

Inner class in java

Inner class means one class which is a member of another class. There are basically four types of inner classes in java.

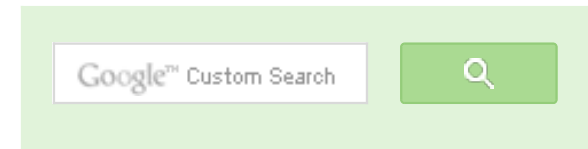
- 1) Nested Inner class
- 2) Method Local inner classes
- 3) Anonymous inner classes
- 4) Static nested classes

Nested Inner class can access any private instance variable of outer class. Like any other instance variable, we can have access modifier private, protected, public and default modifier. Like class, interface can also be nested and can have access specifiers.

Following example demonstrates a nested class.

```
class Outer {  
    // Simple nested inner class  
    class Inner {  
        public void show() {  
            System.out.println("In a nested class method");  
        }  
    }  
}  
  
class Main {  
    public static void main(String[] args) {  
        Outer.Inner in = new Outer().new Inner();  
        in.show();  
    }  
}
```

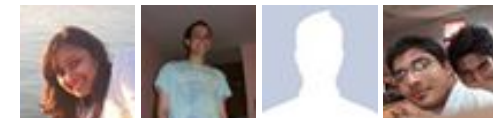
Output:



GeeksforGeeks



53,527 people like [GeeksforGeeks](#).



[Interview Experiences](#)

[Advanced Data Structures](#)

[Dynamic Programming](#)

[Greedy Algorithms](#)

[Backtracking](#)

[Pattern Searching](#)

[Divide & Conquer](#)

[Mathematical Algorithms](#)

[Recursion](#)

[Geometric Algorithms](#)

In a nested class method

As a side note, we can't have static method in a nested inner class because an inner class is implicitly associated with an object of its outer class so it cannot define any static method for itself. For example the following program doesn't compile.

```
class Outer {  
    void outerMethod() {  
        System.out.println("inside outerMethod");  
    }  
    class Inner {  
        public static void main(String[] args){  
            System.out.println("inside inner class Method");  
        }  
    }  
}
```

Output:

Error illegal static declaration in inner class
Outer.Inner public static void main(String[] args)
modifier 'static' is only allowed in constant
variable declaration

An interface can also be nested and nested interfaces have some interesting properties. We will be covering nested interfaces in the next post.

Method Local inner classes

Inner class can be declared within a method of an outer class. In the following example, Inner is an inner class in outerMethod().

```
class Outer {  
    void outerMethod() {  
        System.out.println("inside outerMethod");  
        // Inner class is local to outerMethod()  
        class Inner {  
            void innerMethod() {  
                System.out.println("inside innerMethod");  
            }  
        }  
        Inner y = new Inner();  
        y.innerMethod();  
    }  
}
```



Popular Posts

[All permutations of a given string](#)

[Memory Layout of C Programs](#)

[Understanding "extern" keyword in C](#)

[Median of two sorted arrays](#)

[Tree traversal without recursion and without stack!](#)

[Structure Member Alignment, Padding and Data Packing](#)

[Intersection point of two Linked Lists](#)

[Lowest Common Ancestor in a BST.](#)

[Check if a binary tree is BST or not](#)

[Sorted Linked List to Balanced BST](#)

```

    }
}
class MethodDemo {
    public static void main(String[] args) {
        Outer x = new Outer();
        x.outerMethod();
    }
}

```

Output

Inside outerMethod

Inside innerMethod

Method Local inner classes can't use local variable of outer method until that local variable is not declared as final. For example, the following code generates compiler error (Note that x is not final in outerMethod() and innerMethod() tries to access it)

```

class Outer {
    void outerMethod() {
        int x = 98;
        System.out.println("inside outerMethod");
        class Inner {
            void innerMethod() {
                System.out.println("x= "+x);
            }
        }
        Inner y = new Inner();
        y.innerMethod();
    }
}
class MethodLocalVariableDemo {
    public static void main(String[] args) {
        Outer x=new Outer();
        x.outerMethod();
    }
}

```

Output:

local variable x is accessed from within inner class;
needs to be declared final

Deploy Early. Deploy Often.

DevOps from
Rackspace:

Automation

[FIND OUT HOW ►](#)



But the following code compiles and runs fine (Note that x is final this time)

```
class Outer {
    void outerMethod() {
        final int x=98;
        System.out.println("inside outerMethod");
        class Inner {
            void innerMethod() {
                System.out.println("x = "+x);
            }
        }
        Inner y = new Inner();
        y.innerMethod();
    }
}

class MethodLocalVariableDemo {
    public static void main(String[] args){
        Outer x = new Outer();
        x.outerMethod();
    }
}
```

Output:-

```
Inside outerMethod
X = 98
```

The main reason we need to declare a local variable as a final is that local variable lives on stack till method is on the stack but there might be a case the object of inner class still lives on the heap.

