by convention.

7. What is object in java?

Ans: An object is an instance of a class, created using the blueprint provided by the class. Objects have their own state (attributes) and behavior (methods), which are defined in the class. You can create multiple objects from the same class, each having its own set of data.

Creating an object (instance) of the `Dog` class:

```
Dog myDog = new Dog("Max", 3);
```

8. What is data abstraction?

- Basically in program we don’t really want to know what exactly a function is doing, so we can achieve this with the help of abstraction.
- So we can give the user just the function that they need to execute and they don’t need to worry about how it is implemented and which are the different packages which are the different classes that are used inside of the specific function, with the help of this we can achieve abstraction.

Eg: car, so in case of car only relevant parts are shown like break, clutch, steering, horn etc. because they are necessary for driving. But the driver, he don’t need to know how there are functioning internally.

Thus showing relevant data to the user and hiding implementation or details from the user is called abstraction.

Advantage:

- Avoid code duplication & increase reusability.
- Can change internal implementation of class independently.

9. How do you achieve abstraction in java?

Ans: There are 2 ways for achieving abstraction. We can achieve abstraction using abstract class or We can achieve abstraction using interfaces.

10. Differentiate between an abstract class and an interface?

- Abstract keyword is used to declare abstract class, Interface keyword is used to declare interface class.
- Abstract class does not supports multiple inheritance, Interface supports multiple inheritance.
- Abstract class consists of both abstract and non abstract methods, Interface consists of both static, default and abstract methods.

11. What is an Abstract Method?

- A method declared without a body (no implementation) within an abstract class is know as abstract method.
- If a class is using an abstract method, the class must be declared abstract. But the opposite cannot be true which means abstract class doesn't

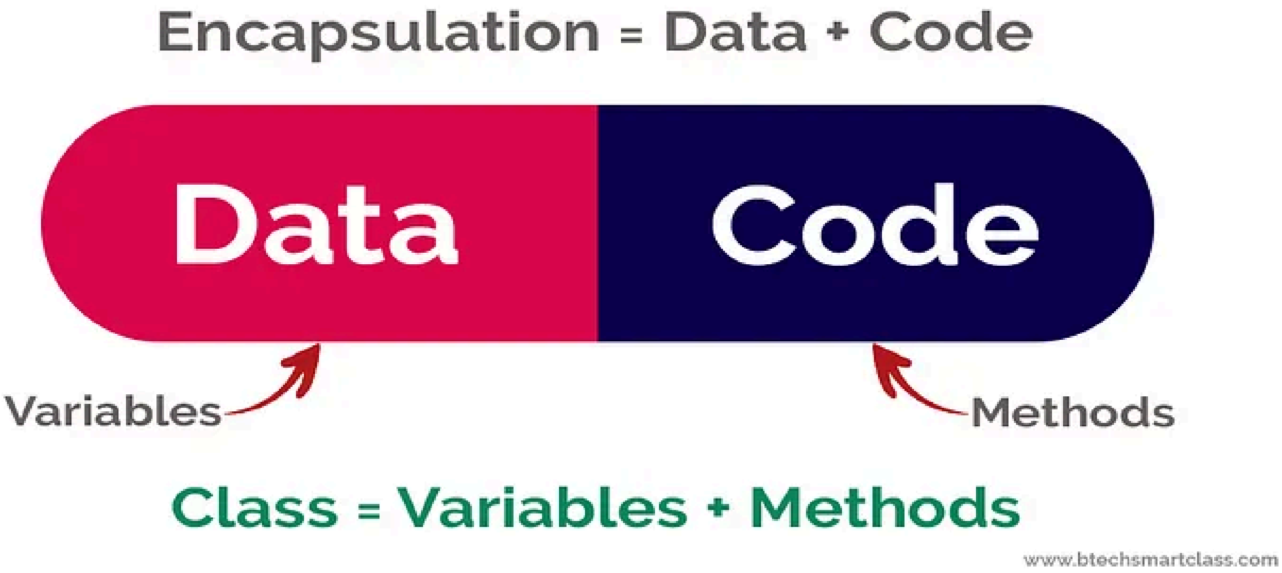
necessarily have an abstract method (which means abstract class may contain abstract method or it may contain concrete method).

