

객체지향 프로그래밍

Object-oriented Programming

본 자료는 한빛아카데미에서 제공한 강의자료를 기반으로 재구성하였습니다.

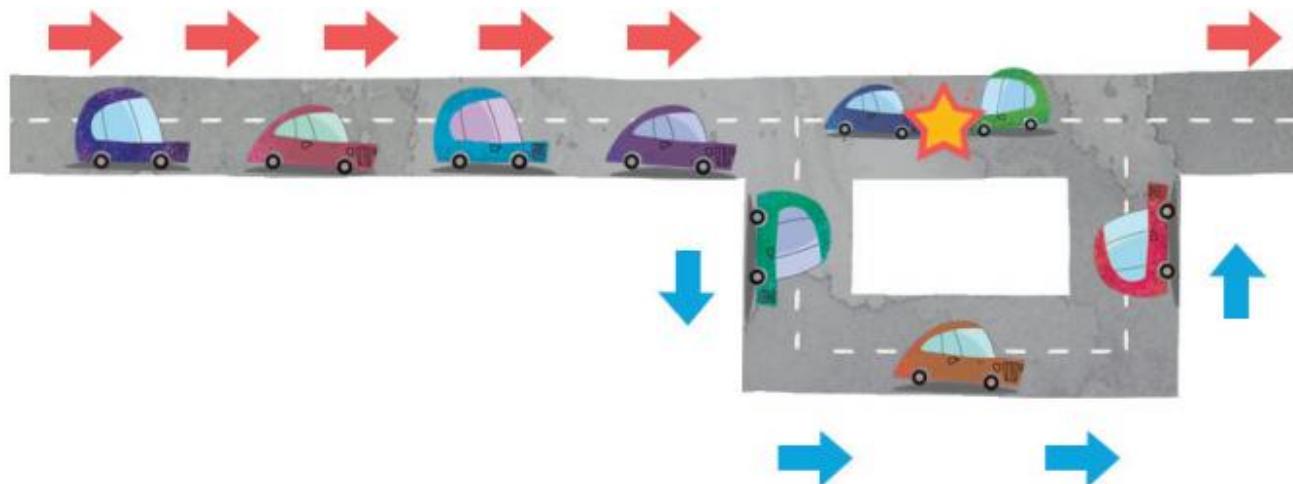
Chapter 09. 예외 처리와 제네릭 프로그래밍

- 예외
- 예외 처리 방법
- 제네릭 클래스와 인터페이스
- 제네릭 상속과 타입 한정
- 제네릭 메서드

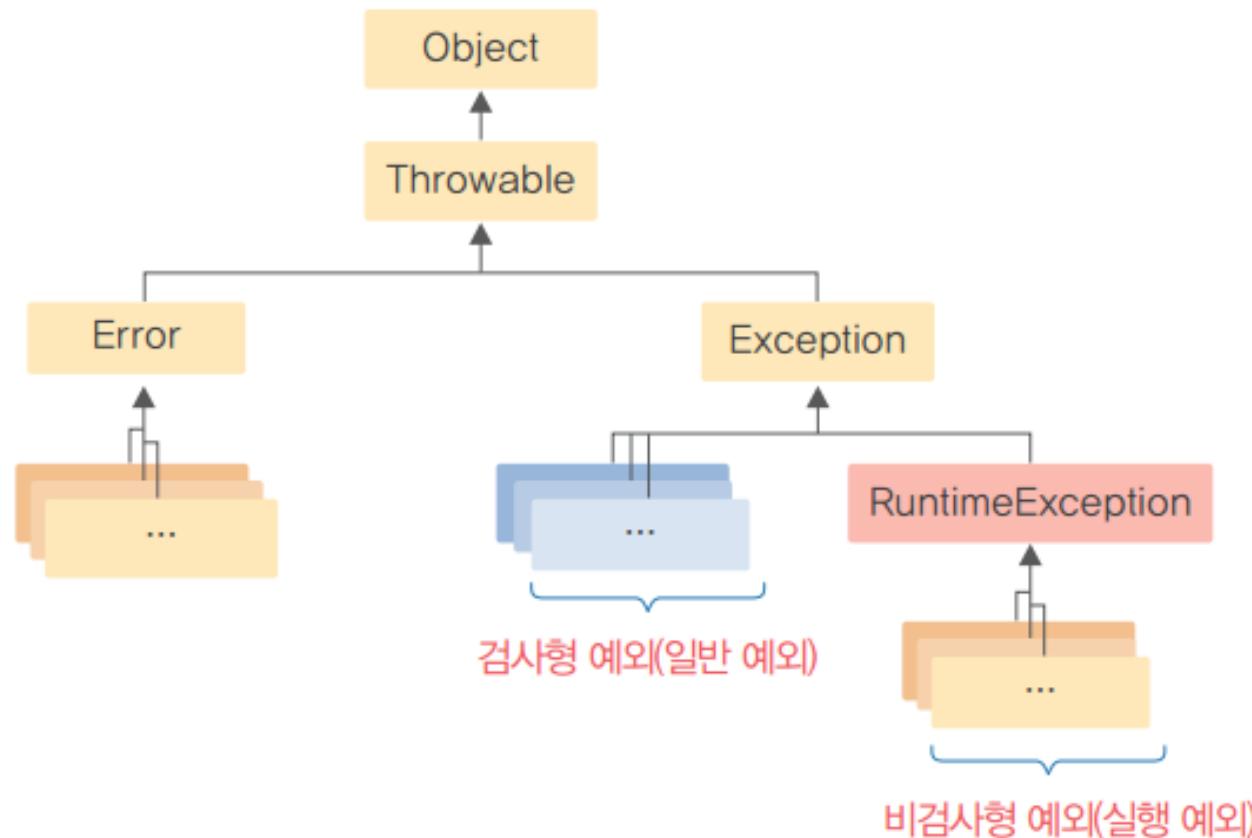
예외

예외

- 에러(error) : 개발자가 해결할 수 없는 치명적인 오류
- 예외(exception) : 개발자가 해결할 수 있는 오류
- 예외가 발생하면 비정상적인 종료를 막고, 프로그램을 계속 진행할 수 있도록 우회 경로를 제공하는 것이 바람직함



자바 예외의 종류



실행 예외 (Runtime Exception)

- 개발자의 실수로 발생할 수 있음
- 예외 처리 코드를 별도로 작성하지 않아도 컴파일 가능한 비검사형 예외 (unchecked exception)
- 대표적인 실행 예외들:

실행 예외	발생 이유
ArithmaticException	0으로 나누기와 같은 부적절한 산술 연산을 수행할 때 발생한다.
IllegalArgumentException	메서드에 부적절한 인수를 전달할 때 발생한다.
IndexOutOfBoundsException	배열, 벡터 등에서 범위를 벗어난 인덱스를 사용할 때 발생한다.
NoSuchElementException	요구한 원소가 없을 때 발생한다.
NullPointerException	null 값을 가진 참조 변수에 접근할 때 발생한다.
NumberFormatException	숫자로 바꿀 수 없는 문자열을 숫자로 변환하려 할 때 발생한다.

일반 예외

- 예외 처리 로직을 작성해야 하는 검사형 예외 (checked exception)
- 예외 처리를 하지 않으면 컴파일 에러 발생
- 대표적인 일반 예외

일반 예외	발생 이유
ClassNotFoundException	존재하지 않는 클래스를 사용하려고 할 때 발생한다.
InterruptedException	인터럽트되었을 때 발생한다.
NoSuchFieldException	클래스가 명시한 필드를 포함하지 않을 때 발생한다.
NoSuchMethodException	클래스가 명시한 메서드를 포함하지 않을 때 발생한다.
IOException	데이터 읽기 같은 입출력 문제가 있을 때 발생한다.

