Java面向对象程序设计课程设计实验报告三

一：补全程序题

* 6.3.2（1）
* 源代码：

|  |
| --- |
| package ExperimentSourceCode\_3**.**fill**;**  class NoLowerLetter **extends** Exception//定义异常类NoLowerLetter  **{**  public void print**()** **{**  System**.**out**.**printf**(**"%c"**,** '#'**);**  **}**  **}**  class NoDigit **extends** Exception //声明异常类NoDigit  **{**  public void print**()** **{**  System**.**out**.**printf**(**"%c"**,** '\*'**);**  **}**  **}**  class Test **{**  void printLetter**(**char c**)** **throws** NoLowerLetter **{**  **if** **(**c **<** 'a' **||** c **>** 'z'**)** **{**  NoLowerLetter noLowerLetter **=** **new** NoLowerLetter**();** // 创建NoLowerLetter对象  **throw** noLowerLetter**;** // 抛出noLowerLetter异常  **}** **else** **{**  System**.**out**.**print**(**c**);**  **}**  **}**  void printDigit**(**char c**)** **throws** NoDigit **{**  **if** **(**c **<** '1' **||** c **>** '9'**)** **{**  NoDigit noDigit **=** **new** NoDigit**();** // 创建NoDigit()类型对象  **throw** noDigit**;** // 抛出noDigit  **}** **else** **{**  System**.**out**.**print**(**c**);**  **}**  **}**  **}**  public class ExceptionExample **{**  public static void main**(**String args**[])** **{**  Test t **=** **new** Test**();**  **for** **(**int i **=** 0**;** i **<** 128**;** i**++)** **{**  **try** **{**  t**.**printLetter**((**char**)** i**);**  **}** **catch** **(**NoLowerLetter e**)** **{**  e**.**print**();**  **}**  **}**  **for** **(**int i **=** 0**;** i **<** 128**;** i**++)** **{**  **try** **{**  t**.**printDigit**((**char**)** i**);**  **}** **catch** **(**NoDigit e**)** **{**  e**.**print**();**  **}**  **}**  **}**  **}** |

* 运行结果：



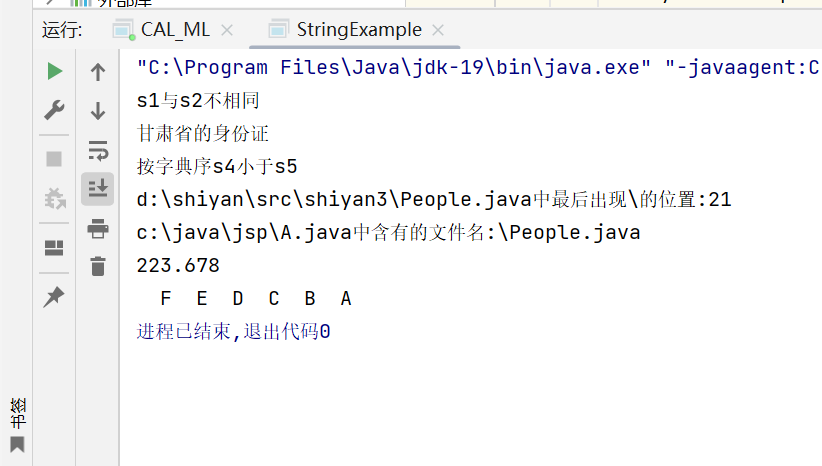
* 分析学习：

代码中演示了如何使用自定义异常类和异常处理机制来处理在程序执行过程中可能发生的错误。在**printLetter**和**printDigit**方法中，如果输入的字符不是小写字母或数字，就会抛出相应的异常，程序将在**try**块中捕获这些异常并输出错误信息。

