

# 자바프로그래밍2

## 12주차 실습

오류 처리하기

# 실습 평가 방법

- 점수 : 100점 만점 기준(문제 별 난이도에 따라 점수 부여)
- 채점 기준 : 완성도, 작동 유무, 일부 오류 등에 따라 감점
  - 프로그램이 동작하지 않거나, 코드 공유, ChatGpt 사용 등의 부정행위 적발 시 0점
  - 소스 코드에 허점, 잘못된 들여쓰기, 일부 입출력 오작동 시 정도에 따라 감점
- 제출 기한 : 실습 당일 23시 59분까지(이후 제출 불가능)
  - 실습 시간(14:00-15:50) 내 제출 시 감점X
  - 18:00 까지 제출 시 채점 점수의 5% 감점
  - 20:00 까지 제출 시 채점 점수의 10% 감점
  - 23:59 까지 제출 시 채점 점수의 20% 감점

# 실습 제출 방법

- 압축 파일명 : `n주차_학번_이름.zip`
- 소스 파일 : Eclipse에서 Export한 zip 파일 내 소스 파일(.java)
- 보고서 : 각 문제별 문제 번호 및 소스 코드 실행 결과 화면 캡처한 **pdf 파일** - **부재 시 감점**
- 소스 파일과 보고서를 압축하여 주차별로 위 압축 파일명과 같이 e-루리에 제출
- e-루리 접속 오류 등 특별한 사유로 인해 제출하지 못하는 경우
  - [rkdwlgh01@naver.com](mailto:rkdwlgh01@naver.com) 해당 e-mail을 통해 제출

# 실습 조교 및 질의응답

- e-루리 Q&A 게시판 활용 (작성 후 e-루리 메시지 시 빠른 응답 가능)
- 실습 TA의 e-mail 활용
  - 강지호 : [rkdwlgh01@naver.com](mailto:rkdwlgh01@naver.com)

# [자바프로그래밍2] 12주차 실습 문제

제출 기한 11.21(목) 23:59 전까지

# 문제 1 (20점)

- 아래 작성된 **Exercise01**에서 발생 가능한 예외를 다음 조건에 맞게 처리하세요.

- 다중 try-catch문을 사용하여 예외 처리
- Exception이 아닌 **명확한 유형의 예외**를 처리
- 예외 유형에 따라 동작 결정
  - 발생한 예외 설명(출력)하고 다시 입력을 받거나 계속 진행
  - 실행할 수 없거나 실행할 의미가 없으면  
발생한 예외 설명(출력)하고 더 이상 진행하지 않고 종료

- 클래스 명 : **Exercise01**  Exercise01.java

```
public class Exercise01 {  
  
    public static void main(String[] args) {  
        Scanner input = new Scanner(System.in);  
        int[] result = new int[5];  
        int idx = 0;  
  
        while(true) {  
            System.out.print("첫 번째 숫자를 입력하세요: ");  
            int n1 = Integer.parseInt(input.nextLine());  
            System.out.print("두 번째 숫자를 입력하세요: ");  
            int n2 = Integer.parseInt(input.nextLine());  
  
            result[idx] = n1 % n2;  
            System.out.println("나머지 연산 결과: " + result[idx]);  
            idx++;  
  
            System.out.print("\n계속하려면 y를 입력하세요: ");  
            if(!input.nextLine().toLowerCase().equals("y")) break;  
            System.out.println();  
        }  
        input.close();  
    }  
}
```

## 문제 2 (40점)

- 4주차 실습 - 문제 2의 은행 시스템에서 발생하는 예외 상황을 사용자 정의 예외로 처리하여 **BankSystem, Exercise02**을 다음 조건에 맞게 수정하세요.
  - 사용자 정의 예외 생성 (AccountException.java에서 모두 생성)
    - 계좌 관련 예외(AccountException)
      - > DuplicateAccountException, AccountNotFoundException, AuthenticationFailedException
    - 금액 관련 예외(AmountException)
      - > InvalidAmountException, InsufficientBalanceException
  - BankSystem 클래스 수정
    - 예외 상황에 맞는 **사용자 정의 예외**를 던지도록 수정(각 Exception의 이름을 보고 적절한 예외 상황에 던짐)
    - 한 번에 처리하여 하위 예외 중 어느 예외가 발생하는 지 알 수 없으면 상위 예외를 던짐
  - Exercise02 클래스에서 **다형성**으로 한 번에 예외 처리
  - Exercise02 클래스에서 계좌 개설 시, 입력 받는 각 정보에 대한 예외를 **assert**로 처리
- 클래스 명 : **AccountException(그 외 사용자 정의 예외), BankSystem, Exercise02**