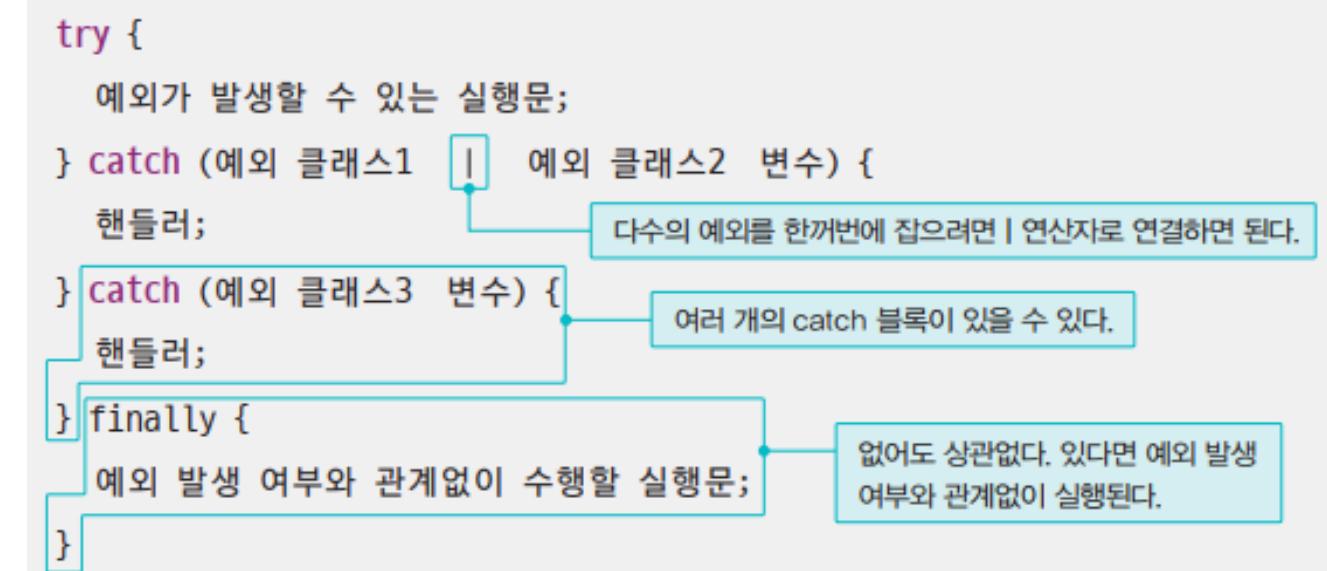
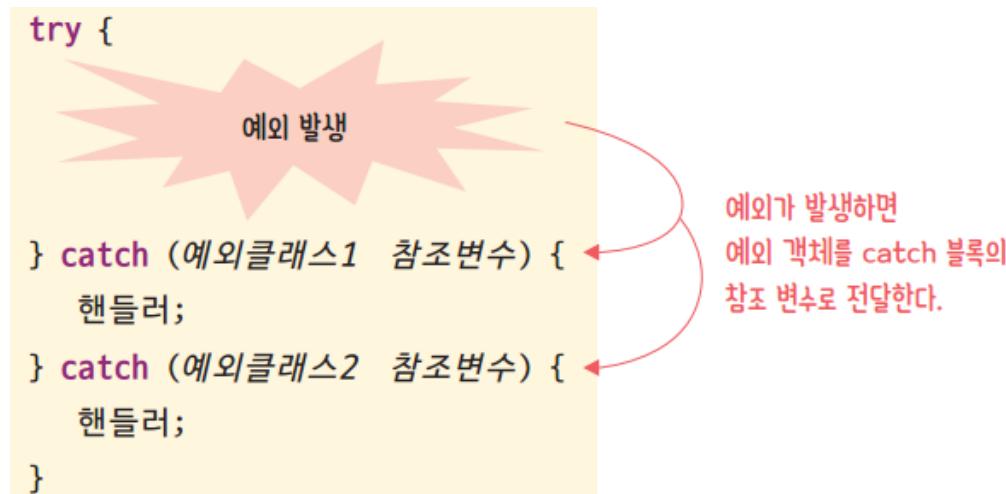
예외 처리 방법

예외 처리 방법

- **방법 1:** 예외 잡아 처리하기
- **방법 2:** 예외 떠넘기기

예외 잡아 처리하기 (try~catch)

- 예외가 발생 가능한 코드를 try~catch로 감싸 예외 발생 시 로직을 별도로 정의



예외 처리 방법

Try with resource

- 파일, 네트워크 입출력 등을 수행 시 입출력을 완료하고 나면 자원을 반환해야 함
- Try with resource는 예외 발생 여부에 상관 없이 안전하게 자원을 반환하도록 도와주는 구문

```
try (리소스) {  
}  
} catch ( ... ) {  
}
```

```
try (PrintWriter writer = new PrintWriter("output.txt")) {  
  
    writer.println("ABCDE");  
    writer.println("FGHIJ");  
  
} catch (IOException e) {  
    System.err.println("오류 발생: " + e.getMessage());  
}
```

예외 처리 방법

예외 떠넘기기

- 메서드에서 발생한 예외를 내부에서 처리하기가 부담스러울 때는 throws 키워드를 사용해 예외를 상위 코드 블록으로 양도 가능



예외 처리 방법

예외 떠넘기기

- 메서드에서 발생한 예외를 내부에서 처리하기가 부담스러울 때는 throws 키워드를 사용해 예외를 상위 코드 블록(메서드를 호출한 코드)으로 양도 가능
- 메서드 선언 시 throws 키워드를 사용해 떠넘길 예외 클래스를 명시

```
public void write(String filename)
    throws IOException, ReflectiveOperationException {
    // 파일 쓰기와 관련된 실행문 ...
}
```

throws는 예외를 다른 메서드로 떠넘기는 키워드이다.

