JAVA Exception

프로그램은 사람이 이해하는 코드를 작성.

느려도 꾸준하면 경기에서 이긴다.

작성자 : 홍효상
이메일 : hyomee@naver.com
소스 : https://github.com/hyomee/JAVA_EDU

Content

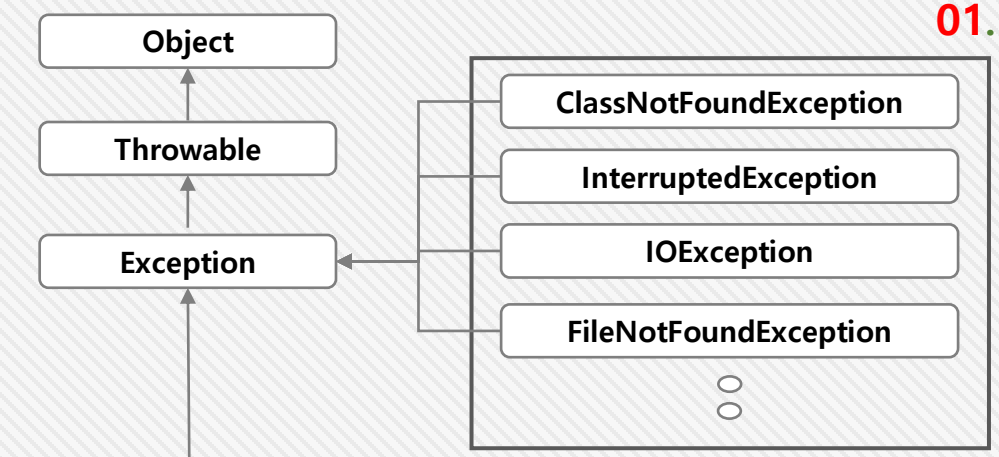
12. Exception

1. Exception 기본
2. try .. catch .. Finally
3. 사용자 정의

1. Exception

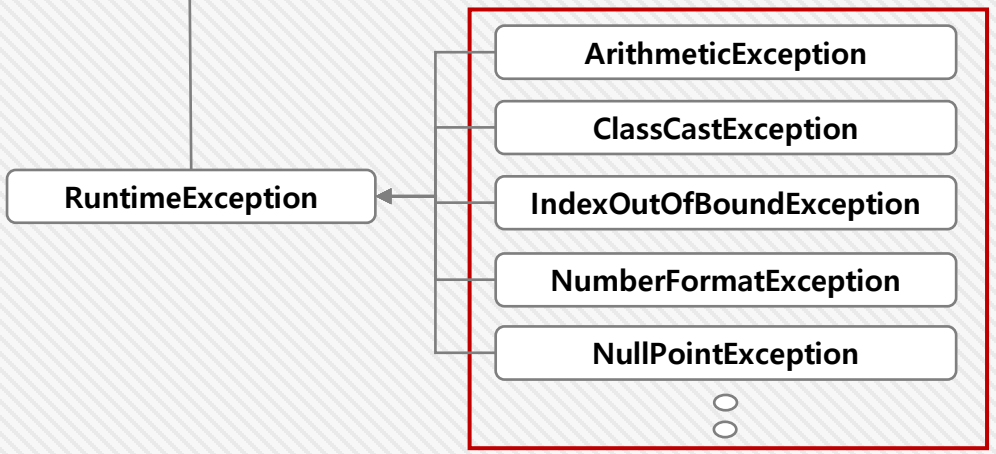
"다양하게 발생하는 오류에 대해서 해결방안 제시"

예외 클래스 구조



01. Chedked Exception : 컴파일 전 체크 -> 반드시 예외 처리 Exception 작성

- 해당 클래스 존재 하지 않을 때 발생 : Class 클래스 사용시 작성
- Thread 사용시 반드시 구현 해야 하는 Exception
- Java I/O 시 발생 하는 것을 반드시 예외 처리 해야 하는 Exception
- 파일이 경로에 없는 경우 처리 하기 위한 예외 처리 해야 하는 Exception



02. UnChedked Exception : 실행 할 때 체크 (Runtime Exception)

- 산술 연산 발생 하는 오류 처리
- Casting 시 발생 하는 오류 처리
- 배열과 같은 객체에서 index 범위 밖에 있는 index 사용시 발생 하는 오류 처리
- 문자열을 숫자, 실수 형으로 변환 시 발생 하는 오류 처리
- 참조 변수가 실제 객체가 없어서 발생 하는 오류 처리

