



1학기 고급자바 실습

week 2-2

김민진(18)

김지희(18)

문의 메일 : genie02166@duksung.ac.kr

Part1. 예외 처리

1) 개념 확인 문제

- Q1. 예외
- Q2. try-catch-finally
- Q3. throws
- Q4. throw
- Q5. 틀린 예외 처리 예시 고르기

2) 발생한 예외에 대한 정보 얻기

- `printStackTrace()`
- `getMessage()`

3) 로그인 기능 구현 예제

- 존재하지 않는 ID 입력 시 예외 발생
- 잘못된 비밀번호 입력 시 예외 발생

1) 개념 확인 문제

- 예외

Q1. 예외에 대한 설명 중 틀린 것은 무엇입니까?

- 1) 예외는 사용자의 잘못된 조작, 개발자의 잘못된 코딩으로 인한 프로그램 오류를 말한다.
- 2) RuntimeException의 하위 예외는 컴파일러가 예외 처리 코드를 체크하지 않는다.
- 3) 예외는 try-catch 블록을 사용해서 처리된다.
- 4) 자바 표준 예외만 프로그램에서 처리할 수 있다.

1) 개념 확인 문제

- try-catch-finally

Q2. try-catch-finally 블록에 대한 설명 중 틀린 것은 무엇입니까?

- 1) try { } 블록에는 예외가 발생할 수 있는 코드를 작성한다.
- 2) catch { } 블록은 try { } 블록에서 발생한 예외를 처리하는 블록이다.
- 3) try { } 블록에서 return문을 사용하면 finally { } 블록은 실행되지 않는다.
- 4) catch { } 블록은 예외의 종류별로 여러 개를 작성할 수 있다.

1) 개념 확인 문제

- throws

Q3. throws에 대한 설명으로 틀린 것은 무엇입니까?

- 1) 생성자나 메소드의 선언 끝 부분에 사용되어 내부에서 발생한 예외를 떠넘긴다.
- 2) throws 뒤에는 떠넘겨야 할 예외를 쉼표(,)로 구분해서 기술한다.
- 3) 모든 예외를 떠넘기기 위해 간단하게 throws Exception으로 작성할 수 있다.
- 4) 새로운 예외를 발생시키기 위해 사용된다.

1) 개념 확인 문제

- throw

Q4. throw에 대한 설명으로 틀린 것은 무엇입니까?

- 1) 예외를 최초로 발생시키는 코드이다.
- 2) 예외를 호출한 곳으로 떠넘기기 위해 메소드 선언부에 작성된다.
- 3) throw로 발생한 예외는 일반적으로 생성자나 메소드 선언부에 throws로 떠넘겨진다.
- 4) throw 키워드 뒤에는 예외 객체 생성 코드가 온다.

1) 개념 확인 문제

- 틀린 예외 처리 예시 고르기

Q5. 다음과 같은 메소드가 있을 때 예외를 잘못 처리한 것은 무엇입니까?

```
public void method1() throws NumberFormatException, ClassNotFoundException { ... }
```

- 1) try { method1 (); } catch (Exception e) { }
- 2) void method2() throws Exception { method1(); }
- 3) try { method1(); } catch (Exception e) { } catch (ClassNotFoundException e) { }
- 4) try { method1(); } catch (ClassNotFoundException e) { } catch (NumberFormatException e) { }

2) 발생한 예외에 대한 정보 얻기

메소드	기능
<code>printStackTrace()</code>	예외발생 당시의 호출스택(Call Stack)에 있었던 메소드의 정보와 예외 메시지를 화면에 출력
<code>getMessage()</code>	발생한 예외클래스의 인스턴스에 저장된 메시지를 얻을 수 있음

2) 발생한 예외에 대한 정보 얻기

ExceptionExample.java

```
1 package week2_2;
2
3 public class ExceptionExample {
4
5     public static void main(String[] args) {
6         System.out.println(1);
7         System.out.println(2);
8         try {
9             System.out.println(3);
10            System.out.println(0/0);    // 예외 발생
11            System.out.println(4);    // 실행되지 않음
12        } catch (ArithmeticException ae) {
13            ae.printStackTrace();
14            System.out.println("예외메시지 : " + ae.getMessage());
15        }
16        System.out.println(6);
17    }
18 }
```

실행 결과

Console Problems Javadoc Declaration

<terminated> ExceptionExample [Java Application] C:\Program Files\Java\jdk-13.0.2\bin\

1

2

3

java.lang.ArithmeticException: / by zero
at week2_2.ExceptionExample.main(ExceptionExample.java:10)

예외메시지 : / by zero

6

3) 로그인 기능 구현 예제

Q. 로그인 기능을 Member 클래스의 login() 메소드에서 구현하려고 합니다.

존재하지 않는 ID를 입력했을 경우 NotExistIDException을 발생시키고,

잘못된 패스워드를 입력했을 경우 WrongPasswordException을 발생시키려고 합니다.

