**《Java程序设计》上机作业5**

|  |  |
| --- | --- |
| **学号：** | **202103151422** |
| **姓名：** | **温家伟** |
| **班级：** | **大数据分析2101** |
| **日期：** | **11.19** |

1. 编写程序：**ArithmeticT1**，要求以命令行参数的形式读取三个数据：两个整数及其中间的一个运算符（+, ‐, \*, /）。程序中针对输入的整数和运算符进行四则运算，并输出运算公式和结果。

注意：程序需要使用异常处理机制对非整数数值输入进行处理（提示“Wrong Input：”后跟对应的错误输入）。另外，程序也需要对运算符的输入进行判断。

|  |
| --- |
| >java ArithmeticT1 12 + 34  12 + 34 = 46.0  >java ArithmeticT1 12 - 34  12 - 34 = -22.0  >java ArithmeticT1 12 "\*" 34  12 \* 34 = 408.0  >java ArithmeticT1 12 / 34  12 / 34 = 0.35294117647058826  >java ArithmeticT1 7/8 – 34  Wrong Input: 7/8  >java ArithmeticT1 78 – 3;4  Wrong Input: 3;4 |

|  |
| --- |
| * **问题分析与解答**   读取命令行参数：程序需要能够接收命令行输入的三个数据，包括两个整数和一个运算符。  异常处理：对于非整数数值输入，程序需要捕获异常并给出相应提示；另外，对于不支持的运算符输入，也需要进行异常处理。  四则运算：针对输入的整数和运算符进行相应的四则运算，并输出结果。 |
| * **代码实现** |
| * **运行结果** |
| * **思考及总结**   首先，我们可以使用 try-catch 块来捕获可能出现的异常，包括对非整数数值输入的处理以及对运算符输入的判断。在捕获到异常后，需要给出相应的错误提示信息。其次，根据接收到的整数和运算符进行相应的四则运算，并输出运算公式和结果。在整个过程中，需要充分考虑用户可能输入的各种情况，确保程序能够健壮地运行，同时给出清晰的用户提示信息。  并且，我最开始在windows下做这道题，”\*”也报错，试了很久，然后用ubuntu试了一下，成功了，我想可能是当时在Windows里装java环境的时候哪里错了吧哈哈哈。 |

1. (ArrayIndexOutBoundsException) 按照以下要求编写程序：

* 创建一个包含100个整数的数组，元素随机生成；
* 提示用户输入一个整数，程序返回以用户的输入作为下标的数组元素值；
* 如果下标超出数组下标范围，则提示：下标越界，无此元素。

|  |
| --- |
| * **问题分析**   我们需要首先生成包含100个随机整数的数组，可以使用Java中的Random类来生成随机数并填充数组。  然后，程序需要提示用户输入一个整数作为数组下标，可以使用Scanner类来接收用户输入。  接着，程序需要根据用户输入的下标返回对应的数组元素值，同时需要使用try-catch语句捕获可能出现的数组下标越界异常，并给出相应的提示信息。 |
| * **代码实现** |
| * **运行结果** |
| * **思考及总结**   在实现此程序时，我们需要注意生成随机数的范围、用户输入的下标验证以及异常处理等方面。  通过合理地使用数组、随机数生成、用户输入和异常处理等知识点，能够很好地完成这道题目的要求。  此题也提醒我们在处理数组下标时要格外小心，确保不会发生数组下标越界的情况，同时要考虑异常情况的处理，提高程序的健壮性。 |

1. 自定义一个异常类，叫做负年龄异常: **MinusAgeException**。再定义类**Person，**包含姓名name，年龄age等属性，在**Person**类中有一个方法setAge()，若年龄为负，则抛出**MinusAgeException**异常。

