

Java



Exception 클래스

익셉션 클래스에 대하여

익셉션의 종류

익셉션 클래스의 사용 방법

익셉션 클래스의 선언 방법



익셉션 클래스

01. 익셉션 클래스에 대하여

익셉션

• 익셉션(exception, 예외)

- 프로그램 실행 중에 발생하는 에러
- try-catch 문에 의해 처리되거나
- 메소드 밖으로 던져짐

```
class MallProgram {  
    public static void main(String args[]) {  
        try {  
            ...  
            StockManager obj = new StockManager();  
            ...  
            obj.updateStock();  
            ...  
        }  
        catch (Exception e) {  
            System.out.println("에러 발생");  
        }  
    }  
}
```

3) catch 블록으로 던져지는 익셉션

2) 메소드를 호출한 곳으로
던져지는 익셉션

```
class StockManager {  
    void updateStock() throws Exception {  
        ...  
        GoodsStock obj = new GoodsStock();  
        ...  
        obj.subtractStock(num);  
        ...  
    }  
}
```

1) 메소드를 호출한 곳으로
던져지는 익셉션

```
class GoodsStock {  
    ...  
    void subtractStock(int amount) throws Exception {  
        if (stockNum < amount)  
            throw new Exception("재고가 부족합니다.");  
        ...  
    }  
}
```

익셉션 클래스

01. 익셉션 클래스에 대하여

익셉션 객체

- 익셉션 객체(exception object) : 익셉션을 표현하는 객체

```
throw new Exception("재고가 부족합니다.");
```

↑
익셉션 객체

- 익셉션을 잡는 catch 절

```
catch (Exception e) {
```

...

```
}
```

↑
변수 타입

↑
변수 이름

익셉션 클래스

● 01. 익셉션 클래스에 대하여

● 익셉션 클래스

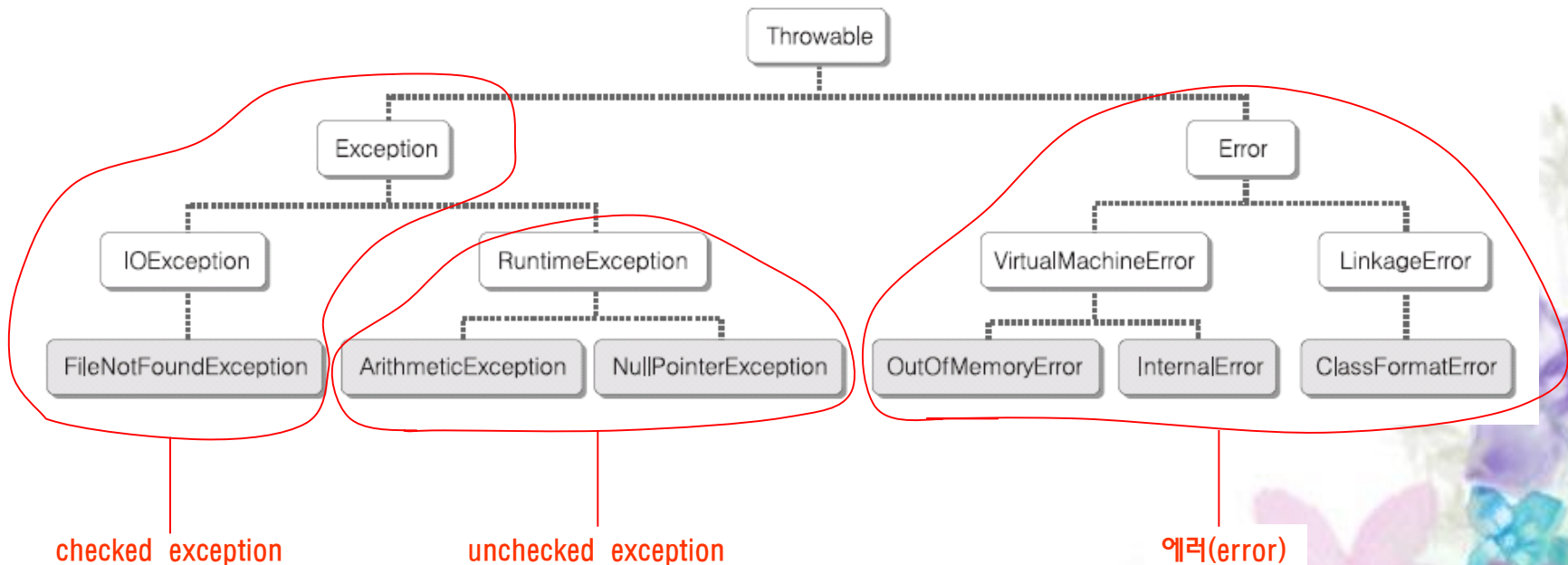
- 익셉션 클래스(exception class) : 익셉션 객체를 만들 수 있는 클래스
 - [예] Exception 클래스
- 익셉션 클래스가 될 수 있는 클래스는?
 - `java.lang.Throwable` 클래스와 서브클래스들