* 6.3.2（2）
* 源代码：

|  |
| --- |
| package ExperimentSourceCode\_3**.**fill**;**  class StringExample **{**  public static void main**(**String args**[])** **{**  String s1 **=** **new** String**(**"you are a student"**),**  s2 **=** **new** String**(**"how are you"**);**  **if** **(**s1**.**equals**(**s2**))** **{** // 判断s1与s2是否相同  System**.**out**.**println**(**"s1与s2相同"**);**  **}** **else** **{**  System**.**out**.**println**(**"s1与s2不相同"**);**  **}**  String s3 **=** **new** String**(**"620102197210015331"**);**  **if** **(**s3**.**startsWith**(**"620102"**))** **{** // 判断s3的前缀是否是“620102”  System**.**out**.**println**(**"甘肃省的身份证"**);**  **}**  String s4 **=** **new** String**(**"你"**),**  s5 **=** **new** String**(**"我"**);**  **if** **(**s4**.**compareTo**(**s5**)** **>** 0**)** **{**  System**.**out**.**println**(**"按字典序s4大于s5"**);**  **}** **else** **{**  System**.**out**.**println**(**"按字典序s4小于s5"**);**  **}**  int position **=** 0**;**  String path **=** "d:\\shiyan\\src\\shiyan3\\People.java"**;**  position **=** path**.**lastIndexOf**(**"\\"**);**//获取path中最后出现目录分隔符号的位置  System**.**out**.**println**(**"d:\\shiyan\\src\\shiyan3\\People.java中最后出现\\的位置:" **+** position**);**  String fileName **=** path**.**substring**(**position**);** //获取path中“People.java”子字符串  System**.**out**.**println**(**"c:\\java\\jsp\\A.java中含有的文件名:" **+** fileName**);**  String s6 **=** **new** String**(**"100"**),** s7 **=** **new** String**(**"123.678"**);**  int n1 **=** Integer**.**parseInt**(**s6**);** // 将s6转化成int型数据  double n2 **=** Double**.**parseDouble**(**s7**);** // 将s7转化成double型数据  double n **=** n1 **+** n2**;**  System**.**out**.**println**(**n**);**  String s8 **=** **new** String**(**"ABCDEF"**);**  char a**[]** **=** s8**.**toCharArray**();** // 将s8存放到数组a中  **for** **(**int i **=** a**.**length **-** 1**;** i **>=** 0**;** i**--)** **{**  System**.**out**.**printf**(**"%3c"**,** a**[**i**]);**  **}**  **}**  **}** |

