Chapter8

1. 예외처리의 정의와 목적
    - 예외처리의 정의
        - 프로그램 실행 시 발생할 수 있는 예외에 대비한 코드를 작성하는 것
    - 목적
        - 프로그램의 비정상 종료를 막고, 정상적인 실행 상태를 유지하는 것.
2. 예외가 발생하여 화면에 출력된 내용중 옳지 않은 것
    - d - method1이 method2를 호출
3. 오버라이딩이 잘못 된것
    - d - Exception은 예외클래스의 조상이므로 제일 많은 예외를 처리한다.
    - e -
4. 예외 처리를 잘못 한 것
    - c - 작은 게 먼저 나와야함
5. 실행 결과

    1
    3
    5
    1
    2
    5
    6
6. 실행 결과

    3
    5
7. 실행 결과

    1

8. 숫자가 아닌 값을 입력했을때 예외 처리

```
try {
  input = new Scanner...
} catch(Exception e) {
  sout("숫자가 아님");
}
```

9. 예외 작성

```
class UnsupportedFunctionException extends RuntimeException {
  private final int ERR_CODE = 100;

  public UnsupportedFunctionExcepton(String msg, int err) {
    super(msg);
    this.ERR_CODE = err;
  }

  public UnsupprtedFunctionException() { }

  public int getErrorCode( {
    return this.ERR_CODE;
  }

  public String getMessage() {
    return "[" + this.ERR_CODE + "]" + super.getMessage();
  }
}
```

10. 실행 결과

    2
    4
    7