SY-023

实 验 报 告

课程名称： Python程序设计 \_

系部名称： 计算机科学与技术 \_

专业班级： 计科22-3班 \_\_

学生姓名： 邹晨骏 \_\_

学 号： 20222163 \_

指导教师： 曲天伟 \_\_ \_

**黑龙江工程学院教务处制**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 实验项目 | 实验七 异常编程实验 | | | | | 实验日期 | 2024.10.21 |
| 实验地点 | A607 | 同组人数 | 1 | 分工 |  | 实验台号 |  |
| 实验类型 | □验证性 □ 综合性 □ 设计性 □ 其 他 | | | | | | |
| 一、实验目的  掌握异常、异常的捕获方式，熟悉 raise语句和assert语句的使用，会使用自定义异常解决问题。 | | | | | | | |
| 二、实验仪器设备实验工具软件  笔记本电脑，python3.8.2解释器，pycharm2021编辑器 | | | | | | | |
| 三、实验要求   1. **练习异常的捕获。**   1）使用try-except语句捕获异常  try:  可能出错的代码  except [异常类型 [as error]]: # as error将捕获到的异常对象赋error，可以捕获异常描述信息  捕获异常后的处理代码  2）输入下面的代码，练习捕获单个异常  num\_one = int(input("请输入被除数："))  num\_two = int(input("请输入除数："))  try:  print("结果为", num\_one / num\_two)  except ZeroDivisionError:  print("出错了")  3）输入下面的代码，练习捕获多个异常  num\_one = int(input("请输入被除数："))  num\_two = int(input("请输入除数："))  try:  print("结果为", num\_one / num\_two)  except (ZeroDivisionError, ValueError) as error:  print("出错了，原因：", error)  4）输入下面的代码，练习捕获全部异常  num\_one = int(input("请输入被除数："))  num\_two = int(input("请输入除数："))  try:  print("结果为", num\_one / num\_two)  except Exception as error:  print("出错了，原因：", error)  5）输入下面的代码练习异常结构中的else子句  else子句可以与try-except语句组合成try-except-else结构，若try监控的代码没有异常，程序会执行else子句后的代码。  first\_num = int(input("请输入被除数："))  second\_num = int(input("请输入除数："))  try:  res = first\_num/second\_num  except ZeroDivisionError as error:  print('异常原因：',error)  else:  print(res)  6）finally子句可以和try-except一起使用，无论try子句监控的代码是否产生异常，finally子句都会被执行。finally子句多用于预设资源的清理操作，如关闭文件、关闭网络连接  输入下面的代码练习finally子句的使用  try:  file = open('text.txt', mode='r', encoding='utf-8')  print(file.read())  except (FileNotFoundError,NameError) as error:  file = open('text.txt', mode='w', encoding='utf-8')  finally:  file.close()  print('文件已关闭')   1. **练习raise语句和assert语句的使用**   1）使用raise语句可以显式地抛出异常， raise语句的语法格式如下  raise 异常类 # 格式1：使用异常类名引发指定的异常  raise 异常类对象 # 格式2：使用异常类的对象引发指定的异常  raise # 格式3： 使用刚出现过的异常重新引发异常  练习下面异常抛出的语句  1）raise IndexError()  2）raise IndexError('索引下标超出范围')  3）  try:  raise IndexError  except:  raise  2）使用assert语句抛出异常  assert 表达式[, 异常信息]（表达式为false会出发asserterror异常，表达式为true什么都不做）  练习下面的assert语句的使用  num\_one = int(input("请输入被除数："))  num\_two = int(input("请输入除数："))  assert num\_two != 0, '除数不能为0' # assert语句判定num\_two不等于0  result = num\_one / num\_two  print(num\_one, '/', num\_two, '=', result)   1. **自定义异常编程练习**   有时需要自定义异常类，以满足当前程序的需求。就是将不满足程序逻辑的情况反馈给用户，用来报错。自定义异常的方法比较简单，只需要创建一个继承Exception类或Exception子类的类（类名一般以“Error”为结尾）即可。  需求：密码长度不足，则报异常（用户输入密码，如果输入不足3位，则报错，即抛出自定义异常，并捕获该异常。）