* 运行结果：



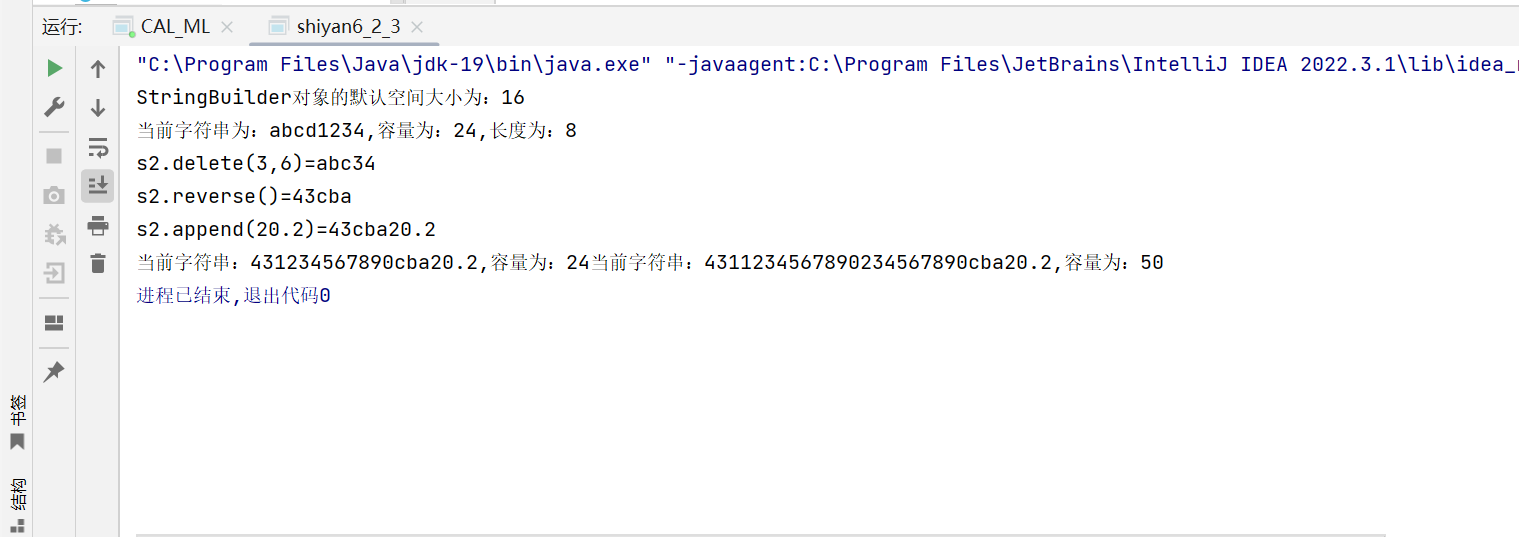
* 分析学习：

代码主要是使用了一些**String**类的方法来演示如何操作字符串。在这段代码中没有显式地抛出任何异常，但这段代码中方法的调用可能会出现异常，例如**Inter.parseInt()**和**Double.parseDouble()**方法，它们在转换字符串时可能会抛出**NumberFormatException**(目标转换的结果不是数字)异常。

* 6.3.2（3）
* 源代码：

|  |
| --- |
| package ExperimentSourceCode\_3**.**fill**;**  public class shiyan6\_2\_3 **{**  public static void main**(**String**[]** args**)** **{**  StringBuilder s1 **=** **new** StringBuilder**();**  System**.**out**.**println**(**"StringBuilder对象的默认空间大小为：" **+** s1**.**capacity**());**  StringBuilder s2 **=** **new** StringBuilder**(**"abcd1234"**);**  System**.**out**.**println**(**"当前字符串为：" **+** s2**.**toString**()** **+** ",容量为：" **+** s2**.**capacity**()** **+** ",长度为：" **+** s2**.**length**());**  System**.**out**.**println**(**"s2.delete(3,6)=" **+** s2**.**delete**(**3**,** 6**).**toString**());**  System**.**out**.**println**(**"s2.reverse()=" **+** s2**.**reverse**().**toString**());**  System**.**out**.**println**(**"s2.append(20.2)=" **+** s2**.**append**(**20.2**).**toString**());**  s2 **=** s2**.**insert**(**2**,** "1234567890"**);**  System**.**out**.**print**(**"当前字符串：" **+** s2**.**toString**()** **+** ",容量为：" **+** s2**.**capacity**());**  s2 **=** s2**.**insert**(**3**,** "1234567890"**);**  System**.**out**.**print**(**"当前字符串：" **+** s2**.**toString**()** **+** ",容量为：" **+** s2**.**capacity**());**  **}**  **}** |

* 运行结果：



* 分析学习：

这段代码中演示到了StringBuilder类的使用。StringBuilder类是Java中用于字符串操作的一个类，它与String类的区别在于它是可变的，而String是不可变的。StringBuilder类的主要作用是对字符串进行增、删、改等操作，同时也可以进行字符串的拼接、反转等操作。

二：设计程序题

* 6.3.3（1）：
* 题目描述：设计一个命令行计算器程序，处理所有可能的输入异常。
* 设计思路：命令行计算程序，类似于实验二中计算交通类中命令行参数的输入，可以将整个计算过程分为三个过程：

1.命令行参数输入args[…]，并将这些参数输入定义的计算器类中，这个过程可能会出现的异常：命令行输入参数不足，可能出现**ArrayIndexOutOfBoundsException**（数组下标越界）的异常情况。

