ch10-2 异常

除0错误 ZeroDivisionError

```
print(5/0)

Traceback (most recent call last):
File "division.py", line 1, in <module>
print(5/0)
ZeroDivisionError: division by zero
```

try-except

```
try:
    print(5/0)
except ZeroDivisionError:
    print("You can't divide by zero!")
```

文件不存在FileNotFoundError

方法split() 以空格为分隔符将字符串分拆成多个部分，并将这些部分都存储到一个列表

```
filename = 'alice.txt'
try:
    with open(filename) as f_obj:
        contents = f_obj.read()
except FileNotFoundError:
    msg = "Sorry, the file " + filename + " does not exist."
    print(msg)
else:
# 计算文件大致包含多少个单词
    words = contents.split()
    num_words = len(words)
    print("The file " + filename + " has about " + str(num_words) + " words.")
```

仅当try 代码块成功执行时才执行else 代码块。输出指出了文件alice.txt包含多少个单词

try-except-else 代码块的工作原理大致如下：

- Python尝试执行try 代码块中的代码；
- 只有可能引发异常的代码才需要放在try 语句中。
- 有时候，有一些仅在try 代码块成功执行时才需要运行的代码；这些代码应放在else 代码块中。
- except 代码块告诉Python，如果它尝试运行try 代码块中的代码时引发了指定的异常，该怎么办。

```
def count_words(filename):
    """计算一个文件大致包含多少个单词"""
    try:
        # --snip--
    except FileNotFoundError:
     pass
    else:
        #--snip--

filenames = ['alice.txt', 'siddhartha.txt', 'moby_dick.txt', 'little_women.txt']
for filename in filenames:
    count_words(filename)
```

```
The file alice.txt has about 29461 words.
The file moby_dick.txt has about 215136 words.
The file little_women.txt has about 189079 words.
```

pass 语句。现在，出现FileNotFoundError 异常时，将执行except 代码块中的代码，但什么都不会发生。
pass 语句还充当了占位符，它提醒你在程序的某个地方什么都没有做，并且以后也许要在这里做些什么。例如，在这个程序中，我们可能决定将找不到的文件的名称写入到文件missing_files.txt中。用户看不到这个文件，但我们可以读取这个文件，进而处理所有文件找不到的问题。