

예외 떠 넘기기, 사용자 정의 예외

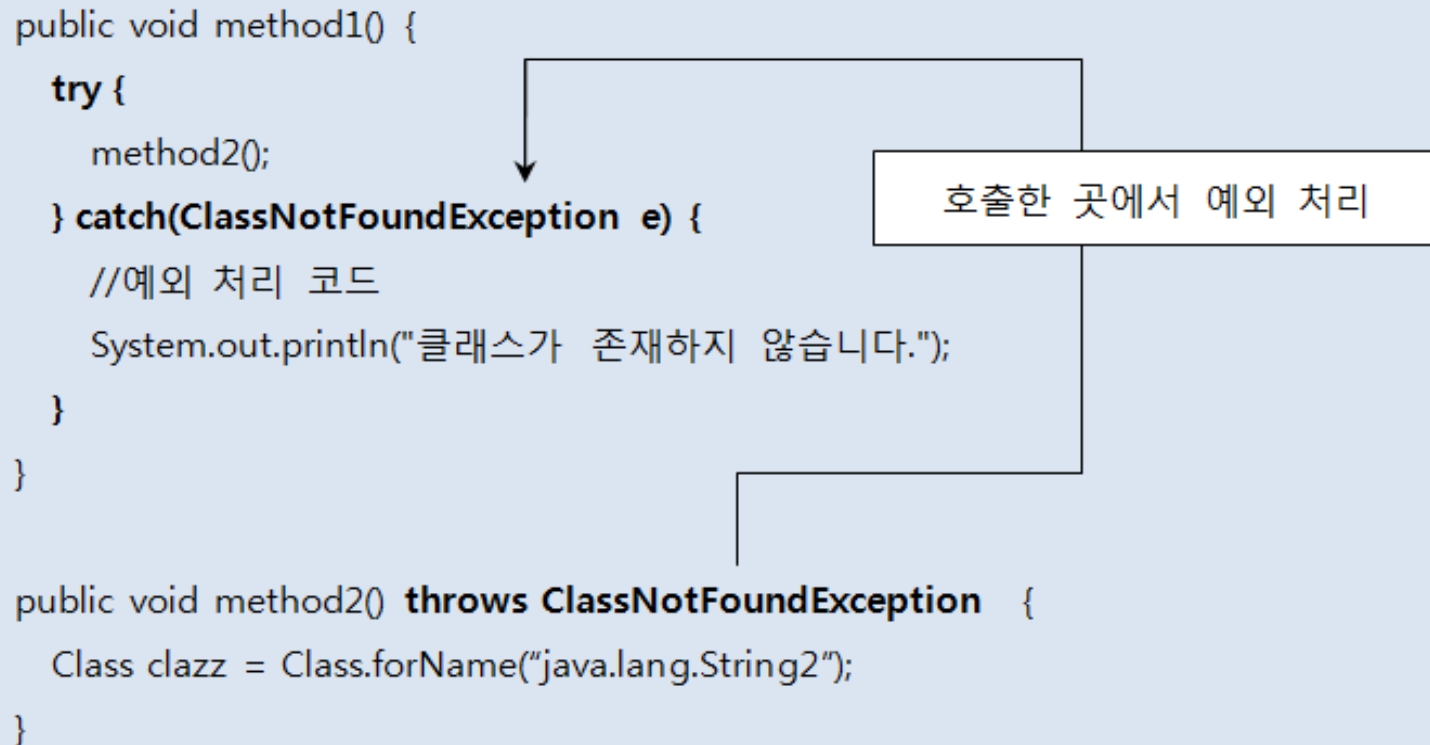
예외 떠 넘기기

❖ throws

- 메소드 선언부 끝에 작성

```
리턴타입 메소드명(매개변수,...) throws 예외클래스 1, 예외클래스 2, ... {  
}
```

- 메소드에서 처리하지 않은 예외를 호출한 곳으로 떠 넘기는 역할



예외 떠 넘기기

❖ 예외 처리 떠넘기기: ThrowsExample.java

```
public class ThrowsExample {  
    public static void main(String[] args) {  
        try {  
            findClass();  
        } catch (ClassNotFoundException e) {  
            System.out.println("클래스가 존재하지 않습니다.");  
        }  
    }  
  
    public static void findClass() throws ClassNotFoundException {  
        Class clazz = Class.forName("java.lang.String2");  
    }  
}
```

