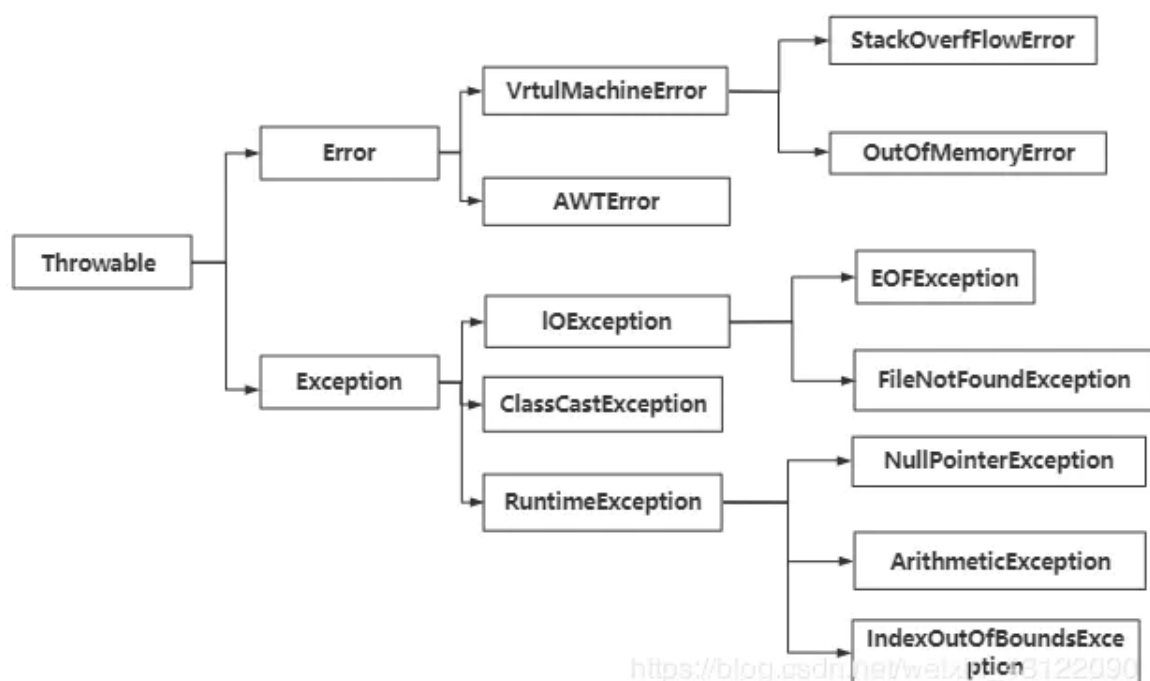


Java异常 与异常关 字

1. Java异常 介

- Java 异常 Java 以 中 代 业 代 , 代 优 , 下, what, where, why 3个 : 了“什么” , 了“ ” , 了“为什么”会

2. Java异常



1. Throwable

- Throwable Java 中与
- Throwable 两个：Error () Exception ()，们于了
- Throwable 了，了 printStackTrace() 于

2. Error ()

- **定义：**Error 中，中 了严
- ：代 JVM Virtual MachineError () NoClassDefFoundError (义) OutOfMemoryError: 不 ; StackOverflowError: , JVM
- 些 不，代，不 Java，们不任 Error !

3. Exception (异常)

- 以 且 以 Exception 为两：

时异常

- **定义：**RuntimeException，JVM
- ：Java 不会也，中，" throws "，也" try-catch "，会 NullPointerException ArrayIndexOutOfBoundsException 下 ClassCastException ArithmeticException 于不，