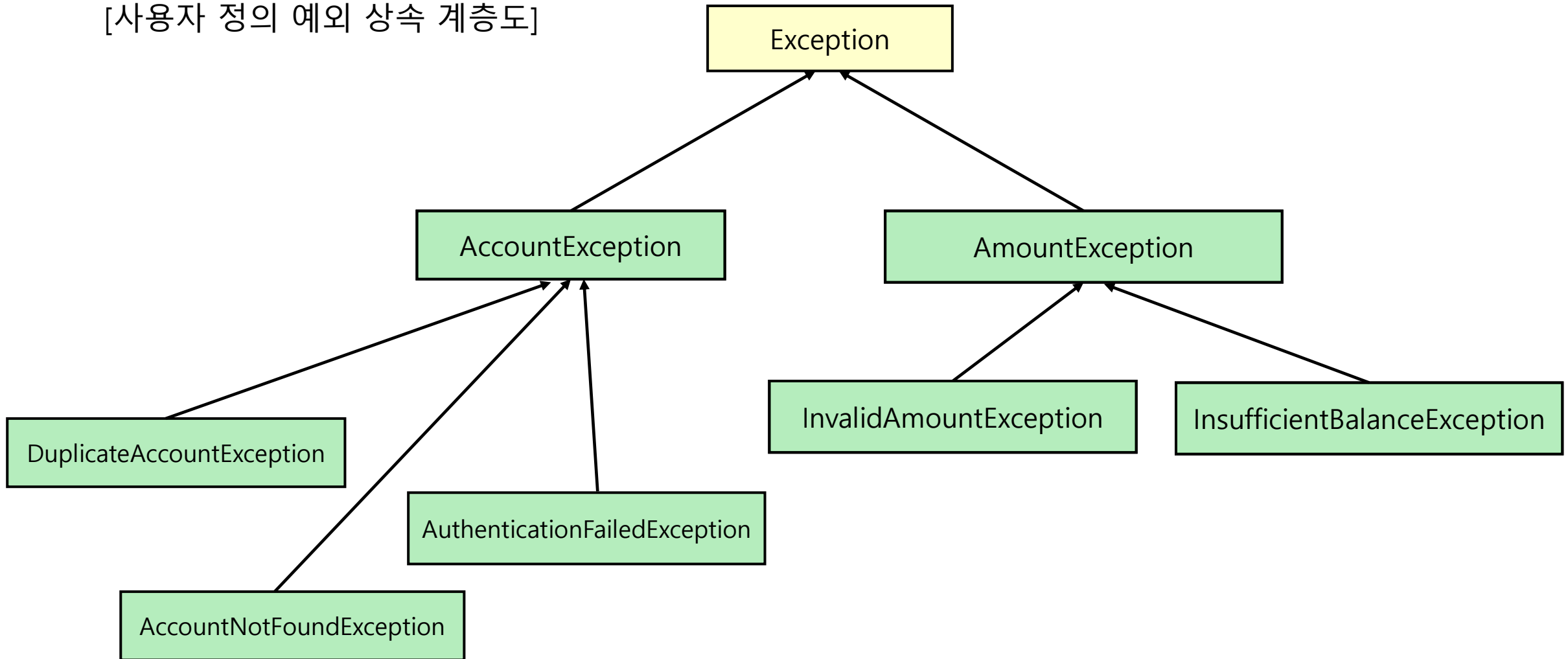
📄 AccountException.java

📄 BankSystem.java

📄 Exercise02.java

## 문제 2 (40점)

[사용자 정의 예외 상속 계층도]



■ 에 대해서만 구현하면 됨



## 문제 2 (40점)

실행결과 예시 :

```
=====menu=====
0. 계좌 개설
1. 계좌 로그인(재로그인)
2. 입출금/이체/조회
3. 종료
=====
메뉴를 선택하세요: 0
이름:
Exception in thread "main" java.lang.AssertionError: 이름은 비어있거나 공백일 수 없습니다.
    at week12.Exercise02.main(Exercise02.java:33)
```

```
=====menu=====
0. 계좌 개설
1. 계좌 로그인(재로그인)
2. 입출금/이체/조회
3. 종료
=====
메뉴를 선택하세요: 0
이름: 홍길동
계좌번호: asdf
Exception in thread "main" java.lang.AssertionError: 올바른지 않은 계좌번호 형식: asdf
    at week12.Exercise02.main(Exercise02.java:37)
```


