

# 자동 리소스 닫기

# 자동 리소스 닫기

---

## ❖ try-with-resources

- 예외 발생 여부와 상관 없음
- 사용했던 리소스 객체의 `close()` 메소드 호출해 리소스 닫음
- 리소스 객체
  - 각종 입출력스트림, 서버소켓, 소켓, 각종 채널
  - `java.lang.AutoCloseable` 인터페이스 구현하고 있어야 함

# 자동 리소스 닫기

---

## ❖ try-with-resources

```
FileInputStream fis = null;

try {
    ...
    fis = new FileInputStream("file.txt");
} catch(IOException e) {
    ...
} finally {
    ...
    if(fis != null){
        try {
            fis.close();
        } catch (IOException e) { }
    }
}
```

## 자동 리소스 닫기

---

### ❖ AutoCloseable 구현 클래스: FileInputStream.java

```
public class FileInputStream2 implements AutoCloseable {
    private String file;

    public FileInputStream2(String file) {
        this.file = file;
    }

    public void read() {
        System.out.println(file + "을 읽습니다.");
    }

    @Override
    public void close() throws Exception {
        System.out.println(file + "을 닫습니다.");
    }
}
```

## 자동 리소스 닫기

---

### ❖ AutoCloseable 구현 클래스: TryWithResourceExample.java

```
public class TryWithResourceExample {  
    public static void main(String[] args) {  
        try (FileInputStream2 fis = new FileInputStream2("file.txt")) {  
            fis.read();  
            throw new Exception();  
        } catch (Exception e) {  
            System.out.println("예외 처리 코드가 실행되었습니다.");  
        }  
    }  
}
```