LoginExample의 실행 결과를 보고 빈칸을 채워보세요.

NotExistIDException.java WrongPasswordException.java LoginE

```

1 package week2_2;
2
3 public class NotExistIDException extends Exception {
4
5     public NotExistIDException() {}
6     public NotExistIDException(String message) {
7
8
9
10
11 }
12 }
13

```

1

NotExistIDException.java WrongPasswordException.java LoginExar

```

1 package week2_2;
2
3 public class WrongPasswordException extends Exception {
4
5     public WrongPasswordException() {}
6     public WrongPasswordException(String message) {
7
8
9
10
11 }
12 }
13

```

2

NotExistIDException.java WrongPasswordException.java LoginExample.java

```

1 package week2_2;
2
3 public class LoginExample {
4     public static void main(String[] args) {
5         try {
6             login("white", "12345");
7         } catch (Exception e) {
8             System.out.println(e.getMessage());
9         }
10
11     try {
12         login("blue", "54321");
13     } catch (Exception e) {
14         System.out.println(e.getMessage());
15     }
16
17     public static void login(String id, String password) {
18         //id "blue" NotExistIDException
19         if(!id.equals("blue")) {
20
21
22
23         }
24
25         //password "12345" WrongPasswordException
26         if(!password.equals("12345")) {
27
28
29
30         }
31     }
32 }
33 }
34

```

3

4

5

실행 결과

아이디가 존재하지 않습니다.

패스워드가 틀립니다.

NotExistIDException.java WrongPasswordException.java LoginE

```

1 package week2_2;
2
3 public class NotExistIDException extends Exception {
4
5     public NotExistIDException() {}
6     public NotExistIDException(String message) {
7
8         super(message);
9     }
10 }
11
12 }
13

```

NotExistIDException.java WrongPasswordException.java LoginExar

```

1 package week2_2;
2
3 public class WrongPasswordException extends Exception {
4
5     public WrongPasswordException() {}
6     public WrongPasswordException(String message) {
7
8         super(message);
9     }
10 }
11
12 }
13

```

NotExistIDException.java WrongPasswordException.java LoginExample.java

```

1 package week2_2;
2
3 public class LoginExample {
4     public static void main(String[] args) {
5         try {
6             login("white", "12345");
7         } catch (Exception e) {
8             System.out.println(e.getMessage());
9         }
10
11         try {
12             login("blue", "54321");
13         } catch (Exception e) {
14             System.out.println(e.getMessage());
15         }
16
17         public static void login(String id, String password) {
18             //id "blue" NotExistIDException
19             if(!id.equals("blue")) {
20
21                 throw new NotExistIDException("아이디가 존재하지 않습니다.");
22             }
23
24             //password "12345" WrongPasswordException
25             if(!password.equals("12345")) {
26
27                 throw new WrongPasswordException("패스워드가 틀립니다.");
28             }
29
30         }
31     }
32 }
33
34

```

throws NotExistIDException, WrongPasswordException

throw new NotExistIDException("아이디가 존재하지 않습니다.");

throw new WrongPasswordException("패스워드가 틀립니다.");

실행 결과

아이디가 존재하지 않습니다.

패스워드가 틀립니다.

출석 과제 (3/26 금 오후 11:55 마감)

Q. 다음과 같은 메소드가 있을 때 예외를 잘못 처리한 것은 무엇입니까?

정답의 번호와, 해당 코드를 예외를 올바르게 처리하도록 바꿔서 새로운 코드를 작성해주세요.

```
public void method1() throws NumberFormatException, ClassNotFoundException { ... }
```

- 1) try { method1 (); } catch (Exception e) { }
- 2) void method2() throws Exception { method1(); }
- 3) try { method1(); } catch (Exception e) { } catch (ClassNotFoundException e) { }
- 4) try { method1(); } catch (ClassNotFoundException e) { } catch (NumberFormatException e) { }