

Hướng dẫn xử lý exception trong Python – Phần I

Giáo viên: Nguyễn Hùng Cường

Exception: Là những lỗi xảy ra trong quá trình thực thi một chương trình, khiến chương trình bị dừng một cách đột ngột không như mong muốn. Exception khiến thay đổi luồng thực thi bình thường của chương trình.

Khi một đoạn mã Python phát sinh ngoại lệ, ta cần xử lý ngoại lệ, nếu không chương trình sẽ bị dừng một cách đột ngột, không mong muốn.

Cơ chế xử lý ngoại lệ của Python được cài đặt thông qua việc sử dụng các khối lệnh: try, except và finally.

Khối try: Dùng để thử lỗi. Khối try sẽ chứa các đoạn mã có khả năng gây ra lỗi.

Khối except: Dùng để xử lý lỗi. Khối except sẽ chứa các đoạn mã xử lý lỗi được phát sinh trong khối try. Khối except có ý nghĩa giống như khối catch ở các ngôn ngữ khác như: Java, C#.

Khối finally: Dùng để chứa các đoạn mã sẽ được thực thi cho dù chương trình có lỗi hay không.

Khối này thường dùng để chứa các đoạn lệnh giải phóng tài nguyên sau khi thực thi xong chương trình, VD như đóng file, đóng kết nối đến CSDL v.v...

Cơ chế xử lý lỗi của Python như sau:

Nếu chương trình có lỗi, khối try sẽ dừng lại, và chương trình sẽ chuyển đến khối except để thực thi và xử lý lỗi. Nếu chương trình có nhiều khối except, thì từng khối sẽ lần lượt được đánh giá theo thứ tự từ trên xuống dưới, khối nào xử lý được lỗi thì sẽ được thực thi.

Nếu chương trình không có lỗi, khối try sẽ được thực thi toàn bộ, chương trình không thực thi khối except.

Ta có thể sử dụng nhiều khối except sau khối try, trong trường hợp ta muốn xử lý nhiều loại lỗi khác nhau.

Ví dụ:

Trong ví dụ này, ta in ra giá trị của biến age trong khối try. Biến age chưa được khai báo, vì vậy khi sử dụng sẽ có lỗi. Trong chương trình, ta khai báo 2 khối except, một khối xử lý lỗi về tên

biến (NameError), một khối xử lý lỗi nói chung.

```
try:
    print(age)
except NameError:
    print("Biến age chưa được định nghĩa")
except:
    print("Có lỗi xảy ra trong chương trình !")
```

Khi thực thi chương trình, ta thấy chương trình hiện thông báo lỗi như ta mong muốn. Khối except đầu tiên được thực thi, vì ta chưa khai báo biến.

```
E:\DemoPythonBasics\venv\Scripts\python.exe
Biến age chưa được định nghĩa

Process finished with exit code 0
```

Nếu ta thay đổi lại chương trình, ta sẽ thấy thông báo lỗi khác. Ở đây ta đã sửa lại chương trình, ta khai báo biến age, rồi in ra giá trị của biến age. Tuy nhiên, trong bài này, ta đã cộng chuỗi sai, vì biến age thuộc kiểu int, lẽ ra ta phải chuyển thành kiểu string.

```
try:
    age = 5
    print("age = " + age)
except NameError:
    print("Biến age chưa được định nghĩa")
except:
    print("Có lỗi xảy ra trong chương trình !")
```

Ta thấy chương trình đã không thực thi khối except đầu tiên (vì không có lỗi này), mà chương trình đã thực thi khối except thứ hai. Kết quả được hiển thị như hình bên dưới.

```
E:\DemoPythonBasics\venv\Scripts\python.exe
Có lỗi xảy ra trong chương trình !

Process finished with exit code 0
```

Ngoài ra, ta có thể sử dụng từ khóa else để định nghĩa một khối lệnh sẽ được thực thi nếu

không có lỗi phát sinh.

Ví dụ:

Trong ví dụ này, ta đã sử dụng khối try..except để minh họa việc xử lý lỗi. Tuy nhiên, trong ví dụ này, khối try không có lỗi, vì vậy chương trình không thực thi khối except, mà sẽ thực thi khối else.

```
try:
    print("To be or not to be")
except:
    print("Co loi xay ra !")
else:
    print("Ket thuc chuong trinh !")
```

Kết quả của chương trình hiển thị khối else đúng như ta mong muốn.

```
E:\DemoPythonBasics\venv\Scripts\python.exe
To be or not to be
Ket thuc chuong trinh !
```

Ví dụ minh họa về khối finally.

Trong bài này, ta sử dụng khối try..except. Khối try sẽ có lỗi vì ta in ra một biến chưa được khai báo. Chương trình sẽ thực thi khối except, sau đó sẽ thực thi khối finally, vì khối finally sẽ luôn được thực hiện, bất kể chương trình có xảy ra lỗi hay không.

```
try:
    print(x)
except:
    print("Co loi xay ra trong chuong trinh !")
finally:
    print("Thuc thi khoi finally !")
```

Kết quả của chương trình hiển thị đúng như ta mong muốn:

```
E:\DemoPythonBasics\venv\Scripts\python.exe  
Co loi xay ra trong chuong trinh !  
Thuc thi khoi finally !
```