- **定义：**Exception

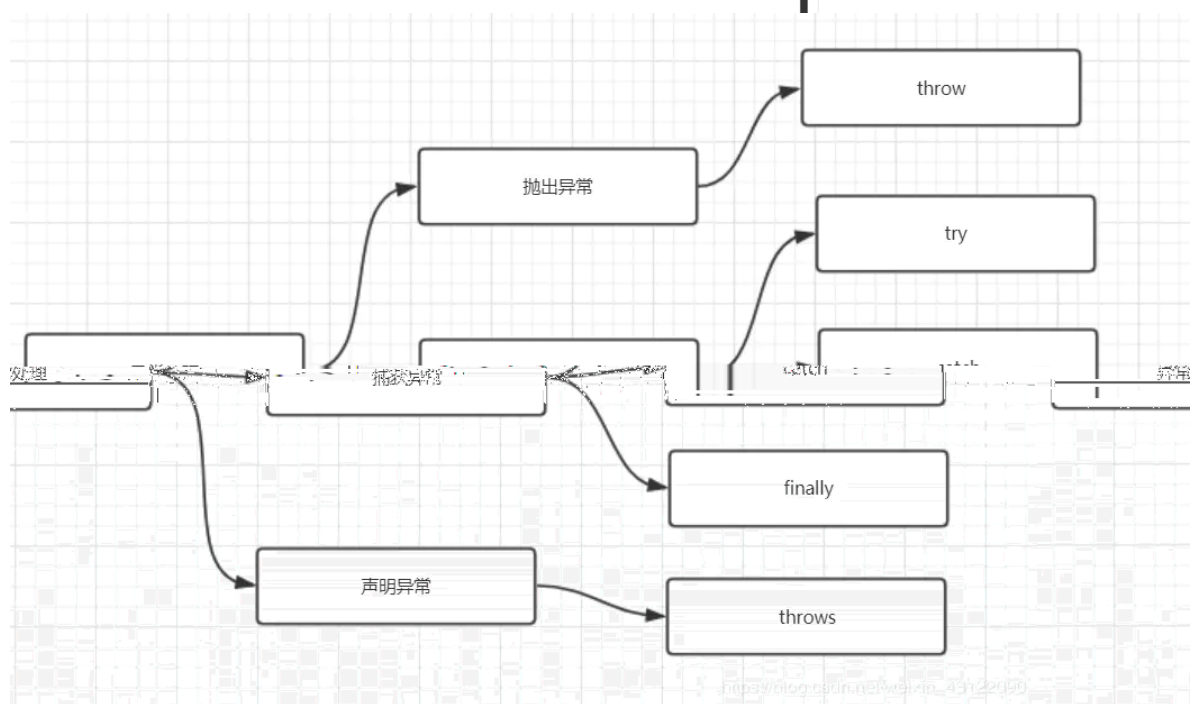
受 异常

- 不会 且不 , 也 中 , 我们
try-catch , 也 throws , 也会 异常包括 时异常
(RuntimeException 其子) 和 (Error)

3. Java异常关 字

- try - 于 代 (代) try 之 , try
,
- catch - 于 catch try 中
- finally - finally 会 主 于 try (中 return
件) finally , 之 , 会 try catch 中 return
throw , finally中 了return throw , 不会 ,
- throw - 于
- throws - 中, 于

3. Java异常处



- Java (Exception Handler) , 不 , 了
Java 中, 个 个 中 , Throwable 个 个
以 Java 5 个 : try catch throw throws
finally
- Java 中, 为 ,

1. 声明异常

- , 些 , 不 传 下 传 以
throws 会
意

- (Error RuntimeException 们) 不 throws

- 个 , try-catch/ throws | , 会

2. 抛出异常

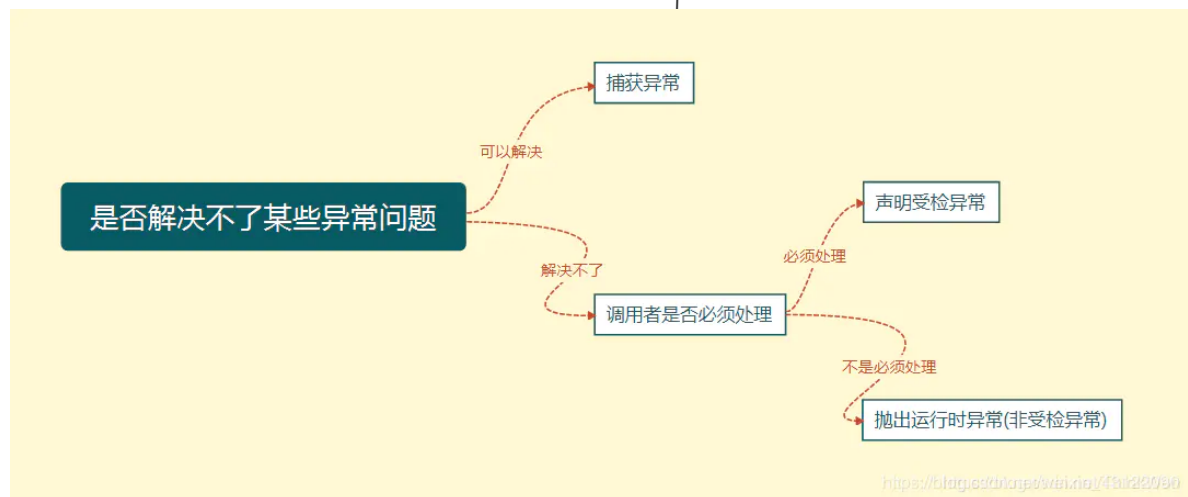
- 不了 些 , 且不 , 么 以
- throw 个 Throwable 任 Java代 以 throw