익셉션 클래스

02. 익셉션의 종류

익셉션 클래스

• 익셉션 클래스의 상속 관계



익셉션 클래스

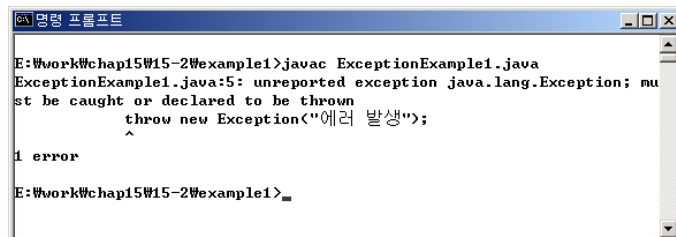
02. 익셉션의 종류

익셉션의 종류

- [예제 15-1] *checked exception*과 *unchecked exception*을 발생하는 프로그램

```
class ExceptionExample1 {  
    public static void main(String args[]) {  
        int sum = 1 + -2;  
        if (sum < 0)  
            throw new Exception("에러 발생");  
        System.out.println(sum);  
    }  
}
```

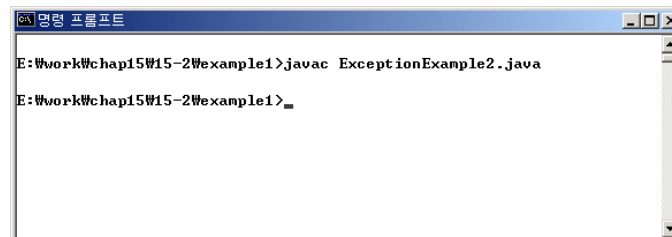
[ExceptionExample1의 컴파일 결과]



```
E:\work\chap15\15-2\example1>javac ExceptionExample1.java  
ExceptionExample1.java:5: unreported exception java.lang.Exception; must  
be caught or declared to be thrown  
    throw new Exception("에러 발생");  
    ^  
1 error  
E:\work\chap15\15-2\example1>
```

```
class ExceptionExample2 {  
    public static void main(String args[]) {  
        int num1 = 3, num2 = 0;  
        int result = num1 / num2;  
        System.out.println(result);  
    }  
}
```

[ExceptionExample2의 컴파일 결과]



```
E:\work\chap15\15-2\example1>javac ExceptionExample2.java  
E:\work\chap15\15-2\example1>
```