Method local inner class can't be marked as private, protected, static and transient but can be marked as abstract and final, but not both at the same time.

Static nested classes

Static nested classes are not technically an inner class. They are like a static member of outer class.

```
class Outer {
    private static void outerMethod() {
        System.out.println("inside outerMethod");
    }
}
```

705



Subscribe

Recent Comments

Abhi You live US or India?

[Google \(Mountain View\) interview](#) · 35 minutes ago

[Aman](#) Hi, Why arent we checking for conditions...

[Write a C program to Delete a Tree.](#) · 1 hour ago

kzs please provide solution for the problem...

[Backtracking | Set 2 \(Rat in a Maze\)](#) · 1 hour ago

[Sanjay Agarwal](#) bool

tree::Root_to_leaf_path_given_sum(tree...

Root to leaf path sum equal to a given number · 1 hour ago

[GOPI GOPINATH @admin](#) Highlight this sentence "We can easily...

Count trailing zeroes in factorial of a number · 1 hour ago

[newCoder3006](#) If the array contains negative numbers also. We...

[Find subarray with given sum](#) · 2 hours ago

AdChoices

[▶ Java Array](#)

[▶ Java Void](#)

[▶ Class Class Java](#)

AdChoices

[▶ String Java](#)

```
// A static inner class
static class Inner {
    public static void main(String[] args) {
        System.out.println("inside inner class Method");
        outerMethod();
    }
}
```

```
}
```

Output

```
inside inner class Method
inside outerMethod
```

Anonymous inner classes

Anonymous inner classes are declared without any name at all. They are created in two ways.

a) As subclass of specified type

```
class Demo {
    void show() {
        System.out.println("i am in show method of super class");
    }
}
class Flavor1Demo {
```

```
// An anonymous class with Demo as base class
static Demo d = new Demo() {
    void show() {
        super.show();
        System.out.println("i am in Flavor1Demo class");
    }
};


public static void main(String[] args){
    d.show();
}
}
```

Output

```
i am in show method of super class
i am in Flavor1Demo class
```

[▶ Java Args](#)

[▶ Java Object](#)

AdChoices 

[▶ Java Args](#)

[▶ Java Object](#)

[▶ Java Public](#)

In the above code, we have two class Demo and Flavor1Demo. Here demo act as super class and anonymous class acts as a subclass, both classes have a method show(). In anonymous class show() method is overridden.

a) As implementer of the specified interface

```
class Flavor2Demo {  
  
    // An anonymous class that implements Hello interface  
    static Hello h = new Hello() {  
        public void show() {  
            System.out.println("i am in anonymous class");  
        }  
    };  
  
    public static void main(String[] args) {  
        h.show();  
    }  
}  
  
interface Hello {  
    void show();  
}
```

Output:

```
i am in anonymous class
```

In above code we create an object of anonymous inner class but this anonymous inner class is an implementer of the interface Hello. Any anonymous inner class can implement only one interface at one time. It can either extend a class or implement interface at a time.

This article is contributed by **Pawan Kumar**. Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above

HIGH-PERFORMANCE COMPUTING ON A UNIVERSITY BUDGET

Define your ideal server

Download
the infographic

SERVERSDIRECT

Related Topics:

- How to compare two arrays in Java?
- Can we override private methods in Java?
- Can we Overload or Override static methods in java ?
- Static class in Java
- Do we need forward declarations in Java?
- Checked vs Unchecked Exceptions in Java
- Bitwise right shift operators in Java
- Private and final methods in Java



12



Tweet

3



1

Writing code in comment? Please use ideone.com and share the link here.

3 Comments

GeeksforGeeks

Sort by Newest ▼

Divjot Singh • 7 days ago

please tell us how can we use inner classes in practical world and what are th
in same package.

^ | v • Reply • Share ›

abhi → Divjot Singh • 4 days ago

event listeners mostly use inner class

^ | v • Reply • Share ›

KDCRAZY → Divjot Singh • 6 days ago

I am not sure if you have coded android applications or not. but when v
some actions based on touch events then we call a function setOnlten
we actually create an instance of an Anonymous inner class.
and that inner class actually implements an interface so in the argume
interface is given.

for example :

```
listView.setOnItemClickListener(new OnItemClickListener() {
```

```
@Override
```

```
public void onItemClick(AdapterView arg0, View arg1,
```

```
int position, long id)
```

```
{
```

```
// do something here
```

```
}
```

Google it to know more.

2 ^ | v • Reply • Share ›

 [Subscribe](#)

 [Add Disqus to your site](#)

@geeksforgeeks, **Some rights reserved**

[Contact Us!](#)

Powered by **WordPress** & **MooTools**, customized by geeksforgeeks team