12. What is encapsulation in java?

Ans: The process of grouping/binding data members(variables) and corresponding methods into a single unit is called Encapsulation. If any component follows **data hiding** and **abstraction**, then the component is said to be **encapsulated**. The encapsulation principle says hiding the data behind a method. Every Java class is an example of encapsulation.

i) Data Hiding: An outside person can not access out internal data directly, this concept of OOPS is called data hiding. After successful validation/authentication, an outside person can access our internal data. The main advantage is Security.

ii) Abstraction: Hiding Internal implementation, just highlighting the set of services that we are going to offer is the concept of abstraction. By using the *interface* and *abstract classes* we can implement abstraction.



Advantage of Encapsulation: Security, Enhancement becomes easier, maintainability, modularity etc.

Disadvantage of Encapsulation: It increases the line of code due to getter, setter method, validation logic etc, and hence it becomes time-consuming, performance becomes down.

13. What are the differences between Abstraction and Encapsulation?

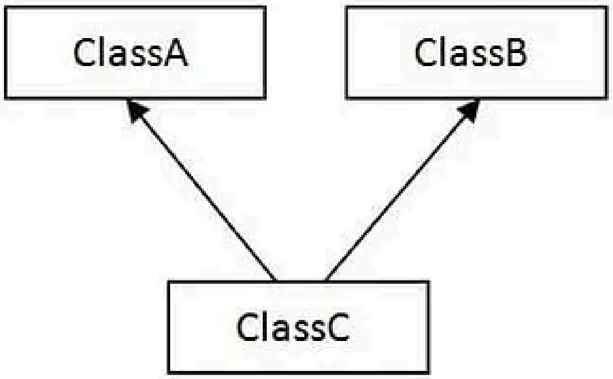
- Abstraction is used to hide the unwanted information or hiding the implementation from the user, Encapsulation is used hide the data in a single entity along with a method to protect information from outside.
- We can achieve the abstraction or we can hide the implementation by using abstract classes and interfaces, While in Encapsulation, the data is hidden by making the variable as private and by providing the getters setters method.

14. What is inheritance in java?

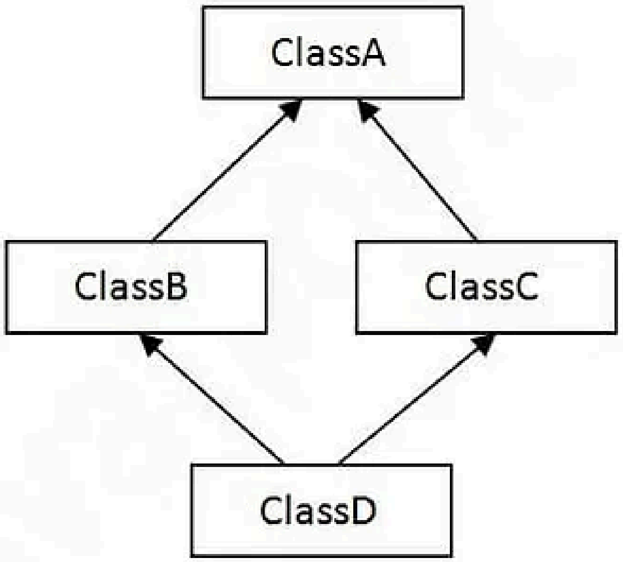
Ans: Inheritance is also known as the **IS-A relationship**, says all the methods and variables available in the parent class are available in child class but not vice-versa. By using **extends** keyword we can implement inheritance. Example: In Java, **Object** Class is the parent class of all other classes.

There are 5 types of inheritance concepts available.

