异常处理机制

礼 欣

北京理工大学

异常处理机制的引入

- 如果处理错误或特殊情况的分支语句过多，那么处理正常情况的主程序就会变得不清晰易读、
- 以前面讲述的二次方程求解为例

```
discRt = otherSqrt(b * b - 4 * a * c)
    if discRt < 0:
        print "No real roots."
    else:
        ...
```

- 引入异常处理机制来解决程序运行时的错误，而不是显式检查算法的每一步是否成功。

python 语言程序设计

异常处理语句

- Python使用try…except…来进行异常处理，基本格式如下：

```
try:
    <body>
except <ErrorType1>:
    <handler1>
except <ErrorType2>:
    <handler2>
except:
    <handler0>
```

- 当Python解释器遇到一个try语句，它会尝试执行try语句体<body>内的语句
  - 如果没有错误，控制转到try-except后面的语句
  - 如果发生错误，Python解释器会寻找一个符合该错误的异常语句，然后执行处理代码

TryException.py

```
def main():
    try:
        number1, number2 = eval(input("Enter two numbers, separated by a comma"))
        result = number1 / number2

    except ZeroDivisionError:
        print("Division by zero!")
    except SyntaxError:
        print("A comma may be missing in the input")
    except:
        print("Something wrong in the input")
    else:
        print("No exceptions, the result is", result)
    finally:
        print("executing the final clause")
main()
```

python 语言程序设计

TryException.py执行

- 下面是带错误处理的运行实例：

```
>>> TryException.main()
Enter two numbers, separated by a comma:3,4
No exceptions, the result is 0.75
executing the final clause
>>> TryException.main()
Enter two numbers, separated by a comma:2,0
Division by zero!
executing the final clause
```

python 语言程序设计

二次方程求解（版本5）

```
# quadratic5.py
import math
def main():
    print("This program finds the real solutions to a quadratic\n")
    try:
        a, b, c = input("Please enter the coefficients (a, b, c): ")
        discRoot = math.sqrt(b * b - 4 * a * c)
        root1 = (-b + discRoot) / (2 * a)
        root2 = (-b - discRoot) / (2 * a)
        print ("\nThe solutions are:", root1, root2)
    except ValueError:
        print ("\nNo real roots")
main()
```

python 语言程序设计

版本5执行

- 下面是带错误处理的运行实例：

```
>>> quadratic5.main()
This program finds the real solutions to a quadratic

Please enter the coefficients (a, b, c): 1,2,3

No real roots
>>>
```

- Try…except可以捕捉任何类型的错误
  - 对于二次方程，还会有其他可能的错误，如：输入非数值类型 (NameError)，输入无效的表达式(SyntaxError)等。此时可以用一个try语句配多个except来实现。

二次方程求解（版本6）

```
# quadratic6.py
import math
def main():
    print("This program finds the real solutions to a quadratic.\n")
    try:
        a, b, c = eval(input("Please enter the coefficients (a, b, c): "))
        discRoot = math.sqrt(b * b - 4 * a * c)
        root1 = (-b + discRoot) / (2 * a)
        root2 = (-b - discRoot) / (2 * a)
        print("\nThe solutions are:", root1, root2)
    except ValueError as excObj:
        if str(excObj) == "math domain error":
            print("No Real Roots.")
        else:
            print("You didn't give me the right number of coefficients.")
    except NameError:
        print("\nYou didn't enter three numbers.")
    except TypeError:
        print("\nYour inputs were not all numbers.")
    except SyntaxError:
        print("\nYour input was not in the correct form. Missing comma?")
    except:
        print("\nSomething went wrong, sorry!")
main()
```

python 语言程序设计