人工智能程序设计

M1 Python程序设计基础

6 Python异常

张 莉

筛选数字

编写程序，输入字符串（可能分成若干段），将字符串中所有的数字字符串(除去尾部标点符号外可构成一个整数或浮点数，例如100times不是，句子中的56.78.23不是，句尾的45.78.是，或是一个单独的数字)转换成浮点数并输出。如果没有数字字符串，则输出 ‘Not Found!'。字符串中可能包括以下标点符号：","、"."、“”、“?”和"!"，标点符号不会连续出现。

输入样例：

I have 5.67 yuan. You have 5.68 yuan. I am 100times angry.

输出样例：

5.67

5.68

人工智能程序设计

1 PYTHON中的异常

程序设计错误

- 语法错误
- 运行时错误
- 逻辑错误

```
6 = x * 2
```

```
x = 3 / 0
```

```
area = 2 * 3.14 * 3
```

异常（Exception）

Source

```
>>> 1 / 0
```

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

Source

```
>>> y = x + 1
```

```
Traceback (most recent call last):
  File "<pyshell#1>", line 1, in <module>
    y = x + 1
NameError: name 'x' is not defined
```

用异常对象（exception object）表示异常情况

- 查看异常类
  dir(__builtins__)

类 名	描 述
BaseException	所有异常的基类
Exception	常规异常的基类
AttributeError	对象不存在此属性
IndexError	序列中无此索引
IOError	输入/输出操作失败
KeyboardInterrupt	用户中断执行(通常输入Ctr-C)
KeyError	映射中不存在此键
NameError	找不到名字（变量）
SyntaxError	Python 语法错误
TypeError	对类型无效的操作
ValueError	传入无效的参数
ZeroDivisionError	除(或取模)运算的第二个参数为0

异常处理

```
if y != 0:
        print(x / y)
else:
        print('division by zero')
```

VS

try-except
异常处理语句

人工智能程序设计

2 捕捉异常

异常


```
num1 = int(input('Enter the first number: '))
num2 = int(input('Enter the second number: '))
print(num1 / num2)
```

```
Enter the first number: a
Traceback (most recent call last):
  File "C:\Python\programs\exception1.py", line 1, in <module>
    num1 = int(input('Enter the first number: '))
ValueError: invalid literal for int() with base 10: 'a'
```

try-except语句

File

```
try:
    num1 = int(input('Enter the first number: '))
    num2 = int(input('Enter the second number: '))
    print(num1 / num2)
except ValueError:
    print('Please input a digit!')
```

```
try:
    被检测的语句块
except Exception:
    异常处理语句块
```

try-except语句


```
try:
    num1 = int(input('Enter the first number: '))
    num2 = int(input('Enter the second number: '))
    print(num1 / num2)
except ZeroDivisionError:
    print('The second number cannot be zero!')
```

多个except子句


```
try:
    num1 = int(input('Enter the first number: '))
    num2 = int(input('Enter the second number: '))
    print(num1 / num2)
except ValueError:
    print('Please input a digit!')
except ZeroDivisionError:
    print('The second number cannot be zero!')
```

一个except块捕捉多个异常


```
try:
    num1 = int(input('Enter the first number: '))
    num2 = int(input('Enter the second number: '))
    print(num1 / num2)
except (ValueError, ZeroDivisionError):
    print('Invalid input!')
```

空except子句


```
try:
    num1 = int(input('Enter the first number: '))
    num2 = int(input('Enter the second number: '))
    print(num1 / num2)
except:
    print('Something went wrong!')
```

一了百了：except:

as子句


```
try:
    num1 = int(input('Enter the first number: '))
    num2 = int(input('Enter the second number: '))
    print(num1 / num2)
except Exception as err:
    print('Something went wrong!')
    print(err)
```

as子句

```
try:
    被检测的语句块
except 异常类名 as 错误原因名:
    异常处理语句块
    print(错误原因名)
```

else子句


```
try:
    num1 = int(input('Enter the first number: '))
    num2 = int(input('Enter the second number: '))
    print(num1 / num2)
except(ValueError, ZeroDivisionError):
    print('Invalid input!')
else:
    print('Aha, I am smart.')
```

```
Enter the first number: 3
Enter the second number: 5
0.6
Aha, I am smart.
```

加入循环

File

```
while True:
    try:
        num1 = int(input('Enter the first number: '))
        num2 = int(input('Enter the second number: '))
        print(num1 / num2)
    except ValueError:
        print('Please input a digit!')
    except ZeroDivisionError:
        print('The second number cannot be zero!')
    else:
        break
```

```
Enter the first number: a
Please input a digit!
Enter the first number: 3
Enter the second number: 0
The second number cannot be zero!
Enter the first number: 3
Enter the second number: 5
0.6
```

break语句的位置

File

```
while True:
    try:
        num1 = int(input('Enter the first number: '))
        num2 = int(input('Enter the second number: '))
        print(num1 / num2)
        break
    except ValueError:
        print('Please input a digit!')
    except ZeroDivisionError:
        print('The second number cannot be zero!')
```