1. **Single Inheritance**: Single child class extends single parent class.
2. **Multi-level Inheritance**: Single child class extend another child class which extends a parent class.
3. **Hierarchical Inheritance:** Multiple child class extends single parent.
4. **Hybrid Inheritance:** Combination of above 4 inheritance.
5. **Multiple Inheritance**: Single child class extends multiple parent class. This is not supported by Java because if two class contains two methods with the same name, then when the child class extends both parent class following multiple in-inheritance an ambiguity arises which class method to pick. Hence, Java doesn’t support multiple inheritances and hybrid inheritance in class. This problem is also called as **diamond access problem**. However, in the case of interface, multiple inheritance is supported in java because in the case of inheritance each class just contains the declaration and the corresponding implemented class only implements the method, hence there is no chance of ambiguity.



4) Multiple



5) Hybrid

.

Advantage of Inheritance: The main advantage is code re-usability. Always write the common methods for all other child classes in the parent class. Inside child class only write those methods which are specific to that child class.

Limitations of Inheritance: The child and parent class are linked tightly. In case any change in the program is required we need to change parent and child classes. It increases execution time and implementation time.

15. What is Polymorphism?

Polymorphism is considered one of the important features of Object-Oriented Programming and polymorphism is nothing but one function is used to perform the multiple tasks.

16. What are the types of polymorphism? Define them?

Ans: There are two different types of Polymorphism

- Static or (Compile Time) Polymorphism
- Dynamic Binding or (Run Time) Polymorphism

Static or (Compile Time) Polymorphism:-

Static Polymorphism occurs at the compile-time and is achieved through method overloading. i.e. Compiler decides what value has to be taken by the body in the picture. There are certain conditions for static polymorphism.

- The parameters numbers must be different from other parameters.
- The type of parameter should be different.

Due to different parameters, each method has a different signature.

Dynamic Binding or (Run-Time) Polymorphism:-

Run time Polymorphism as the name suggests occurs at run time. Method overriding is an example of dynamic polymorphism. There are certain conditions for static polymorphism.

- With the help of pointers and virtual functions, we can achieve method overriding.
- When a derived class has the same functions as that of the base class, that base class method is overridden.

17. What is method overloading in java?

Method Overloading is a concept in which two or more methods can have the same name with different arguments(signatures). Overloading is related to compile-time polymorphism and can be achieved in three ways…

- Changing the number of parameters.
- Changing the data type of parameters.
- Changing the sequence of the data type of parameters.

18. What is method overriding in Java?

Method Overriding is a concept of OOPs. It means if a subclass and superclass have the same arguments, then the argument of the subclass overrides the argument/method of the superclass. Method Overriding is related to Run-Time Polymorphism. The method overriding can be done when

- The arguments of both parent and child are the same.
- private, static, and final methods cant be overridden.
- The abstract method of superclass should always be overridden.

OOPs in java provides the ability to simulate real-world event much more effectively using the concept of **encapsulation** which gives the data security, **inheritance** which makes the code reusable followed by easy maintainability and **polymorphism** which gives us more flexibility in order to design and write the program.

Objects Object Oriented Programming Java Oops Concepts

Written by Akshata Kanaje

9 followers · 17 following

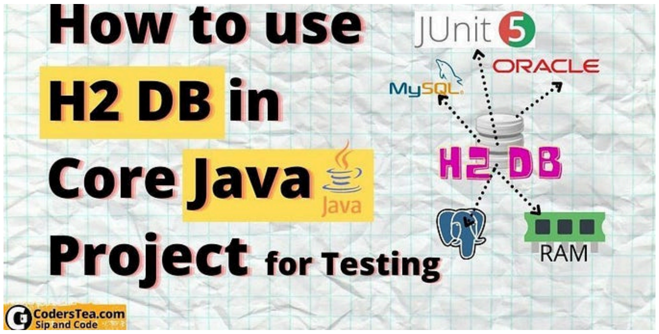
Follow

No responses yet

Hlgsagar

What are your thoughts?