3. 捕 异常

- 之 不 , 但 会 些 之 , 但 不 上
- , 么 try...catch... 之 不

4. 如何 择异常 型

- 以 下 ,



5. 常 异常处 方式

接抛出异常

- , 些 , 不 传 下 传 以
- throws 会

```

private static void readFile(String filePath) throws IOException {
    File file = new File(filePath);
    String result;
    BufferedReader reader = new BufferedReader(new FileReader(file));
    while((result = reader.readLine())!=null) {
        System.out.println(result);
    }
    reader.close();
}

```

封 异常再抛出

- 们会从 catch 中 个 , 为了 于 , 个 , 以 , 不

```
private static void readFile(String filePath) throws MyException {
    try {
        // code
    } catch (IOException e) {
        MyException ex = new MyException("read file failed.");
        ex.initCause(e);
        throw ex;
    }
}
```

捕 异常

- 个 try-catch 中 以 个 , 不 不

```
private static void readFile(String filePath) {
    try {
        // code
    } catch (FileNotFoundException e) {
        // handle FileNotFoundException
    } catch (IOException e){
        // handle IOException
    }
}
```

- 个 catch 也 以 , |

```
private static void readFile(String filePath) {
    try {
        // code
    } catch (FileNotFoundException | UnknownHostException e) {
        // handle FileNotFoundException or UnknownHostException
    } catch (IOException e){
        // handle IOException
    }
}
```

定义异常

- 习 上, 义 个 两个 , 个 个
(Throwable toString 会 些 ,)

```
public class MyException extends Exception {
    public MyException(){ }
    public MyException(String msg){
        super(msg);
    }
    // ...
}
```

try-catch-finally

- 中，之代不会，之了些，
catch 中代，代，finally
以个

```
private static void readFile(String filePath) throws MyException {
    File file = new File(filePath);
    String result;
    BufferedReader reader = null;
    try {
        reader = new BufferedReader(new FileReader(file));
        while((result = reader.readLine())!=null) {
            System.out.println(result);
        }
    } catch (IOException e) {
        System.out.println("readFile method catch block.");
        MyException ex = new MyException("read file failed.");
        ex.initCause(e);
        throw ex;
    } finally {
        System.out.println("readFile method finally block.");
        if (null != reader) {
            try {
                reader.close();
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }
}
```

- ，件，代会 catch代，之 finally代；
件，会 catch代 finally代以代中
，finally中代会
- catch代中 return，finally中代会？以上代中 catch
下：

```
catch (IOException e) {
    System.out.println("readFile method catch block.");
    return;
}
```

- readFile，catch中return，finally

```
readFile method catch block.  
readFile method finally block.
```

- , catch 中 了 return , finally 会 finally 中也 return , finally 中 return 会 return.

try-with-resource

- 上 中, finally 中 close 也 IOException, 从 了 JAVA 7 了 优 , 了 AutoCloseable

```
private static void tryWithResourceTest(){  
    try (Scanner scanner = new Scanner(new FileInputStream("c:/abc"),"UTF-8")){  
        // code  
    } catch (IOException e){  
        // handle exception  
    }  
}
```

- try 代 , 会 scanner.close , scanner.close finally 代 中不 , scanner.close | , 会 , 仍 为 会 addSuppressed , 以 getSuppressed

Java异常常

1. Error 和 Exception 区别 什么?

- Error 为 , , 不 , , 不会 , JAVA 也不 , , 会 , 仅 ;
- Exception 以 中 , , , 以

2. 时异常和 异常(受 异常)区别 什么?