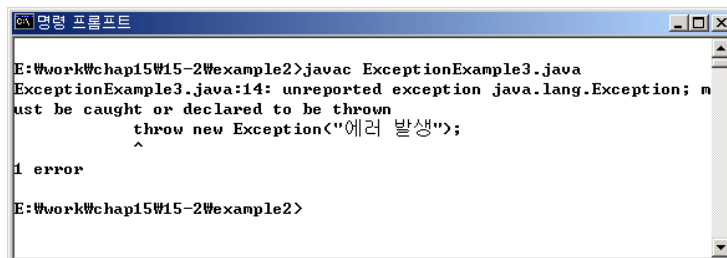
익셉션 클래스

02. 익셉션의 종류

익셉션의 종류

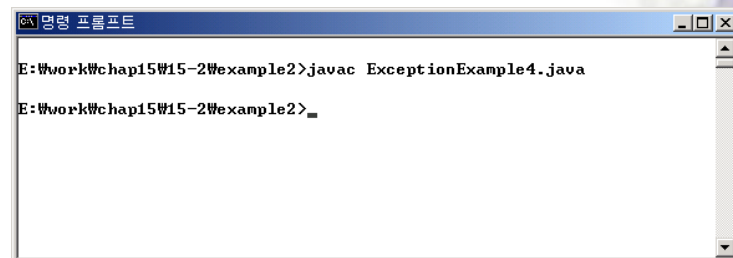
- [예제 15-2] checked exception과 unchecked exception을 던지는 메소드

```
class ExceptionExample3 {  
    public static void main(String args[]) {  
        try {  
            int result = add(1, -2);  
            System.out.println(result);  
        }  
        catch (Exception e) {  
            System.err.println("에러 발생");  
        }  
    }  
    static int add(int a, int b) {  
        int result = a + b;  
        if (result < 0)  
            throw new Exception("에러 발생");  
        return result;  
    }  
}
```



```
E:\work\chap15\15-2\example2>javac ExceptionExample3.java  
ExceptionExample3.java:14: unreported exception java.lang.Exception; must be caught or declared to be thrown  
        throw new Exception("에러 발생");  
                ^  
1 error  
E:\work\chap15\15-2\example2>
```

```
class ExceptionExample4 {  
    public static void main(String args[]) {  
        try {  
            int result = devide(3, 0);  
            System.out.println(result);  
        }  
        catch (ArithmeticException e) {  
            System.err.println("에러 발생");  
        }  
    }  
    static int devide(int a, int b) {  
        int result = a / b;  
        return result;  
    }  
}
```



```
E:\work\chap15\15-2\example2>javac ExceptionExample4.java  
E:\work\chap15\15-2\example2>
```


익셉션 클래스

03. 익셉션 클래스의 사용 방법

익셉션 클래스의 메소드

- 에러 메시지를 가져오는 `getMessage` 메소드

```
String str = e.getMessage();
```



익셉션 객체에 있는
에러 메시지를 리턴하는 메서드

익셉션 클래스

03. 익셉션 클래스의 사용 방법


getMessage 메소드

- [예제 15-3] getMessage 메소드의 사용 예 (1)

```
1  class ExceptionExample5 {  
2      public static void main(String args[]) {  
3          int num1 = 2, num2 = 0;  
4          try {  
5              int result = num1 / num2;  
6              System.out.println(result);  
7          }  
8          catch (ArithmeticException e) {  
9              String str = e.getMessage();  
10             System.err.println(str);  
11         }  
12     }  
13 }
```

ArithmeticException이 발생하면 catch 절로 제어가 넘어갑니다.

ArithmeticException 객체에 있는 에러 메시지를 가져옵니다.



```
명령 프롬프트  
E:\work\chap15\15-3-1>java ExceptionExample5  
by zero  
E:\work\chap15\15-3-1>
```

03. 익셉션 클래스의 사용 방법

getMessage 메소드

- 다음과 같이 생성된 Exception 객체의 에러 메시지는?

throw new Exception();

↑
에러 메시지를 포함하지 않는
Exception 객체를 생성

throw new Exception("잘못된 값입니다.");

↑
에러 메시지를 포함하는
Exception 객체를 생성

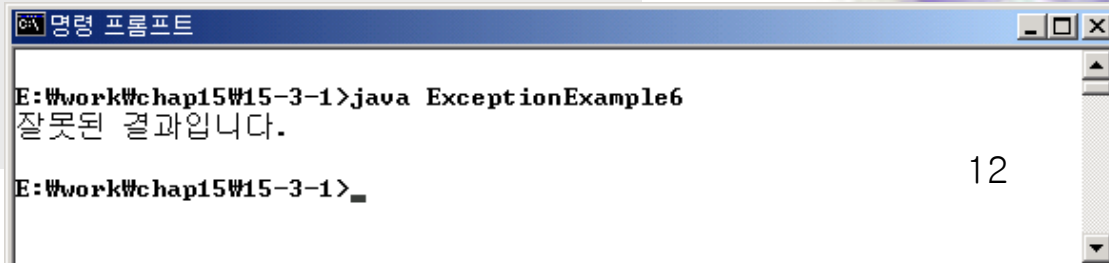
익셉션 클래스

03. 익셉션 클래스의 사용 방법

getMessage 메소드

- [예제 15-4] getMessage 메소드의 사용 예 (2)

```
1  class ExceptionExample6 {
2      public static void main(String args[]) {
3          int num1 = 2, num2 = 3;
4          try {
5              int result = num1 - num2;
6              if (result < 0)
7                  throw new Exception("잘못된 결과입니다."); ----- Exception 객체를 생성합니다.
8              System.out.println(result);
9          }
10         catch (Exception e) {
11             String str = e.getMessage(); ----- Exception 객체에 있는 에러 메시지를 가져옵니다.
12             System.err.println(str);
13         }
14     }
15 }
```



```
명령 프롬프트
E:\work\chap15\15-3-1>java ExceptionExample6
잘못된 결과입니다.
E:\work\chap15\15-3-1>
```

