

Java Keywords

Key Concepts

Key Concepts #2

OOPS in java

Java Collections#1

Java Collections #2

Exceptions #1

Exceptions #2

Threads#1

Threads#2

InnerClass

More InnerClass

Serialization

Immutable Class

Cloning in Java

Garbage Collection

JSP #1

JSP #2

Programming Q#1

Programming Q#2

Programming Q#3

# Java Innerclass Interview Questions

Q11) How to create an inner class instance from outside the outer class instance code?

Ans) To create an instance of the inner class you must have the instance of its enclosing class.

```
class EnclosingOuter {
    class Inner{ }
}
```

To create the instance of inner class from class other than the enclosing class.

```
1) class OtherThanOuter{
    EnclosingOuter out = new EnclosingOuter();
    EnclosingOuter.Inner in = out.new Inner();
}

2) class OtherThanOuter{
    EnclosingOuter.Inner out = new EnclosingOuter.Inner ();
}
```

Q13) Which modifiers can be applied to the inner class?

Ans) Following are modifiers that can be applied to the inner:

- public
- private
- abstract
- final
- protected
- strictfp
- static – turns the inner class into
- static nested class.

Q14) Can the method local inner class object access method's local variables?

Ans) No, a method local inner class object can not access the method local variable.

**Reason:** The local variables are not guaranteed to live as long as the local inner class object. The method local variable live on stack and exist only till the method lives, their scope is limited only code inside the method they are declared in. But the local inner class object created within the method lives on heap and it may exist even after the method ends if in case the reference of this local inner class is passed into some other code and is stored in an instance variable. So we can not be sure that the local variables will live till the method local inner class object lives, therefore the method local inner class object can not access the method local variable. To access the method local variables, the variable has to be declared as final.



Q15) Can a method local inner class access the local final variables?Why?

Ans) Yes. Because the final variables are stored on heap and live as long as the method local inner class object may live.

Q16) Which modifiers can be applied to the method local inner class?

Ans) Only abstract or final keyword isallowed.

Q17) Can a local class declared inside a static method have access to the instance members of the outer class?

Ans) No. There is no this reference available in the static method .The static method class can not have access to any members of the outer class other than static members.

Q18) Can a method which is not in the definition of the superclass of an anonymous class be invoked on that anonymous class reference?

Ans) No. Compilation will fail.As the reference variable type of the anonymous class will be of superclass which will not know of any method defined inside the anonymous class the compilation will fail.

```
class SuperClass{
    void doSomething() {
        System.out.println("In the Super class");
    }
}
class hasAnonymous{
    SuperClass anon = new SuperClass(){
        void doSomething() {
            System.out.println("In the Anonymous class");
        }

        void doStuff() {
            System.out.println("An Anonymous class method not present in
superClass");
        }
    };

    public void doIt(){
        anon.doSomething(); // legal superClass has this method
        anon.doStuff(); // Not legal }
}
```

The above code does not compile as the superClass does not know about the anonymous class method doStuff().

Q19) Can an anonymous class define method of its own?

Ans) Yes. But there will be no way by which the methods defined in the anonymous class which are not present in its superclass be invoked. As only those methods which are defined in the suprclass which the anonymous class extends be invoked defining the methods in the anonymous class will be of no use.

Q20) Can an anonymous class implement an interface and also extend a class at the same time?

Ans) No. An anonymous class can either extend a class or implement a single interface. If the anonymous class is extending a class then it becomes the implementer of all the interfaces implemented by its superclass automatically.

0 Comments

www.java-questions.com

1 Login

Recommend 1

Share

Sort by Best

Start the discussion...

LOG IN WITH

OR SIGN UP WITH DISQUS

Name

Be the first to comment.

ideapad 320S

Shine with the one that shines!

Metallic finish

Ideapad 320S  
Intel® Core™ i5 processor  
8th Gen Intel® Core™ processors:  
Designed for what's coming next

KNOW MORE

Lenovo

jscrambler

CLIENT-SIDE  
THE NEW BATTLEFRONT

From code to webpage, Jscrambler  
tamper-proofs your application.  
START NOW!

Try it for Free

## Recommend Reading

- [How java manages Memory?](#)
- [Why is it advised to use hashCode and equals together?](#)
- [Comparable and Comparator in java](#)
- [How to create Singleton class in Java?](#)
- [Lambdas in Java8](#)
- [How Concurrent HashMap Works?](#)
- [Difference between final, finally and finalize](#)
- [Log Structured Merge Trees](#)

Subscribe to get latest updates.

Email Address \*

Subscribe