Từ khóa throws trong Java

Từ khóa **throws** trong Java được sử dụng để khai báo một Exception. Nó cung cấp một thông tin tới Lập trình viên rằng có thể xuất hiện một Exception, để họ nên cung cấp một code để xử lý ngoại lệ để duy trì luồng chuẩn của chương trình.

Xử lý ngoại lệ (Exception Handling) chủ yếu được sử dụng để xử lý các Checked Exception. Nếu có thể xuất hiện bất cứ Unchecked Exception nào chẳng hạn như NullPointerException, thì đó là lỗi của lập trình viên vì họ đã không thực hiện kiểm tra code trước khi sử dụng.

Cú pháp của từ khóa throws trong Java

```
kieu_tra_ve ten_phuong_thuc() throws ten_lop_exception{
//phan code cua phuong thuc
}
```

Câu hỏi: Kiểu Exception nên được khai báo?

Chỉ dành cho Checked Exception, bởi vì:

- **Unchecked Exception**: dưới sự điều khiển của bạn bằng việc kiểm tra và sửa lỗi code.
- **Error**: Vượt quá tầm kiểm soát của bạn, ví dụ: bạn không thể làm điều gì nếu xuất hiện VirtualMachineError hoặc StackOverflowErrror.

Lợi thế của từ khóa throws trong Java

Bây giờ, bởi sử dụng từ khóa throws, các Checked Exception có thể được lan truyền (trong Call Stack). Nó cung cấp thông tin tới người gọi phương thức về Exception đó.

Ví dụ về throws trong Java

Chương trình Java sau sử dụng từ khóa throws để minh họa việc Checked Exception có thể được lan truyền (trong Call Stack).

```
import java.io.IOException;
class Testthrows1{
  void m()throws IOException{
    throw new IOException("device error");//checked exception
  }
```

```
  void n()throws IOException{
    m();
  }
  void p(){
   try{
    n();
   }catch(Exception e){System.out.println("Exception duoc xu ly");}
  }
  public static void main(String args[]){
   Testthrows1 obj=new Testthrows1();
   obj.p();
   System.out.println("Luong chuan...");
  }
}
```

Chương trình Java trên sẽ cho kết quả:

```
Exception duoc xu ly
Luong chuan...
```

Qui tắc: Nếu bạn đang gọi một phương thức mà khai báo một exception, bạn phải hoặc bắt hoặc khai báo exception đó.

Có hai trường hợp:

- Trường hợp 1: Bạn bắt exception (sử dụng try-catch để xử lý exception đó)
- Trường hợp 2: Bạn khai báo exception (xác định từ khóa throws với phương thức đó)

Trường hợp 1: Bạn xử lý exception

Trong trường hợp này, code sẽ được thực thi tốt dù cho exception có xuất hiện trong chương trình hay không.

```
import java.io.*;
class M{
 void method()throws IOException{
  throw new IOException("device error");
```

```
 }
}
public class Testthrows2{
   public static void main(String args[]){
    try{
     M m=new M();
     m.method();
    }catch(Exception e){System.out.println("Exception duoc xu ly");}

    System.out.println("Luong chuan...");
  }
}
```

Chương trình Java trên sẽ cho kết quả:

```
Exception duoc xu ly
Luong chuan...
```

Trường hợp 2: Bạn khai báo exception

- Nếu exception không xuất hiện, code sẽ được thực thi tốt.
- Nếu exception xuất hiện, một exception sẽ được ném tại runtime bởi vì throws không xử lý exception đó.

A. Chương trình ví dụ nếu exception không xuất hiện

```
import java.io.*;
class M{
 void method()throws IOException{
  System.out.println("Thiet bi hoat dong tot");
 }
}
class Testthrows3{
   public static void main(String args[])throws IOException{//Khai bao exception
     M m=new M();
     m.method();
```

```
    System.out.println("Luong chuan...");
  }
}
```

Chương trình Java trên sẽ cho kết quả:

```
Thiet bi hoat dong tot
Luong chuan...
```

B. Chương trình ví dụ nếu exception xuất hiện

```
import java.io.*;
class M{
 void method()throws IOException{
  throw new IOException("device error");
 }
}
class Testthrows4{
   public static void main(String args[])throws IOException{//Khai bao exception
     M m=new M();
     m.method();

    System.out.println("Luong chuan...");
  }
}
```

Chương trình Java trên sẽ cho Runtime Error.

Câu hỏi: Chúng ta có thể tái ném một exception không?

Có, bằng cách ném cùng exception đó trong khối catch.

Phân biệt throw và throws trong Java

Có nhiều điểm khác nhau giữa hai từ khóa throw và throws. Bảng dưới liệt kê các điểm khác nhau này.

throw	throws
Từ khóa throw được sử dụng để ném tường minh một exception	Từ khóa throws được sử dụng để khai báo một exception
Checked Exception không thể được lan truyền chỉ bởi sử dụng throw	Checked Exception không thể được lan truyền với throws
Throw được theo sau bởi một instance	Throws được theo sau bởi một lớp
Throw được sử dụng bên trong một phương thức	Throws được sử dụng với khai báo phương thức
Bạn có thể ném nhiều exception	Bạn có thể khai báo nhiều exception, ví dụ public void phuong_thuc()throws IOException,SQLException

Ví dụ về throw trong Java

```
void m(){
throw new ArithmeticException("sorry");
}
```

Ví dụ về throws trong Java

```
void m()throws ArithmeticException{
//Phan code cua phuong thuc
}
```

Ví dụ về throw và throws trong Java

```
void m()throws ArithmeticException{
throw new ArithmeticException("sorry");
}
```