익셉션 클래스

03. 익셉션 클래스의 사용 방법

printStackTrace 메소드

- 상세한 익셉션 정보를 추적하는 `printStackTrace` 메소드

`e.printStackTrace();`



익셉션에 대한 자세한 정보를
출력하는 메소드

익셉션 클래스

03. 익셉션 클래스의 사용 방법

printStackTrace 메소드

- [예제 15-5] printStackTrace 메소드의 사용 예

```
1 class ExceptionExample7 {
2     public static void main(String args[]) {
3         try {
4             int arr[] = new int[0];
5             System.out.println("합계" + getTotal(arr));
6             System.out.println("평균" + getAverage(arr));
7         }
8         catch (Exception e) {
9             e.printStackTrace();
10        }
11    }
12    static double getAverage(int arr[]) throws Exception {
13        if (arr.length == 0)
14            throw new Exception("비어있는 배열입니다.");
15        return getTotal(arr) / arr.length;
16    }
17    static int getTotal(int arr[]) throws Exception {
18        if (arr.length == 0)
19            throw new Exception("비어있는 배열입니다.");
20        int total = 0;
21        for (int num : arr)
22            total += num;
23        return total;
24    }
25 }
```

익셉션이 처리되는 부분

익셉션이 발생하는 부분

익셉션이 발생하는 부분

```
E:\work\chap15\15-3-1>java ExceptionExample7
java.lang.Exception: 비어있는 배열입니다.
    at ExceptionExample7.getTotal(ExceptionExample7.java:19)
    at ExceptionExample7.main(ExceptionExample7.java:5)

E:\work\chap15\15-3-1>
```

03. 익셉션 클래스의 사용 방법

익셉션 파라미터 변수

• 익셉션 파라미터 변수 : `catch` 절에서 익셉션을 받을 때 사용하는 변수

```
catch (ArithmeticException e) {
```

익셉션 파라미터 변수의 타입

익셉션 파라미터 변수

익셉션 클래스

03. 익셉션 클래스의 사용 방법

익셉션 파라미터 변수의 다형성

- [예제 15-6] 익셉션 파라미터 변수의 다형성을 보여주는 프로그램

```
1  import java.io.*;
2  class ExceptionExample8 {
3      public static void main(String args[]) {
4          try {
5              FileReader reader = new FileReader("some.txt");
6              reader.close();
7          }
8          catch (IOException e) {
9              String str = e.getMessage();
10             System.err.println(str);
11         }
12     }
13 }
```

----- FileNotFoundException이
발생하는 부분

명령 프롬프트

E:\work\chap15\15-3-2>java ExceptionExample8
some.txt <지정된 파일을 찾을 수 없습니다>

E:\work\chap15\15-3-2>

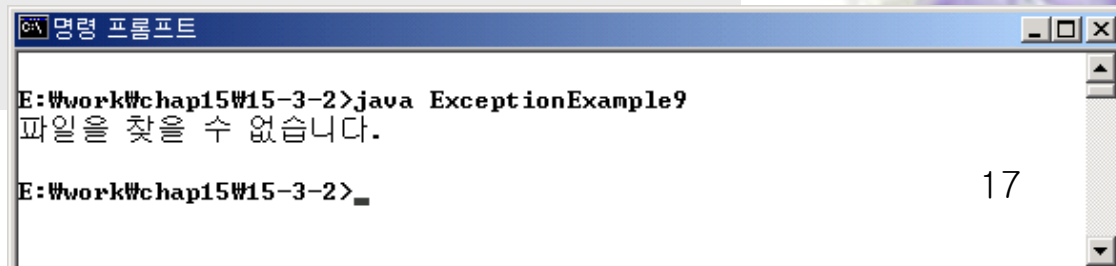
익셉션 클래스

03. 익셉션 클래스의 사용 방법

익셉션 파라미터 변수의 다형성

- [예제 15-7] 상속 관계에 있는 두 종류의 익셉션을 처리하는 try 문의 예

```
1  import java.io.*;
2  class ExceptionExample9 {
3      public static void main(String args[]) {
4          try {
5              FileReader reader = new FileReader("some.txt"); ----- FileNotFoundException이 발생하는 부분
6              reader.close();
7          }
8          catch (FileNotFoundException e) {
9              System.err.println("파일을 찾을 수 없습니다.");
10         }
11         catch (IOException e) {
12             System.err.println("입출력 에러가 발생했습니다.");
13         }
14     }
15 }
```



```
E:\work\chap15\15-3-2>java ExceptionExample9
파일을 찾을 수 없습니다.

E:\work\chap15\15-3-2>
```