- RuntimeException , JVM Java 不会
- Exception 中 RuntimeException 之 Java 会
- RuntimeException异常和受 异常之 区别: , 么 , 们 RuntimeException (RuntimeException) ,

3. JVM 如何处 异常 ?

- 个 中 , 个 会 个 , 交 JVM, 交 JVM 为 , ,

- JVM 会以 代 为 JVM 代

4. throw 和 throws 区别 什么?

- Java 中 throws 了 上 throw
- throws 关 字和 throw 关 字在使 上 几 区别如下:
- throw 于 代 中
- throws 上, 以 个 中 代

5. final finally finalize 什么区别?

- final 以 不
- finally try-catch 代 中, 代 会 些 代
- finalize 个 , 于 Object 个 中 , Object , Java 中 finalize()

6. NoClassDefFoundError 和 ClassNotFoundException 区别?

- NoClassDefFoundError 个 Error , JVM , 不 个
- JVM ClassLoader 中 不 义, 了
- ClassNotFoundException 个 , try-catch Class.forName, ClassLoader.loadClass ClassLoader.findSystemClass 会 ; 个 个 中, 个

7. try-catch-finally 中哪个 分可以 ?

- : catch 以

原因

- 为严 : try , try+catch + 也 , try 不 以 catch , 不 , 为 , catch 以 上 catch 也
- 上, 任 代 不 , 以 代 上 try, 于 也 不 上 但 代 上

- 于上finally, 不

8. try-catch-finally 中, 如 catch 中 return 了, finally 会执吗?

- 意: finally 中
- 不会

代 例1:

```
public static int getInt() {
    int a = 10;
    try {
        System.out.println(a / 0);
        a = 20;
    } catch (ArithmeticException e) {
        a = 30;
        return a;
    } /*
        * return a 在程序执行到这一步的时候, 这里不是return a 而是 return 30; 这个返回
        路径就形成了
        * 但是呢, 它发现后面还有finally, 所以继续执行finally的内容, a=40
        * 再次回到以前的路径, 继续走return 30, 形成返回路径之后, 这里的a就不是a变量了, 而是
        常量30
        */
    } finally {
        a = 40;
    }
    return a;
}
```

- : 30

代 例2:

```
public static int getInt() {
    int a = 10;
    try {
        System.out.println(a / 0);
        a = 20;
    } catch (ArithmeticException e) {
        a = 30;
        return a;
    } finally {
        a = 40;
        //如果这样, 就又重新形成了一条返回路径, 由于只能通过1个return返回, 所以这里直接返回40
        return a;
    }
}
```

```
}
```

- : 40

9. ExampleA 继承 Exception, ExampleB 继承 ExampleA

- 下代 :

```
try {
    throw new ExampleB("b")
} catch (ExampleA e) {
    System.out.println("ExampleA");
} catch (Exception e) {
    System.out.println("Exception");
}
```

- 代 什么?
- : : ExampleA (代 [,
- ExampleA catch try 中 ExampleB)
- - 下代 (Java 书)

```
class Annoyance extends Exception {
}
class Sneeze extends Annoyance {
}
class Human {
    public static void main(String[] args)
        throws Exception {
        try {
            try {
                throw new Sneeze();
            } catch ( Annoyance a ) {
                System.out.println("Caught Annoyance");
                throw a;
            }
        } catch ( Sneeze s ) {
            System.out.println("Caught Sneeze");
            return ;
        } finally {
            System.out.println("Hello world!");
        }
    }
}
```

-

```
Caught Annoyance
Caught Sneeze
Hello world!
```

10. 常 RuntimeException 哪些?

- `ClassCastException`()
- `IndexOutOfBoundsException`()
- `NullPointerException`()
- `ArrayStoreException`(, 不)
- `IOException` `BufferOverflowException`

