



Bài 5

Xử Lý Ngoại Lệ



Chủ đề

- Định nghĩa ngoại lệ
- Hiểu được mục đích của việc xử lý ngoại lệ
- Hiểu được các kiểu ngoại lệ khác nhau trong Java
- Mô tả mô hình xử lý ngoại lệ



Giới thiệu về ngoại lệ

- Là một kiểu lỗi đặc biệt
- Nó xảy ra trong thời gian thực thi đoạn lệnh
- Thông thường các điều kiện thực thi chương trình gây ra ngoại lệ
- Nếu các điều kiện này không được quan tâm, thì việc thực thi có thể kết thúc



Mục đích của việc xử lý ngoại lệ

- Giảm thiểu việc kết thúc bất thường của hệ thống và của chương trình.
- Ví dụ, thao tác xuất/nhập trong một tập tin, nếu việc chuyển đổi kiểu dữ liệu không thực hiện đúng, một ngoại lệ sẽ xảy ra và chương trình bị hủy mà không đóng tập tin. Lúc đó tập tin sẽ bị hư hại và các nguồn tài nguyên được cấp phát cho tập tin không được thu hồi lại cho hệ thống.



Xử lý ngoại lệ

- Khi một ngoại lệ xảy ra, đối tượng tương ứng với ngoại lệ đó sẽ được tạo ra.
- Đối tượng này sau đó được truyền tới phương thức nơi mà ngoại lệ xảy ra.
- Đối tượng chứa các thông tin chi tiết về ngoại lệ. Thông tin này có thể được lấy và xử lý.
- Lớp 'throwable' mà Java cung cấp là lớp trên nhất của lớp ngoại lệ.



Mô hình xử lý ngoại lệ

- Mô hình được biết đến là mô hình 'catch and throw'
- Khi một lỗi xảy ra, ngoại lệ sẽ được chặn và được vào một khối.
- Từ khóa để xử lý ngoại lệ:
 - try
 - catch
 - throw
 - throws
 - finally



Cấu trúc của mô hình xử lý ngoại lệ

- **Cú pháp**

```
try { .... }
```

```
catch(Exception e1) { .... }
```

```
catch(Exception e2) { .... }
```

```
catch(Exception eN) { .... }
```

```
finally { .... }
```



Mô hình 'Catch and Throw' nâng cao

- Người lập trình chỉ quan tâm tới các lỗi khi cần thiết.
- Một thông báo lỗi có thể được cung cấp trong exception-handler.



Khối 'try' và 'catch'

- Được sử dụng để thực hiện trong mô hình 'catch and throw' của xử lý ngoại lệ.
- Khối lệnh 'try' gồm tập hợp các lệnh thực thi
- Một phương thức mà có thể bắt ngoại lệ, cũng bao gồm khối lệnh 'try'.
- Một hoặc nhiều khối lệnh 'catch' có thể tiếp theo sau một khối lệnh 'try'
- Khối lệnh 'catch' này bắt ngoại lệ trong khối lệnh 'try'.



Khởi lệnh 'try' và 'catch' Blocks (tt)

- Để bắt bất kỳ loại ngoại lệ nào, ta phải chỉ ra kiểu ngoại lệ là 'Exception'
catch(Exception e)
- Khi ngoại lệ bị bắt không biết thuộc kiểu nào, chúng ta có thể sử dụng lớp 'Exception' để bắt ngoại lệ đó.
- Lỗi sẽ được truyền thông qua khối lệnh 'try catch' cho tới khi chúng bắt gặp một 'catch' tham chiếu tới nó, hoặc chương trình sẽ bị kết thúc



Khối lệnh chứa nhiều Catch

- Các khối chứa nhiều 'catch()' xử lý các kiểu ngoại lệ khác nhau một cách độc lập.
- Ví dụ

```
try
{ doFileProcessing();
  displayResults(); }
catch(LookupException e)
{ handleLookupException(e); }
catch(Exception e)
{ System.err.println("Error:"+e.printStackTrace()); }
```



Khối lệnh chứa nhiều Catch (tt)

- Khi sử dụng các 'try' lồng nhau, khối 'try' bên trong được thi hành đầu tiên
- Bất kỳ ngoại lệ nào bị chặn trong khối lệnh 'try' sẽ bị bắt giữ trong khối lệnh 'catch' tiếp ngay sau.
- Nếu khối lệnh 'catch' thích hợp không được tìm thấy, thì các khối 'catch' của khối 'try' bên ngoài sẽ được xem xét
- Ngược lại, Java Runtime Environment sẽ xử lý ngoại lệ.



Tóm tắt

- Một ngoại lệ xảy ra vào thời gian thực thi tuần tự các câu lệnh
- Việc xử lý ngoại lệ bằng cách sử dụng khối lệnh `'try'` và `'catch'`.
- Khối lệnh nhiều catch.