步骤如下：  1）自定义异常类  2）抛出异常  3）捕获该异常  输入下面代码，练习自定义异常使用  class ShortInputError(Exception):  '''自定义异常类'''  def \_\_init\_\_(self, length, atleast):  self.length = length # 输入的密码长度  self.atleast = atleast # 限制的密码长度  try:  text = input("请输入密码：")  if len(text) < 3:  raise ShortInputError(len(text), 3)  except ShortInputError as result:  print("ShortInputError：输入的长度是%d，长度至少应是 % d" %  (result.length, result.atleast))  else:  print("密码设置成功")   1. **用自定义异常解决问题** （请自己动手编写代码解决下面的问题。）   1）编写程序，计算圆的面积，若半径为负值则抛出异常（圆的面积公式：）。异常信息为：“输入的半径不能为负数”  2）编写程序，输入三角形三条边判断能否构成三角形，若能构成则计算三角形的周长，否则引发异常，异常信息为：三边长不能构成三角形。  3）假设某网站只允许用户上传jpg、png和jpeg格式的文件，要求编写代码，通过异常捕获语句实现用户上传头像格式检测的功能，如果格式正确打印上传成功，否则输出异常信息：图片格式错误，仅支持jpg/png/bmp格式。 | | | | | | | |
| 四、实验步骤  **4. 用自定义异常解决问题**  (1)  **import** math  **class** CircleError(Exception):  **def** \_\_init\_\_(self, value):  self.value = value  **try**:  value = int(input(**"Enter a number: "**))  **if** value < 0:  **raise** CircleError(value)  **except** CircleError **as** result:  print(**"InputError：输入的半径是%d，小于0"** %(result.value)) **else**:  print(**"半径输入正确，面积是%f"** %(value \*\* 2 \* math.pi))  (2)  **import** math  **class** InputError(Exception):  **def** \_\_init\_\_(self, a:int,b:int,c:int):  self.a = a  self.b = b  self.c = c  **def** check(a:int,b:int,c:int):  **if** b + c <= a **or** a + c <= b **or** a + b <= c:  **return False  else**:  **return True try**:  a, b, c = map(int, input(**"输入三角形的3个边长（以空格分隔）："**).split())   **if** check(a,b,c) == **False**:  **raise** InputError(a,b,c)  **except** InputError **as** result:  print(**"InputError：输入的边长为 %d,%d,%d"** %(result.a,result.b,result.c)) **else**:  print(**"边长输入正确,输入的边长为 %d,%d,%d"** %(a,b,c))  (3)  **class** FormatError(Exception):  **def** \_\_init\_\_(self, filename: str):  self.filename = filename   **def** check\_file\_format(filename: str):  *# 允许的文件格式* allowed\_formats = [**'jpg'**, **'png'**, **'jpeg'**,**'bmp'**]  *# 获取文件扩展名* file\_extension = filename.split(**'.'**)[-1].lower()  *# 检查扩展名是否在允许的格式中* **if** file\_extension **not in** allowed\_formats:  **return False  return True   try**:  *# 获取用户输入的文件名* filename = input(**"请输入上传的文件名："**)   *# 检查文件格式是否正确* **if not** check\_file\_format(filename):  **raise** FormatError(filename)  **except** FormatError **as** e:  print(**f"FormatError：文件 '{**e.filename**}' 格式不正确，只允许上传 jpg、png 或 jpeg 格式的文件。"**) **else**:  print(**f"文件 '{**filename**}' 上传成功，格式正确。"**) | | | | | | | |
| 五．实验结果  **1. 练习异常的捕获。**  **(1)**    (2)    (3)    (4)    (5)    (6)    **2. 练习raise语句和assert语句的使用**  (1)        (2)     1. **自定义异常编程练习**      1. **用自定义异常解决问题**   （1）    （2）    （3） | | | | | | | |
| 六、实验中遇到的问题及解决方法 | | | | | | | |