close方法的处理

1. 利用可变参数和泛型的方法；

**public** **static** <T **extends** Closeable> **void** close(T ...ios){coding}

1. 利用可变参数和多态的方法；

**public** **static** **void** close2(Closeable ...ios){coding}

**public** **class** IOCloseUtils {

**public** **static** <T **extends** Closeable> **void** close(T ...ios) {

**for**(T io:ios) {

**try** {

io.close();

} **catch** (IOException e) {

e.printStackTrace();

}

}

}

**public** **static** **void** close2(Closeable ...ios) {

**for**(Closeable io:ios) {

**try** {

io.close();

} **catch** (IOException e) {

e.printStackTrace();

}

}

}

}

1. **利用JDK1.7的新特性：try- with -resources功能**。

try-with-resources语句是一个声明一个或多个资源的 try 语句。一个资源作为一个对象，必须在程序结束之后随之关闭。 try-with-resources语句确保在语句的最后每个资源都被关闭 。任何实现了 java.lang.AutoCloseable的对象, 包括所有实现了 java.io.Closeable 的对象, 都可以用作一个资源。

注意：此功能是JDK1.7以后提出的，所以必须确保Eclipse及项目的环境是JDK1.7之上。

**public** **static** **void** main(String[] args) **throws** IOException {

**try** (InputStream is = **new** FileInputStream(**new** File("e:\\aaa.txt"));

OutputStream os = **new** FileOutputStream(**new** File("e:\\bbb.txt"));) {

**byte**[] buff = **new** **byte**[10];

**int** len = 0;

**while** (-1 != (len = is.read(buff))) {

os.write(buff, 0, len);

}

os.flush();

}

}