改写

break语句的位置


```
while True:
    try:
        num1 = int(input('Enter the first number: '))
        num2 = int(input('Enter the second number: '))
        print(num1 / num2)
        break
    except Exception as err:
        print(err)
```

break语句的位置

File

```
aList = [1, 2, 3, 4, 5]
i = 0
while True:
    try:
        print(aList[i])
    except IndexError:
        print('index error')
        break
    else:
        i += 1
```

finally子句

File

```
def finallyTest():
    try:
        x = int(input('Enter the first number: '))
        y = int(input('Enter the second number: '))
        print(x / y)
        return 1
    except Exception as err:
        print(err)
        return 0
    finally:
        print('It is a finally clause.')
result = finallyTest()
print(result)
```

```
Enter the first number: 3
Enter the second number: 5
0.6
It is a finally clause.
1

Enter the first number: 3
Enter the second number: 0
division by zero
It is a finally clause.
0
```

筛选数字

编写程序，输入字符串（可能分成若干段），将字符串中所有的数字字符串(除去尾部标点符号外可构成一个整数或浮点数，例如100times不是，句子中的56.78.23不是，句尾的45.78.是，或是一个单独的数字)转换成浮点数并输出。如果没有数字字符串，则输出‘Not Found!'。字符串中可能包括以下标点符号：","、"."、“”、“?”和"!"，标点符号不会连续出现。

```
def isfloat(s):
    if s[-1] == '.':
        s = s.strip('.')
    try:
        float(s)
    except ValueError:
        return False
    return float(s)

if __name__ == "__main__":
    text = input()
    for ch in text:
        if ch in ',?"!':
            text = text.replace(ch, '')
    words = text.split()
    found = 0
    for word in words:
        if isfloat(word) != False:
            print(isfloat(word))
            found = 1
    if found == 0:
        print('Not Found!')
```

人工智能程序设计

3 上下文管理器和WITH语句

上下文管理器（Context Manager）和with语句


```
try:
    f = open('data.txt')
    for line in f:
        print(line, end = '')
except IOError:
    print('Cannot open the file!')
finally:
    f.close()
```

上下文管理器（Context Manager）和with语句

上下文管理器

- 定义和控制代码块执行前的准备动作及执行后的收尾动作
- 通过with语句在支持上下文管理协议的对象（如文件对象）上方便地进行使用


```
with open('data.txt') as f:
    for line in f:
        print(line, end='')
```


with 上下文管理表达式 **as** 变量:
    语句序列

文件异常处理

```
try:
    with open(r'd:\自己的文件目录\test.txt') as fp:
        ... # 各种文件处理
except IOError as err:
    print(err)
```

人工智能程序设计

4 RAISE语句

raise语句

- for语句的实现

```
lst = [1,2,3,4]
it = iter(lst)
while True:
    try:
        x = next(it)
        print(x) # 或其他操作
    except StopIteration:
        break
```

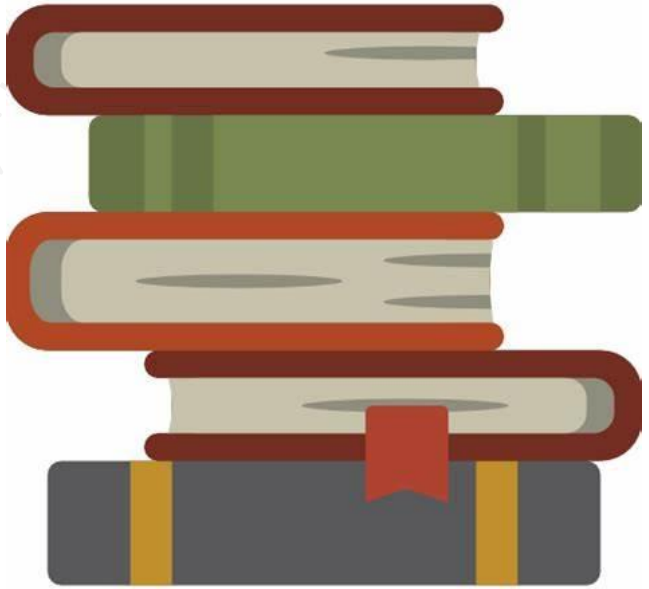
- sqrt()函数的实现

```
def sqrt(x):
        if not isinstance(x, (int, float)):
                raise TypeError('must be real number, not str')
        elif x < 0:
                raise ValueError('math domain error')
        计算平方根的功能模块
```

面向对象和异常示例—栈的实现

用list类实现一个栈（stack）

栈方法	列表实现
S.push(e)	L.append(e)
S.pop()	L.pop()
S.top()	L[-1]
S.is_empty()	len(L)==0
len(S)	len(L)



From《数据结构与算法 Python语言实现》

```
class Empty(Exception):
    pass

class Stack:
    def __init__(self):
        self.data = []
    def length(self):
        return len(self.data)
    def is_empty(self):
        return len(self.data) == 0
    def push(self, e):
        self.data.append(e)
    def top(self):
        if self.is_empty():
            raise Empty('Stack is empty!')
        return self.data[-1]
    def pop(self):
        if self.is_empty():
            raise Empty('Stack is empty!')
        return self.data.pop()
```

```
S = Stack()
S.push(3)
S.push(5)
S.push(8)
print(S.length())
print(S.top())
print(S.length())
print(S.pop())
print(S.length())
```

M1.6 小结

01 异常

02 try-except语句

03 try-finally语句

04 with语句

05 raise语句