More from Akshata Kanaje



Akshata Kanaje

Setting Up H2 for Testing in Spring Boot application

In this article, I will show you how to use H2 for integration testing, using different applicatio...

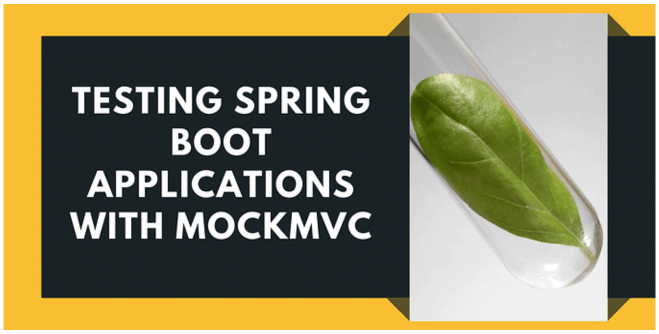
Apr 16, 2024 👏 10

Akshata Kanaje

Implement basic authentication in a Spring Boot application

In this article, we will explore the implementation of basic authentication in a...

May 23, 2024



Akshata Kanaje

Testing REST Endpoints Using MockMvc

In this article, we will learn how to write JUnit test cases for Spring boot REST APIs. We wil...

Feb 25

Akshata Kanaje

Let's Get Some Small Overview of Collection Framework in Java

Collection Framework in Java is a set of interfaces and classes which can use to stor...

Apr 12, 2024

See all from Akshata Kanaje

Recommended from Medium

Pudari Madhavi

Cracking the Barclays Java Developer Interview: A Step-by-...

If you are not a Member — Read for free here :

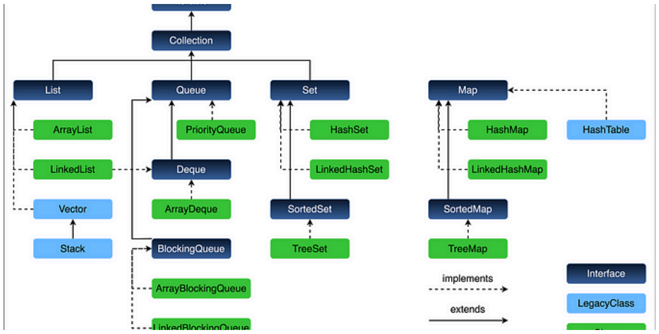
Jun 2 10

Mohamed Enn

Interview Questions: Java Generics

Generics are a powerful feature in Java that enable type-safe collections and APIs. They'...

Mar 13 73

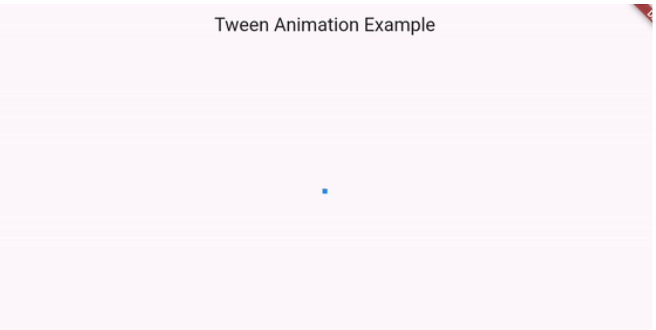


In Towards Dev by Sudhindra Kr. Saxena

A Comprehensive Guide to Java Collection Classes

Java's Collection Framework provides a well-structured hierarchy of interfaces and classe...

Jan 14 1



Tween Animation Example

Vaishanvi Ingole

Interview Questions

Extension method

Jan 13

Duplicates	Order
Yes	Maintains
No	Optional
Key-value	Unique keys

Java Interview

Publicis Sapient Backend Java Developer Interview Questions &…

Here’s a curated list of commonly asked Publicis Sapient Java Developer Interview…

3d ago

In Coding Odyssey by Shivam Srivastava

JP Morgan Java Developer Interview

Java Lead Interview Experience

Jan 24 538 19

See more recommendations