11. Java 常 见 的 异常 有 哪些

- `java.lang.IllegalAccessError`: 个 (Field) , 但 ,
- `java.lang.InstantiationError`: 个 Java new 个
- `java.lang.OutOfMemoryError`: 不 不 以 Java 个
- `java.lang.StackOverflowError`: 个
- `java.lang.ClassCastException`: A B (A 不 B) , O A , 么 O 为 B 为
- `java.lang.ClassNotFoundException`: 不 串 , CLASSPATH 之 不 class 件 ,
- `java.lang.ArithmeticException`: 件 :
- `java.lang.ArrayIndexOutOfBoundsException`: 为 于 于
- `java.lang.IndexOutOfBoundsException`: 个 于0 于 于
- `java.lang.InstantiationException`: newInstance() 个 , 个 ,
- `java.lang.NoSuchFieldException`: 不 个 不
- `java.lang.NoSuchMethodException`: 不 个 不
- `java.lang.NullPointerException`: 了null , : null null null throw null
- `java.lang.NumberFormatException`: 个String 为 , 串 不 ,
- `java.lang.StringIndexOutOfBoundsException`: 串 个 串中 , 于0 于 于 ,

Java异常处理 最佳实践

- Java 中 不 个 事 不 仅 仅 些 , 些 也 会 些 , 之 些 也 不
- 个

1. 在 finally 块中 或 使 try-with-resource 句

- 似InputStream , 个 try

```
public void doNotCloseResourceInTry() {
    FileInputStream inputStream = null;
    try {
        File file = new File("./tmp.txt");
        inputStream = new FileInputStream(file);
        // use the inputStream to read a file
        // do NOT do this
        inputStream.close();
    } catch (FileNotFoundException e) {
        log.error(e);
    } catch (IOException e) {
        log.error(e);
    }
}
```

- , 以 , 代 以 try代 代 会 , 且 以 但 , try代 , 个 个 , 且, 也 会 个 , □ 代 不会 try代 ,

以, 代 finally , try-with-resource

1.1 使 finally代 块

- 与 try 代 不 , finally 代 会 不 try 代 之 catch 代 中 会 , 以 了

```
public void closeResourceInFinally() {
    FileInputStream inputStream = null;
    try {
        File file = new File("./tmp.txt");
        inputStream = new FileInputStream(file);
        // use the inputStream to read a file
    } catch (FileNotFoundException e) {
        log.error(e);
    } finally {
        if (inputStream != null) {
            try {
                inputStream.close();
            } catch (IOException e) {
                log.error(e);
            }
        }
    }
}
```

1.2 Java 7 try-with-resource

- 了 AutoCloseable , 以 个 Java 了 个 try 中 , 会 try 代

```
public void automaticallyCloseResource() {
    File file = new File("./tmp.txt");
    try (FileInputStream inputStream = new FileInputStream(file);) {
        // use the inputStream to read a file
    } catch (FileNotFoundException e) {
        log.error(e);
    } catch (IOException e) {
        log.error(e);
    }
}
```

2. 优先明 异常

- , , 事 个 之 , 会 且
- 他们 API 且
- , 事件 , 个 NumberFormatException 个 IllegalArgumentException 个不

```
public void doNotDoThis() throws Exception {
    ...
}
public void doThis() throws NumberFormatException {
    ...
}
```

3. 对异常 文 明

- 上 , 也 为了 , 从
以
Javadoc @throws , 且

```
public void doSomething(String input) throws MyBusinessException {
    ...
}
```

4. 使 描 抛出异常

- 中, , 人 , 从 以 位 , 中
• 但 不 , 为 Exception ,
两
- 个 , 了 以, 不
个 NumberFormatException 以 String ,
java.lang.Long

```
try {
    new Long("xyz");
} catch (NumberFormatException e) {
    log.error(e);
}
```

5. 优先捕 具 异常

- IDE 以 个 不 , 们会
代
- 但 于, 个 catch 会 ,
IllegalArgumentException 不会 NumberFormatException
catch , 为 entException
- 优 不 catch
- 以 下 代 个 try-catch 个 catch
NumberFormatException 个 NumberFormatException
IllegalArgumentException

```
public void catchMostSpecificExceptionFirst() {
    try {
        doSomething("A message");
    } catch (NumberFormatException e) {
        log.error(e);
    } catch (IllegalArgumentException e) {
        log.error(e)
    }
}
```

6. 不 捕 Throwable

- Throwable 以 catch 中 , 但 不 !
- catch 中 Throwable , 不仅会 , 也 JVM , 不 严 OutOfMemoryError StackOverflowError 两 之 ,
- 以, 不 Throwable , 于 下

```
public void doNotCatchThrowable() {
    try {
        // do something
    } catch (Throwable t) {
        // don't do this!
    }
}
```

