2019-09

1. 倘若MySQL没有设置时区的话，有时候使用JDBC是会出错的，可以通过dos界面进入MySQL进行设置，或是进入MySQL的配置文件my.ini文件进行以下设置

**[mysqld]**

**#设置时区为东八区**

**default-time-zone = '+8:00'**

1. 为什么IO字符流在向文本输出一次字符后，接着前面输入而是覆盖？？？

**倘若不想覆盖，可以如此改动，例：FileWriter fw=new FileWriter(f,true);**

**在括号中加上true**

**源码：**

**public FileWriter(String fileName, boolean append) throws IOException {**

**super(new FileOutputStream(fileName, append));**

**}**

**对于append的解释：**

**append boolean if <code>true</code>, then data will be written**

**to the end of the file rather than the beginning.**

1. 为什么在类中的Scanner，多次调用后，每次都用close()关闭却会报错？？？

错误：Exception in thread "main" java.util.NoSuchElementException

at java.util.Scanner.throwFor(Scanner.java:862)

at java.util.Scanner.next(Scanner.java:1485)

at java.util.Scanner.nextInt(Scanner.java:2117)

at java.util.Scanner.nextInt(Scanner.java:2076)

atStudentsManager.StudentsManagerMain.main(StudentsManagerMain.java:22)

**只需要将close()删除就可以避免这个错误，原因是close不仅仅是关闭Scanner这个流，也将System.in给关闭了，而System.in这个标准输入一旦关闭将无法再次在这次运行里打开**

2019-09-18

1. 对于 类 类名 = 已建类名，以及 类 类名 = 已建类.属性类 的错误理解

**第一种是指新创建的类指向已建类的内存地址，对于这个新类的一切修改（除重新定向外）都是相当于已建类在修改自身的数据。**

**而第二种和第一种是一样的，类名指向的是这个属性类的内存地址。**

**重新定向是指，将类名的指向重新改变，比如 已建类=new 类() ，这时它的内存地址就会改变，如果之前已经有过类 类名 = 已建类名或是类 类名 = 已建类.属性类等将它的内存分配给其它对象的操作，那么它的之前的内存地址就不会释放，而之前赋给操作的新类也不会随着它的指向改变而改变，指向的仍然是之前的内存地址**

**PS：已建类=null 也相当于重定向**

**这些与 基本类型(或String) 变量 = 类.属性 不同，变量在创建时是会分配其内存的，这个操作只不过是将这个类属性的值赋给这个内存，并没有直接的内存引用**

**主要是对于内存地址的思考，一个类在运行时就会被加载，创建对象就会分配内存地址，这个地址贯穿整个程序（除非重定向），任何地方有用到这个对象的地方，都会被指向这个内存地址，它一旦发生改变，就等于内存地址中的属性被改变，所有用到这个对象的东西也必须随之改变。**

2019-09-19

1. BufferedReader br=new BufferedReader(new InputStreamReader(new URL("http://www.baidu.com").openStream(),"UFT-8"));

报错

java.io.UnsupportedEncodingException: UFT-8

at sun.nio.cs.StreamDecoder.forInputStreamReader(StreamDecoder.java:71)

at java.io.InputStreamReader.<init>(InputStreamReader.java:100)

at 文件流.网络操作流.main(网络操作流.java:16)

很简单的问题，报不支持的编码错误，这种一般都是UFT-8这个地方的编码写的不正确或是JDK不支持，比如我这儿就是写错的，应该是UTF-8

1. Test02[] tst=(Test02[]) al.toArray();

报错：类型转换异常

Exception in thread "main" java.lang.ClassCastException: [Ljava.lang.Object; cannot be cast to [LLambda表达式.Test02;

at Lambda表达式.Test02.main(Test02.java:33)

修改为Test02[] tst=al.toArray(new Test02[al.size()]);

2019-09-20

1.在使用Scanner时遇到的错误

***Scanner*** sc=**new** ***Scanner***(***System***.***in***);

***System***.***out***.***println***("输入字符串：");

***String*** str=sc.***next***();

***System***.***out***.***println***("输入目标字符串：");

***String*** srcstr=sc.***next***();

//Delete(StringBuilder str,String srcstr)删除str中的所有srcstr，返回-1或1

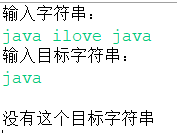
**if**(*Delete*(**new** ***StringBuilder***(str), srcstr)==-1)

***System***.***out***.***println***("没有这个目标字符串");

**else**

***System***.***out***.***println***(str);

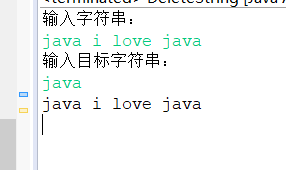
错误：



原因：

经过我调试发现sc.***next***()在你键盘输入时，只会获取空格前的数据，空格后的数据会作为下一个变量的数据返回

在我修改sc.next()变为sc.nextLine()后，出现了以下错误，字符串并没有被成功切割



原因：

**if**(*Delete*(**new** ***StringBuilder***(str), srcstr)==-1)

***System***.***out***.***println***("没有这个目标字符串");

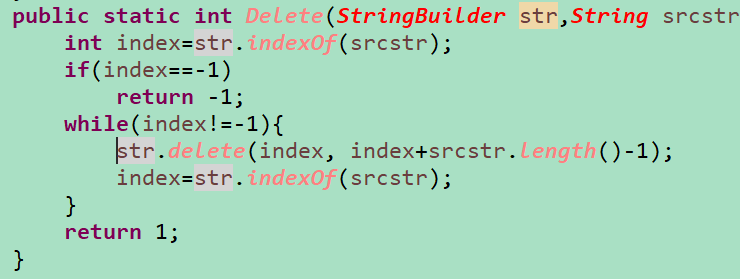
**else**

***System***.***out***.***println***(str);

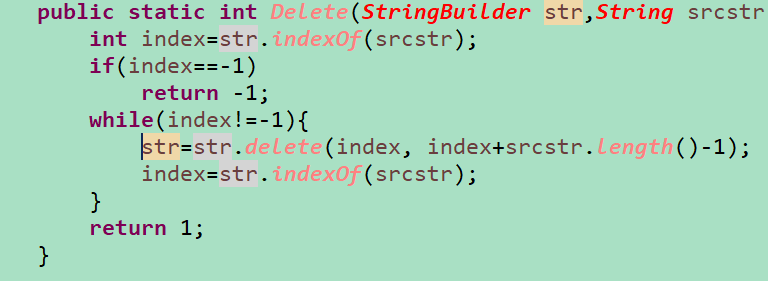
标记为红色的地方**new** ***StringBuilder***(str)将String的str转为StringBuilder，这儿重新为它开辟了一个内存，但是没有对象指向它，而我在Delete(StringBuilder str,String srcstr)中用到了

[delete](http://www.matools.com/file/manual/jdk_api_1.8_google/java/lang/StringBuilder.html#delete-int-int-)(int start, int end)，这个方法返回的是一个StringBuilder，即一个新的StringBuilder，所以我在方法中重新将str切割后的数据重新给了str，变相实现了，切割后，str数据的更新、

修改前



修改后



但是这样改了以后，还是老样子，这就到了上面的内存问题了，str的内存并不是指向这个**new** ***StringBuilder***(str)的，所以我只能在上面定义一个对象指向这个内存StringBuilder str1=new StringBuilder(str);再输出str1，才解决问题