Lambda expression

Thursday, September 18, 2025 12:42 PM

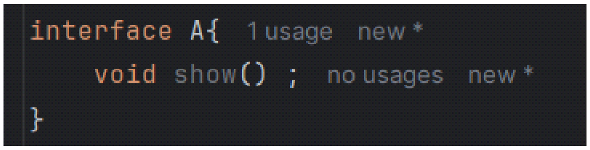
Definition of Lambda Expression in Java

A **lambda expression** in Java is a **short block of code** (like an anonymous function) that you can pass around as data.
It provides the implementation of a **functional interface's single abstract method** directly, without creating a separate class or object.
Introduced in **Java 8**.

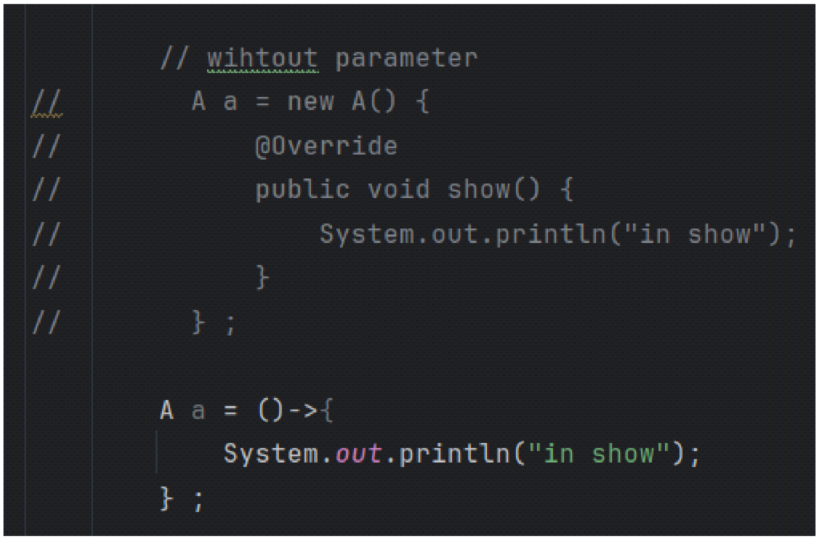
Types by Usage

1️⃣ Without Parameters

Lambda expression with no input arguments.



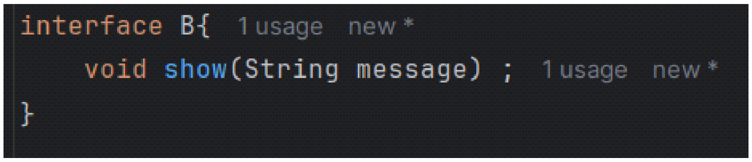
```
interface A{  1 usage  new *
    void show() ;  no usages  new *
}
```



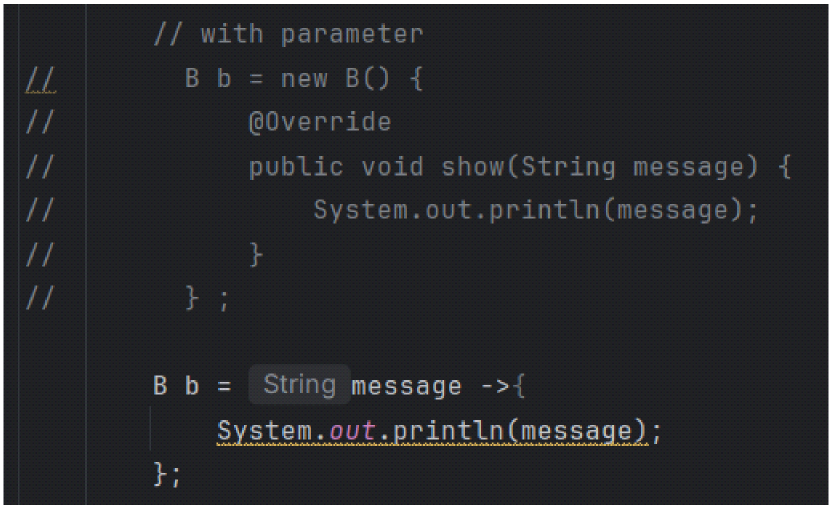
```
        // wihtout parameter
//        A a = new A() {
//            @Override
//            public void show() {
//                System.out.println("in show");
//            }
//        } ;

        A a = ()->{
            System.out.println("in show");
        } ;
```

2️⃣ With Parameters

Lambda expression that takes one or more arguments.



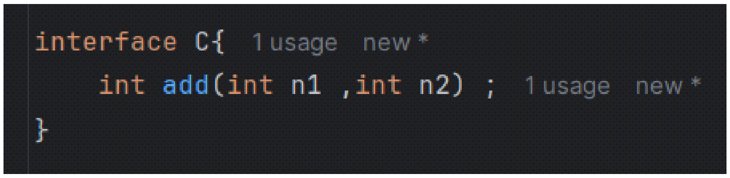
```
interface B{  1 usage  new *
    void show(String message) ;  1 usage  new *
}
```



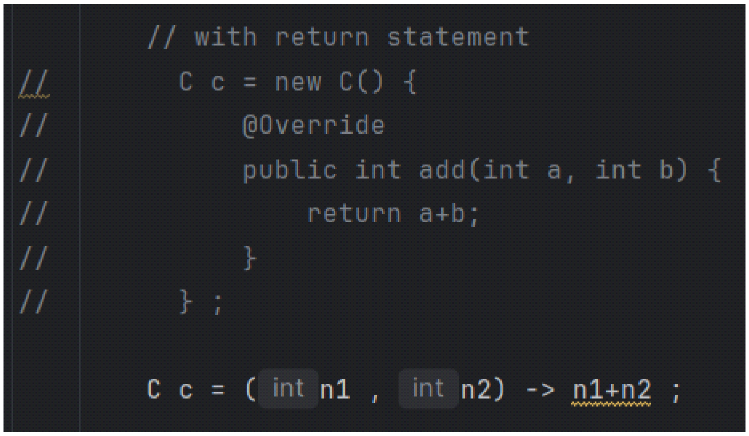
```
        // with parameter
//        B b = new B() {
//            @Override
//            public void show(String message) {
//                System.out.println(message);
//            }
//        } ;

        B b =  String message ->{
            System.out.println(message);
        };
```

3 With Return Statement

Lambda expression that returns a value (explicit return if multiple statements).



```
interface C{  1 usage  new *
    int add(int n1 ,int n2) ;  1 usage  new *
}
```



```
        // with return statement
//        C c = new C() {
//            @Override
//            public int add(int a, int b) {
//                return a+b;
//            }
//        } ;

        C c = ( int n1 ,  int n2) -> n1+n2 ;
```