사용자 정의 예외

❖ 사용자 정의 예외 클래스 선언

- 자바 표준 API에서 제공하지 않는 예외
- 애플리케이션 서비스와 관련된 예외
 - Ex) 잔고 부족 예외, 계좌 이체 실패 예외, 회원 가입 실패 예외...
- 사용자 정의 예외 클래스 선언 방법

```
public class XXXException extends [ Exception | RuntimeException ] {  
    public XXXException() { }  
    public XXXException(String message) { super(message); }  
}
```

사용자 정의 예외

❖ 사용자 정의 예외 클래스: `BalanceInsufficientException.java`

```
public class BalanceInsufficientException extends Exception {  
    public BalanceInsufficientException() {  
    }  
  
    public BalanceInsufficientException(String message) {  
        super(message);  
    }  
}
```

사용자 정의 예외

❖ 예외 발생 시키기

- 코드에서 예외 발생시키는 법

```
throw new XXXException()  
throw new XXXException("메시지");
```

```
public void method() throws XXXException {  
    throw new XXXException("메시지");  
}
```

- 호출된 곳에서 발생한 예외를 처리하도록

사용자 정의 예외

❖ 사용자 정의 예외 발생시키기: Account.java

```
public class Account {  
    private long balance;  
  
    public Account() {  
    }  
  
    public long getBalance() {  
        return balance;  
    }  
  
    public void deposit(int money) {  
        balance += money;  
    }  
  
    public void withdraw(int money) throws BalanceInsufficientException {  
        if (balance < money) {  
            throw new BalanceInsufficientException("잔고부족:" +  
                (money - balance) + " 모자람");  
        }  
        balance -= money;  
    }  
}
```

사용자 정의 예외

❖ getMessage()

- 예외 발생시킬 때 생성자 매개값으로 사용한 메시지 리턴

```
throw new XXXException("예외 메시지");
```

- 원인 세분화하기 위해 예외 코드 포함(예: 데이터베이스 예외 코드)
- catch() 절에서 활용

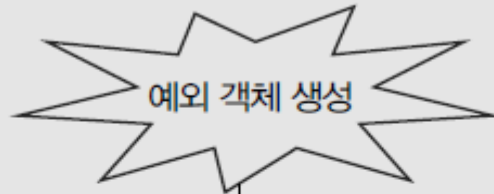
```
} catch(Exception e) {  
    String message = e.getMessage();  
}
```


사용자 정의 예외

❖ `printStackTrace()`

- 예외 발생 코드 추적한 내용을 모두 콘솔에 출력
- 프로그램 테스트하면서 오류 찾을 때 유용하게 활용

```
try {
```



```
    } catch(예외클래스 e) {  
        //예외가 가지고 있는 Message 얻기  
        String message = e.getMessage();  
  
        //예외의 발생 경로를 추적  
        e.printStackTrace();  
    }
```

사용자 정의 예외

❖ 사용자 정의 예외 발생시키기: AccountExample.java

```
public class AccountExample {  
    public static void main(String[] args) {  
        Account account = new Account();  
        // 예금하기  
        account.deposit(10000);  
        System.out.println("예금액: " + account.getBalance());  
        // 출금하기  
        try {  
            account.withdraw(30000);  
        } catch (BalanceInsufficientException e) {  
            String message = e.getMessage();  
            System.out.println(message);  
            System.out.println();  
            e.printStackTrace();  
        }  
    }  
}
```