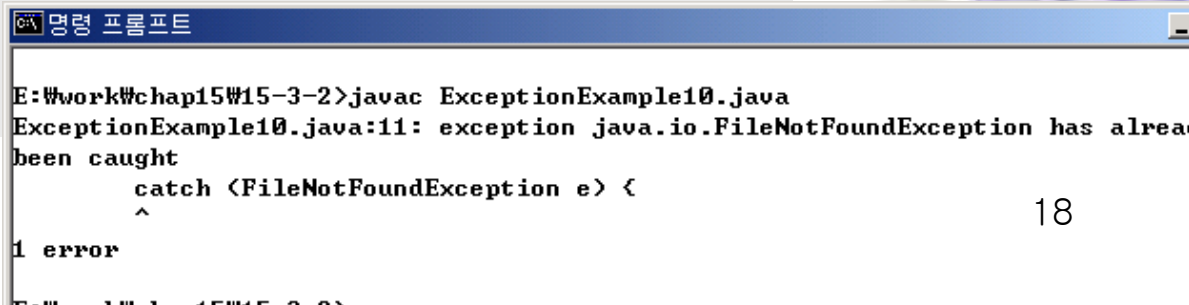
익셉션 클래스

03. 익셉션 클래스의 사용 방법

익셉션 파라미터 변수의 다형성

- [예제 15-8] catch 절의 순서가 잘못된 try 문의 예

```
1  import java.io.*;
2  class ExceptionExample10 {
3      public static void main(String args[]) {
4          try {
5              FileReader reader = new FileReader("some.txt"); ----- FileNotFoundException이 발생하는 부분
6              reader.close();
7          }
8          catch (IOException e) {
9              System.err.println("입출력 에러가 발생했습니다.");
10         }
11         catch (FileNotFoundException e) {
12             System.err.println("파일을 찾을 수 없습니다.");
13         }
14     }
15 }
```



```
E:\work\chap15\15-3-2>javac ExceptionExample10.java
ExceptionExample10.java:11: exception java.io.FileNotFoundException has already
been caught
    catch (FileNotFoundException e) {
    ^
1 error
```

04. 익셉션 클래스의 선언 방법

익셉션 클래스의 선언 방법

- checked exception으로 선언하는 것이 무난
 - 특별히 상속할 클래스가 없으면 Exception 클래스를 바로 상속받아도 됨

```
class InvalidInputException extends Exception {  
  
}
```

Exception 클래스의
서브클래스로 선언 합니다.

04. 익셉션 클래스의 선언 방법

익셉션 클래스의 선언 방법

- [예제 15-9] 익셉션 클래스의 선언 예

```
1    class InvalidInputException extends Exception {  
2        InvalidInputException() {  
3            super("잘못된 입력입니다.");  
4        }  
5    }
```

슈퍼클래스의 생성자를 호출하여
에러 메시지를 저장합니다.

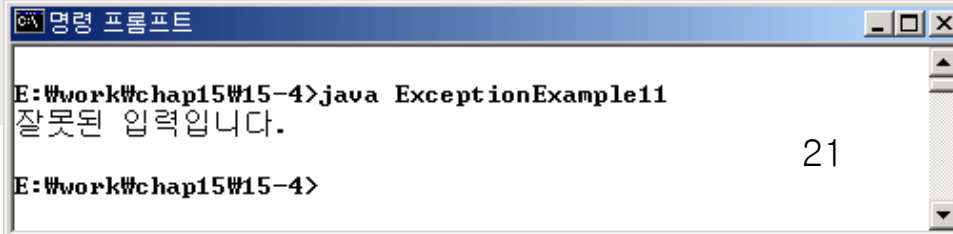
익셉션 클래스

04. 익셉션 클래스의 선언 방법

익셉션 클래스의 선언 방법

- [예제 15-10] `InvalidInputException` 클래스를 사용하는 프로그램

```
1  class ExceptionExample11 {  
2      public static void main(String args[]) {  
3          try {  
4              int result = subtract(5, 100);  
5              System.out.println(result);  
6          }  
7          catch (InvalidInputException e) {  
8              System.err.println(e.getMessage());  
9          }  
10     }  
11     static int subtract(int a, int b) throws InvalidInputException {  
12         if (a < b)  
13             throw new InvalidInputException();  
14         return a - b;  
15     }  
16 }
```



```
명령 프롬프트  
E:\work\chap15\15-4>java ExceptionExample11  
잘못된 입력입니다.  
E:\work\chap15\15-4>
```

영

어

화

시

다

^^!