vxchi@ntt.edu.vn

LAB 5: Exception – Handling, Tập tin(file) trong Python

a. Xử lý ngoại lệ trong Python

Ngoại lệ có thể là bất kỳ điều kiện bất thường nào trong chương trình mà phá vỡ luồng thực thi chương trình đó. Bất cứ khi nào một ngoại lệ xuất hiện, mà không được xử lý, thì chương trình ngừng thực thi và vì thế code không được thực thi.

*Nếu thấy bất cứ code nào là khả nghi (có thể gây ra ngoại lệ) thì có thể phòng thủ chương trình bằng cách đặt các khối code khả nghi này trong một khối **try**. Khối try này được theo sau bởi lệnh **except**. Sau đó, nó được theo sau bởi các lệnh mà xử lý vấn đề đó.*

Cú pháp:

```
try:
   Thực hiện khối lệnh;
   Có thể tạo exception;
   ......................
except ExceptionI:
   Nếu có Exception I, thực thi khối lệnh này
except ExceptionII:
   Nếu có ExceptionII, thực thi khối lệnh
này   ......................
else:
   Nếu không có exception nào thì thực thi tại đây
```

Ví dụ:
Mở một file để ghi trong khi không có quyền ghi, do đó nó sẽ tạo một exception:

```
try:
   fh = open("testfile", "r")
   fh.write("Day la mot kiem tra nho ve xu ly ngoai le!!")
except IOError:
   print "Error: Khong tim thay file"
else:
   print "Thanh cong ghi noi dung vao file"
```

Ví dụ 2:

```
try:
   fh = open("testfile", "w")
   fh.write("Day la mot kiem tra nho ve xu ly ngoai le!!")
```

```
finally:
    print "Error: Khong tim thay file"
```

b. Làm việc với File:

❖ Open()

*Để mở một file, Python cung cấp hàm **open()**. Nó trả về một đối tượng file mà được sử dụng với các hàm khác. Với file đã mở, bạn có thể thực hiện các hoạt động như đọc, ghi mới, ghi thêm ... trên file đó.*

- Cú pháp:

```
file object = open(file_name [, access_mode][, buffering])
```

Trong đó:

- ***filename**: Đối số file_name là một giá trị chuỗi chứa tên của các file.*
- ***access_mode**: Các access_mode xác định các chế độ của file được mở ra như read, write, append,.. Đây là thông số tùy chọn và chế độ truy cập file mặc định là read (r).*
- ***buffering:** Nếu buffer được thiết lập là 0, nghĩa là sẽ không có trình đệm nào diễn ra.*
  - *Nếu xác định là 1, thì trình đệm dòng được thực hiện trong khi truy cập một File.*
  - *Nếu là số nguyên lớn hơn 1, thì hoạt động đệm được thực hiện với kích cỡ bộ đệm đã cho.*
  - *Nếu là số âm, thì kích cỡ bộ đệm sẽ là mặc định.*

Ví dụ:

```
file = open("data.txt", "wb")
print "Tên của file là: ", file.name
print "File có đóng hoặc không?: ", file.closed
print "Chế độ mở file : ", file.mode
```

❖ Close()

*Khi thực hiện xong các hoạt động trên file thì cuối cùng cần đóng file đó. Python tự động đóng một file khi đối tượng tham chiếu của một file đã được tái gán cho một file khác. Tuy nhiên, sử dụng phương thức **close()** để đóng một file vẫn tốt hơn.*

```
Cú pháp:
fileObject.close()
```

Ví dụ:

```
# Mở file
file = open("plc.txt", "r")
# Đóng file
file.close()
```

❖ **Phương thức read()**

Cú pháp:

fileObject.readline()

Ngoài ra còn có :

Ghi File: (fileObject.write()

Thay thế file: os.rename("<tên file hiện tại>","<tên file mới>")

THỰC HÀNH

Câu 1: Đọc 1 file cho trước và in ra màn hình 5 dòng cuối cùng

```
# Mở file
f = open("data.txt", "r", encoding = "utf-8")
# Đọc dòng trong file
a = tuple(f.readlines())
#In ra 5 dòng cuối
for i in a[len(a)-5:]:
    print(i)
```

Câu 2: Đọc 1 file, tìm và in ra nội dung của dòng dài nhất trong file đó

Câu 3: Đọc 1 file, tìm và in ra từ dài nhất trong file

Câu 4: Đọc 1 file, thống kê và in ra tất cả các chữ cái có trong file và số lần xuất hiện của các chữ đó

Câu 5: Đọc 1 file, thống kê và in ra tần xuất xuất hiện của tất cả các từ trong file.