2.命令行输入参数是字符串，对这些字符串需要进行相应处理才能转换为可以进行运算的整型或者字符，这个过程因为调用的相关Interger类的方法将字符串转换为整型，可能会出现的异常：**NumberFormatException**(参数输入异常，无法转换为数字)。

3．以switch循环的方式，判断输入符号类型，对输入的两个整型进行相应运算操作，并输出最终结果，这个阶段由于除数为0时运算无意义，可能发生的异常：**ArithmeticException**（数学类溢出），也有可能由于输入运算符号错误，导致switch没有进行相应计算，前者可以异常抛出，后者只能手动判断。

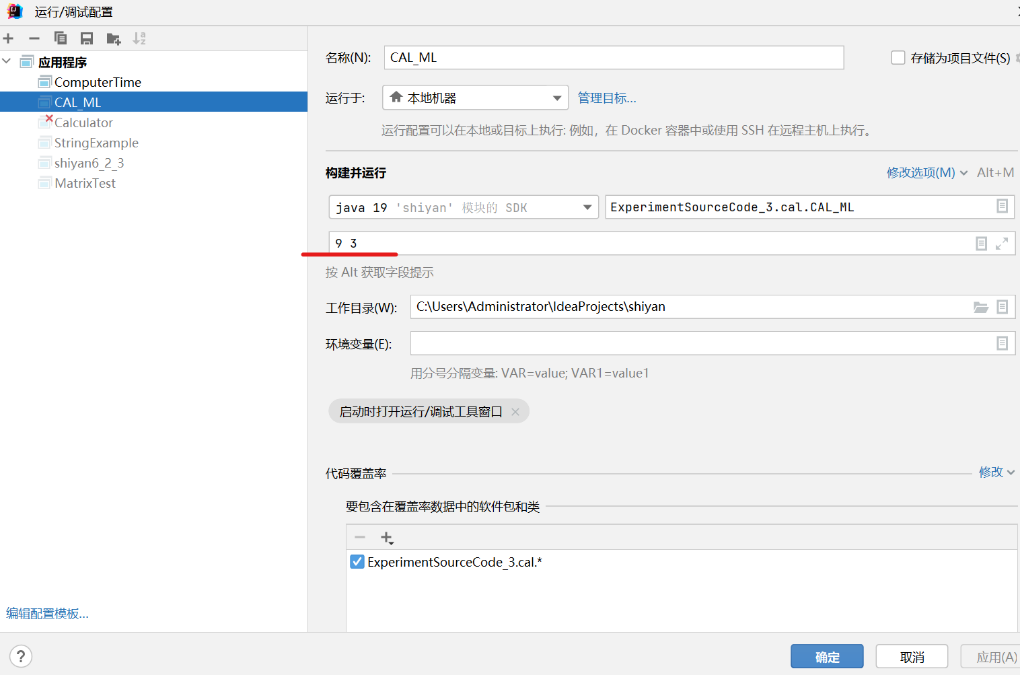
4.异常处理的方式，上述异常可以通过try、catch、finally关键字进行处理，其中try模块中是可能存在问题的代码，catch后接对应的异常种类，try模块中只要出现相关异常，就会转到catch模块运行，不管是否有异常，finally模块的代码最后且一定运行。对上述问题，均可以采用重新输入错误数据的方法得到解决，如被除数为0时，则抛出异常，让使用者重新输入一个不为0的数进行运算。