1. Exception

01. Chedked Exception

```
void checkExceptionThread()  {
  // Unhandled exception: java.lang.InterruptedException
  Thread.sleep(100);
}
```

```
void checkExceptionThread() throws InterruptedException {
  // Unhandled exception: java.lang.InterruptedException
  Thread.sleep(100);
}
```

```
void checkExceptionThread01() {
  try {
    Thread.sleep(100);
  } catch (InterruptedException e) {
    e.printStackTrace();
  }
}
```

```
void iofileExceptioStream() {
  InputStreamReader is = new InputStreamReader(System.in);
  // Unhandled exception: java.io.IOException
  is.read();

  // Unhandled exception: java.io.FileNotFoundException
  FileInputStream fis = new FileInputStream("readme.md");

}
```

```
void iofileExceptioStream() throws IOException {
  InputStreamReader is = new InputStreamReader(System.in);
  // Unhandled exception: java.io.IOException
  is.read();

  // Unhandled exception: java.io.FileNotFoundException
  FileInputStream fis = new FileInputStream("readme.md");

}
```

```
void iofileExceptioStream01()  {
  InputStreamReader is = new InputStreamReader(System.in);
  try {
    is.read();
  } catch (IOException e) {
    e.printStackTrace();
  }

  try {
    FileInputStream fis = new FileInputStream("readme.md");
  } catch (FileNotFoundException e) {
    e.printStackTrace();
  }

}
```

1. try .. catch .. finally

try .. catch .. finally

```
try {
  ....
} catch ( 예외클래스명 참조변수명 )
{
  ....
} finally {
  ....
}
```

```java
public void arithmeticExveption() {
  // Exception in thread "main" java.lang.ArithmeticException: / by zero
  int num  = 10/0;

  int[] arrNum = {1,2,3};
  //Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: Index 3 out of bounds for length 3
  System.out.println(arrNum[3]);
}
```

01. 각각 예외 처리

```java
public void arithmeticExveption() {
  try {
    int num = 10 / 0;
  } catch (ArithmeticException ex) {
    System.out.println("예러 발생 .... " +  ex.getMessage());
  }
  int[] arrNum = {1,2,3};
  try {
    System.out.println(arrNum[3]);
  } catch (ArrayIndexOutOfBoundsException ex) {
    System.out.println("예러 발생 .... " +  ex.getMessage());
  }
}
```

02. 예외 처리 - 모아서

```java
public void arithmeticExveption() {
  try {
    int num = 10 / 0;
    int[] arrNum = {1,2,3};
  } catch (ArithmeticException ex) {
    System.out.println("예러 발생 .... " +  ex.getMessage());
  } catch (ArrayIndexOutOfBoundsException ex) {
    System.out.println("예러 발생 .... " +  ex.getMessage());
  } finally {
    System.out.println("무조건 실행 ");
  }
}
```

1. try .. catch .. finally

try .. catch .. finally

03. 예외 처리 - 한번에

```
public void arithmeticExveption() {
  try {
    int num = 10 / 0;
    int[] arrNum = {1,2,3};
  } catch (ArithmeticException | ArrayIndexOutOfBoundsException ex ) {
    System.out.println("에러 발생 .... " +  ex.getMessage());
  } finally {
    System.out.println("무조건 실행 ");
  }
}
```

04. 예외 처리 - 전가

```
public void arithmeticExveption() {
  try {
    int num = 10 / 1;
    arrayIndexErr();
  } catch (ArithmeticException ex ) {
    System.out.println("에러 발생 .... " +  ex.getMessage());
  } catch (ArrayIndexOutOfBoundsException ex) {
    System.out.println("에러 발생 .... " +  ex.getMessage());
  } catch (Exception ex) {
    System.out.println("모든 예외는 이곳으로.... " +  ex.getMessage());
  } finally {
    System.out.println("무조건 실행 ");
  }
}

public void arrayIndexErr() throws ArrayIndexOutOfBoundsException {
  int[] arrNum = {1, 2, 3};
  System.out.println(arrNum[3]);
}
```

1. 사용자 정의

사용자 정의

01. 사용자 정의 Class 만듦

```
public class MyException extends Exception{
  MyException() { }
  MyException(String errStr) {
    System.out.println("예외 : " + errStr);
  }
}
```

```
public class MyRuntimeException extends RuntimeException{
  MyRuntimeException() { }
  MyRuntimeException(String errStr) {
    System.out.println("Runtime 예외 : " + errStr);
  }
}
```

```
public void userException() {
  int a = 3;
  int[] arrNum = {1, 2, 3};
  if (a >=3) {
    new MyRuntimeException("오류 입니다.");
  }
}
```