예외를 1개 이상 선언할 수 있다.

예외 처리 방법

예외 떠넘기기

- **throw <예외 객체>**: 상위 코드 영역으로 예외를 떠넘김

```
● ● ●

public static void checkAgeForEntry(int age) throws IllegalArgumentException {
    if (age < 19) {
        throw new IllegalArgumentException("미성년자는 출입할 수 없습니다. (나이: " + age + ")");
    }
    System.out.println("나이 확인 완료: " + age + "세. 입장이 가능합니다.");
}
```

Practice

예외, 예외 처리 실습

- 예제 9-1, 9-2 (p.354): 실행 예외
- 예제 9-3 (p.355): 검사형 예외
- 예제 9-5, 9-6 (p.358): 예외 잡아 처리하기
- 예제 9-8 (p.362): 예외 떠넘기기

Practice

Problem 1. 나눗셈 예외 처리

사용자로부터 두 개의 정수를 입력받아 나눗셈 결과를 출력하는 프로그램을 작성하시오.

단, 0으로 나누는 경우 "0으로 나눌 수 없습니다." 를 출력하고 프로그램이 강제 종료되지 않도록 하시오.

실행 예시

정수1 입력: 10

정수2 입력: 0

출력: 0으로 나눌 수 없습니다.

Practice

Problem 2

다음 프로그램을 완성하시오.

입력 과정 중 예외가 발생하더라도 "프로그램을 종료합니다." 메시지가 항상 출력되도록 하시오.

```
● ● ●  
Scanner sc = new Scanner(System.in);  
try {  
    System.out.print("정수 입력: ");  
    int num = sc.nextInt();  
    System.out.println("입력한 수: " + num);  
} catch /* 여기에 코드 작성 */ {  
    System.out.println("정수가 아닙니다.");  
} finally {  
    // 여기에 코드 작성  
}
```

Practice

Problem 3. 점수 검증

int score를 매개변수로 받아, 0보다 작거나 100보다 큰 값이 들어오면
IllegalArgumentException을 발생시키는 메서드 checkScore(int score)를 작성하시오.

실행 예시

```
checkScore(150); // 예외 발생: "점수는 0 이상 100 이하만 가능합니다."
```

Practice

Problem 4

파일에서 숫자 목록을 읽어 모든 숫자의 평균을 계산하는 프로그램을 작성하시오.

요구사항:

- 파일이 존재하지 않으면 FileNotFoundException 발생
- 파일 내용이 정수가 아닌 경우 NumberFormatException 발생
- 두 예외를 readNumbers() 메서드 내부에서 발생시키고, main()에서 try-catch로 처리하시오.
- 예외 발생 시 각각
 - "파일을 찾을 수 없습니다."
 - "파일에 숫자가 아닌 데이터가 포함되어 있습니다."를 출력하시오.

파일 읽기: Scanner sc = new Scanner(new File(fileName));

Practice

Problem 5

학생의 시험 점수를 입력받아 합격 여부를 판단하는 프로그램을 작성하시오.

요구사항:

- RuntimeException을 상속 받는 InvalidScoreException과 FailException 예외 클래스 정의
- 점수가 0~100 범위를 벗어나면 InvalidScoreException 발생
- 점수가 60점 미만이면 FailException 발생
- 입력이 정수가 아니면 "정수를 입력해야 합니다." 출력
- 입력 후 예외 발생 여부와 관계없이 "프로그램 종료"가 출력되어야 함

점수 입력: -10
예외 발생: InvalidScoreException - "점수는 0~100 사이여야 합니다."
프로그램 종료

점수 입력: 50
예외 발생: FailException - "불합격입니다."
프로그램 종료

점수 입력: 80
합격입니다.
프로그램 종료

Practice

Problem 6

- 추상 클래스인 Animal을 상속받는 Dog, Cat, Bat 클래스를 만든다.
- Animal 클래스에는 `void speak(int time)` 메서드가 있으며, 하위 클래스에서 각각
 - Dog: "멍멍"
 - Cat: "야옹"
 - Bat: "찌찌"을 출력한다.
- 단, Cat의 경우 밤 시간(22~6시)에, Bat의 경우 6시~22시에 SleepTimeException(사용자 정의 예외)을 던진다.
- main()에서 현재 시간을 기준으로 Animal 타입의 배열을 순회하며 speak()를 호출하고, 예외 발생 시 "고양이/박쥐는 지금 잘 자고 있습니다." 를 출력하시오.