另外编写驱动类，对上述方法进行测试。

|  |
| --- |
| * **问题分析**   首先，我们需要定义一个名为 MinusAgeException 的自定义异常类，这个类需要继承自 Exception 类。  其次，我们需要定义一个 Person 类，包含姓名和年龄属性，并且实现了 setAge 方法用于设置年龄。在 setAge 方法中，需要对输入的年龄进行判断，若年龄为负数，则抛出 MinusAgeException 异常。  最后，我们需要编写一个驱动类 TestPerson 进行测试，测试中需要分别设置正常年龄和负年龄的情况，观察是否能够正确捕获并处理自定义异常。 |
| * **代码实现** |
| * **运行结果** |
| * **思考及总结**   在本题中，我们学习了如何定义自定义异常类，并且通过实现一个简单的业务逻辑来加深理解异常处理机制。  通过自定义异常 MinusAgeException，我们可以更好地对特定的业务异常情况进行处理，提高代码的可读性和健壮性。  合理使用异常处理机制可以使程序更加稳健，在面对异常情况时能够给出清晰的提示和处理方式，提高了代码的可靠性和容错性。 |

1. 编写一个程序，读取用户的输入：一个加法表达式，程序可以输出计算公式以及计算表达式的结果。表达式形式为“12+24+5”，如果输入格式错误将产生异常。

提示：用String类的split方法 public String[] split(String regex) 可以根据给定分隔符（或正则表达式）的对字符串进行拆分。

|  |
| --- |
| * **问题分析**   输入格式的验证：需要确保用户输入的加法表达式符合指定的格式，即由数字和加号组成。我们可以利用正则表达式或其他方法对用户输入进行验证，以确保输入的合法性。  异常处理：在进行字符串拆分、类型转换等操作时，可能会发生NumberFormatException、ArrayIndexOutOfBoundsException等异常。我们需要通过合适的异常捕获和处理机制，向用户提供友好的错误提示信息，避免程序因为异常而崩溃。 |
| * **代码实现** |
| * **运行结果** |
| * **思考及总结**   通过本次编程练习，我们进一步加深了对异常处理机制的理解，学会了如何使用 try-catch 块来捕获和处理可能出现的异常。  我们也学会了利用 String 类的 split 方法来对字符串进行拆分，这对于处理用户输入的分隔符分割非常有用。  合理的异常处理能够提高程序的健壮性和用户体验，使得程序能够在面对异常输入时给出清晰的提示信息，而不至于崩溃或产生意外结果。 |

1. 定义类**Triangle**，其中有成员 x, y, z 作为三边长，构造方法Triangle(x, y, z)分别给x, y, z 赋值，包含两个方法：显示三角形信息(三个边长) 方法showInfo；返回面积方法getArea。在方法中，当三条边不能构成一个三角形时要抛出自定义异常NoTriangleException，否则显示正确信息。

另外，定义主类**TestTriangle**，主方法中构造一个**Triangle**对象（三边为作为命令行输入的三个整数），显示三角形信息和面积。要求处理程序中可能出现的异常。

|  |
| --- |
| * **问题分析**   需要定义一个 Triangle 类，其中包括成员变量 x、y、z 作为三角形的三条边长，构造方法 Triangle(x, y, z) 用于给 x、y、z 赋值，以及两个方法 showInfo 和 getArea。  在 getArea 方法中需要处理当三条边不能构成一个三角形时抛出自定义异常 NoTriangleException。  另外，需要定义主类 TestTriangle，在主方法中构造一个 Triangle 对象，显示三角形信息和面积，并处理可能出现的异常。 |
| * **代码实现** |
| * **运行结果** |
| * **思考及总结**   首先在 Triangle 类中定义成员变量和构造方法，确保能够正确初始化三角形的三条边长。  然后在 getArea 方法中判断三条边能否构成三角形，若不能则抛出自定义异常 NoTriangleException。  在主类 TestTriangle 的主方法中，接收命令行输入的三个整数作为三角形的三条边长，构造 Triangle 对象，并调用相应的方法处理可能出现的异常。 |