7. 不 忽 异常

- , 不会 , 了 个catch , 但 任

```
public void doNotIgnoreExceptions() {
    try {
        // do something
    } catch (NumberFormatException e) {
        // this will never happen
    }
}
```

- 但 会 于 , 代 不 会 (了 代), 于 , 位
-

```
public void logAnException() {
    try {
        // do something
    } catch (NumberFormatException e) {
        log.error("This should never happen: " + e);
    }
}
```

8. 不 录并抛出异常

- 中 以 代 中 会 下:

```
try {
    new Long("xyz");
} catch (NumberFormatException e) {
    log.error(e);
    throw e;
}
```

- 个 但 会 个 下:

```
17:44:28,945 ERROR TestExceptionHandler:65 - java.lang.NumberFormatException:
For input string: "xyz"
Exception in thread "main" java.lang.NumberFormatException: For input string:
"xyz"
at java.lang.NumberFormatException.forInputString(NumberFormatException.java:65)
at java.lang.Long.parseLong(Long.java:589)
at java.lang.Long.(Long.java:965)
at
com.stackify.example.TestExceptionHandler.logAndThrowException(TestExceptionHan
dling.java:63)
at
com.stackify.example.TestExceptionHandler.main(TestExceptionHandler.java:58)
```

- 上 , 也 , 么 以 为 义

```
public void wrapException(String input) throws MyBusinessException {
    try {
        // do something
    } catch (NumberFormatException e) {
        throw new MyBusinessException("A message that describes the error.", e);
    }
}
```

- , 仅仅 } , 中

9. 包 异常时不 抛弃原始 异常

- 为 义 个 以 为
- 中 e) Exception 为 (: 下 代 NumberFormatException e 了 了 个 Throwable 为 会丢 会 口事件

```
public void wrapException(String input) throws MyBusinessException {
    try {
        // do something
    } catch (NumberFormatException e) {
        throw new MyBusinessException("A message that describes the error.", e);
    }
}
```

10. 不 使 异常控制 序

- 不 if 件 下,
- 不 习 , 会严

11. 使 准异常

- 以 , 不 义 Java API 了上 不
- 中 Java API , 不
- 于 代

12. 异常会影响

- 个 会 1~5ms, 个 个 之 传 , 会 个
- 仅 下 ;
- 下 ;
- 于Java , 但 中 不 , 为 下
- 不 了 ,

13.

- 上 代 , 不 事 为了
- 不仅仅 个 API , 也 个 介 , 为了 , 个 中

异常处 - 巴巴Java开发手册

1. Java 中 义 以 RuntimeException 不 catch : NullPointerException, IndexOutOfBoundsException : catch NumberFormatException : if (obj != null) { try { obj.method(); } catch (NullPointerException e) {...}

2. 不，且，件：中
3. catch代代代不会代于
catch, , 也不于位, 代不任
4. 于, 中, , ,
5. try了事代中, catch, 事, 事
6. finally, 也 try-catch : JDK7
以上, try-with-resources
7. 不 finally 中 return : try 中 return , 不上
, finally 中 , return , 丢 try
中 :

```
private int x = 0;
public int checkReturn() {
    try {
        // x等于1, 此处不返回
        return ++x;
    } finally {
        // 返回的结果是2
        return ++x;
    }
}
```

1. 与, :
2. RuntimeException Throwable
: NoSuchMethodException
什么会 NoSuchMethodError ?
不, 了
NoSuchMethodError, 会
3. 么下会 null, 什
null
4. NPE, NPE产 : 1) 为
return 产 NPE : public int f() { return Integer
}, 为null, 为null 3)
isEmpty,)
, NPE 5) 于Session中 NPE , 6)
obj.getA().getB().getC(); 串, 产 NPE
: JDK8 Optional NPE
5. 义 unchecked / checked , new RuntimeException(), 不
Exception Throwable, 业 义 业 义
义, : DAOException / ServiceException
6. 于 http/api " ";
RPC 优 Result , isSuccess() " " " " :

于RPC Result : 1)
会产 2) 不 , new 义 , error
message, 于 不会 了 , 下,
传 也
7. 代 (Don't Repeat Yourself) , DRY :
代 , 会 代 , 以 ,
, , 件 : 个 中 个public ,
, 个 :
private boolean checkParam(DTO dto) {...}