* 源代码：

|  |
| --- |
| package ExperimentSourceCode\_3**.**cal**;**  **import** java**.**util**.**Scanner**;**  public class CAL\_ML **{**  int a**,** b**;**  char t**;**  public CAL\_ML**(**String a**,** String b**,** String t**)** **{**  Scanner sc **=** **new** Scanner**(**System**.**in**);**  **try** **{**  **this.**a **=** Integer**.**valueOf**(**a**);**  **this.**b **=** Integer**.**valueOf**(**b**);**  **this.**t **=** t**.**charAt**(**0**);**  **}** **catch** **(**NumberFormatException e**)** **{**  System**.**out**.**println**(**"出现输入参数无法转化为数字异常，请重新输入除数与被除数："**);**  **this.**a **=** sc**.**nextInt**();**  **this.**b **=** sc**.**nextInt**();**  **this.**t **=** t**.**charAt**(**0**);**  **}**//可能出现无法转化为数字的异常,出现即重新转化  **}**//输入字符串转换过程  public static void main**(**String**[]** args**)** **{**  String a**,** b**,** t**;**  Scanner sc **=** **new** Scanner**(**System**.**in**);**//键盘输入数据  **try** **{**  CAL\_ML cal\_ml **=** **new** CAL\_ML**(**args**[**0**],** args**[**1**],** args**[**2**]);**  System**.**out**.**println**(**"result=" **+** cal\_ml**.**cal**());**  **}** **catch** **(**ArrayIndexOutOfBoundsException e**)** **{**  System**.**out**.**println**(**"命令行传来参数个数异常" **+** "\n" **+** "请重新输入两个操作数和操作符："**);**  a **=** sc**.**next**();**  b **=** sc**.**next**();**  t **=** sc**.**next**();**  CAL\_ML cal\_ml **=** **new** CAL\_ML**(**a**,** b**,** t**);**  System**.**out**.**println**(**"result=" **+** cal\_ml**.**cal**());**  **}** **finally** **{**  System**.**out**.**print**(**"不管是否异常，总是要执行"**);**  **}**  **}**  public int cal**()** **{**  Scanner sc **=** **new** Scanner**(**System**.**in**);**  int result **=** 0**;**  **try** **{**  **while** **(**t **!=** '+' **&&** t **!=** '-' **&&** t **!=** '\*' **&&** t **!=** '/'**)** **{**  System**.**out**.**println**(**"输入不是运算符号(+、-、\*、/)!" **+** "\n" **+** "重新输入运算符号："**);**  t **=** sc**.**next**().**charAt**(**0**);**  **}**  **switch** **(**t**)** **{**  **case** '+'**:**  result **=** a **+** b**;**  **break;**  **case** '-'**:**  result **=** a **-** b**;**  **break;**  **case** '\*'**:**  result **=** a **\*** b**;**  **break;**  **case** '/'**:**  result **=** a **/** b**;**  **break;**  **}**  **}** **catch** **(**ArithmeticException e**)** **{**  **while** **(**b **==** 0**)** **{**  System**.**out**.**println**(**"出现除数为0的异常" **+** "\n" **+** "请重新输入除数："**);**  b **=** sc**.**nextInt**();**  result **=** a **/** b**;**  **}**  **return** result**;**  **}**  **return** result**;**  **}**//计算方法  **}** |

* 运行结果：

1.命令行参数设置：



2.运行结果（三个过程，每一次都输入对应异常，观察程序中的异常处理）



**==>**首先是命令行传来的参数个数异常，进行处理后刻意输入无法转换为数字的“\”和无法识别的运算符号“？”，观察异常处理，进行相应处理后输入0作为除数进行除法运算，在观察异常处理，最后得到结果：9/3=3，最后跳到finally模块运行输出语句。

* 反思与问题：

遇到的最大问题是在对异常处理过程的修正，由上异常分析共有四种异常，其中有三种是可以通过catch抛出的，最开始我将所有过程代码放到一个try模块里，跟上三个catch模块，最后发现最后虽然抛出了异常，但并不能向之前一样继续处理**NumberFormatException**的异常，因为这时候已经跳出了try模块，进入到第一个检测catch（**ArrayIndexOutOfBoundsException**）模块中了，而catch模块只关注try模块中的异常，所以对异常的考虑，分为三个try-catch模块，确保每一个异常都可以抛出。