계좌 번호는 숫자로만 이뤄져야 함(지난주 실습 참고)

```
=====menu=====
0. 계좌 개설
1. 계좌 로그인(재로그인)
2. 입출금/이체/조회
3. 종료
=====
메뉴를 선택하세요: 0
이름: 홍길동
계좌번호: 1234
비밀번호:
Exception in thread "main" java.lang.AssertionError: 비밀번호는 비어있거나 공백일 수 없습니다.
    at week12.Exercise02.main(Exercise02.java:41)
```

```
=====menu=====
0. 계좌 개설
1. 계좌 로그인(재로그인)
2. 입출금/이체/조회
3. 종료
=====
메뉴를 선택하세요: 0
이름: 홍길동
계좌번호: 1234
비밀번호: 1234
금액: -100
Exception in thread "main" java.lang.AssertionError: 초기 금액은 0 이상이어야 합니다.: -100
    at week12.Exercise02.main(Exercise02.java:46)
```

+그 외 실행 결과는 수정 전과 동일함

## 문제 3 (40점)

- 문제 1에서 구현한 프로그램에 파일 저장 기능과 로깅 설정을 추가하여 **Exercise03**에서 구현하고 테스트 하세요.
  - 연산 결과를 results.txt 파일에 저장
    - 저장 형식은 첫 번째 숫자: 10, 두 번째 숫자: 3, 나머지: 1
    - try-with-resources를 사용하여 자동으로 닫도록 함
  - 로깅 설정
    - 로깅 내용은 logging.txt 파일에 저장
    - 로깅 레벨은 **INFO**로 설정
    - 정상적인 연산 결과는 **INFO** 레벨로 기록
    - 예외 발생 상황은 **WARNING**, **SEVERE** 레벨로 기록
      - > **WARNING** 레벨은 다시 입력을 받아 계속 진행할 수 있는 예외 상황
      - > **SEVERE** 레벨은 실행할 수 없거나 실행할 의미가 없어 더 이상 진행할 수 없는 예외 상황
- 클래스 명 : **Exercise03**  Exercise03.java

# 문제 3 (40점)

## 실행결과 예시 :

첫 번째 숫자를 입력하세요: 5  
두 번째 숫자를 입력하세요:  
예외 발생:  발생한 예외 설명  
11월 19, 2024 3:20:20 오전 week12.Exercise03 main  
경고:  예외 유형 발생

계속하려면 y를 입력하세요: y

첫 번째 숫자를 입력하세요: 10  
두 번째 숫자를 입력하세요: 3  
나머지 연산 결과: 1  
11월 19, 2024 3:20:24 오전 week12.Exercise03 main  
정보: 연산 성공: 첫 번째 숫자=10, 두 번째 숫자=3, 나머지=1

계속하려면 y를 입력하세요: y

첫 번째 숫자를 입력하세요: 8  
두 번째 숫자를 입력하세요: 5  
나머지 연산 결과: 3  
11월 19, 2024 3:20:28 오전 week12.Exercise03 main  
정보: 연산 성공: 첫 번째 숫자=8, 두 번째 숫자=5, 나머지=3

계속하려면 y를 입력하세요: n

results.txt - Windows 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

첫 번째 숫자: 10, 두 번째 숫자: 3, 나머지: 1  
첫 번째 숫자: 8, 두 번째 숫자: 5, 나머지: 3

logging.txt - Windows 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE log SYSTEM "logger.dtd">
<log>
  <record>
    <date>2024-11-19T18:20:20.099389500Z</date>
    <millis>1732040420099</millis>
    <nanos>389500</nanos>
    <sequence>0</sequence>
    <logger>week12.Exercise03</logger>
    <level>WARNING</level>
    <class>week12.Exercise03</class>
    <method>main</method>
    <thread>1</thread>
    <message><input type="text" value="예외 유형" /> 발생</message>
  </record>
  <record>
    <date>2024-11-19T18:20:24.222218300Z</date>
    <millis>1732040424222</millis>
    <nanos>218300</nanos>
    <sequence>1</sequence>
    <logger>week12.Exercise03</logger>
    <level>INFO</level>
    <class>week12.Exercise03</class>
    <method>main</method>
    <thread>1</thread>
    <message>연산 성공: 첫 번째 숫자=10, 두 번째 숫자=3, 나머지=1</message>
  </record>
  <record>
    <date>2024-11-19T18:20:28.058008400Z</date>
    <millis>1732040428058</millis>
    <nanos>8400</nanos>
    <sequence>2</sequence>
    <logger>week12.Exercise03</logger>
    <level>INFO</level>
    <class>week12.Exercise03</class>
    <method>main</method>
    <thread>1</thread>
    <message>연산 성공: 첫 번째 숫자=8, 두 번째 숫자=5, 나머지=3</message>
  </record>
</log>
```