defer

Go의 `defer`

- 함수가 종료될 때 실행할 작업을 예약하여 자원 해제 및 정리 작업을 보장.
- 영어의 뜻대로, 연기하다 (=put off)
- 작동 방식
  - 예: `defer`가 호출되면 해당 함수가 종료될 때 실행할 함수를 스택에 푸시(push).
  - 실행 시점: 예약된 함수는 예약된 순서의 역순(후입선출, LIFO)으로 실행됨.
- 사용 예
  - 파일 열기 및 닫기:

```
file, err := os.Open("example.txt")
if err != nil {
    // 오류 처리
}
defer file.Close() // 파일 닫기를 예약
// 파일 작업 수행
```

  - 네트워크 연결:

```
conn, err := net.Dial("tcp", "example.com:80")
if err != nil {
    // 오류 처리
}
defer conn.Close() // 연결 닫기를 예약
// 네트워크 작업 수행
```

  - 리소스 해제 보장: 함수 중간에 오류가 발생하더라도 함수가 종료될 때 예약된 작업이 실행됨.
- 특징
  - `defer` 사용 시, 예약된 순서의 역순으로 실행됨:

```
defer fmt.Println("First")
defer fmt.Println("Second")
defer fmt.Println("Third")
// 출력: Third, Second, First (함수 종료 시)
```

자바랑 비교

- 예외과 실행 시점
  - Go[`defer`]: 함수 종료 시 예약된 작업을 역순으로 실행.
  - Java[`try-with-resources`]: `try` 블록이 종료되거나 예외가 발생할 때 자원을 자동으로 닫음.
- 사용 편의성
  - Go[`defer`]: 간단한 구문으로 다양한 작업을 예약 가능.
  - Java[`try-with-resources`]: `AutoCloseable` 인터페이스를 구현한 자원만 자동 닫기 가능.
- 에러 처리
  - Go: `defer`를 통해 예외가 발생하더라도 자원 해제를 보장.
  - Java: `try-with-resources`를 통해 예외가 발생하더라도 자원을 자동으로 닫음.

예제

- **Go** 예제

```
package main

import (
    "fmt"
    "os"
)

func main() {
    file, err := os.Open("example.txt")
    if err != nil {
        fmt.Println("Error:", err)
        return
    }
    defer file.Close() // 파일 닫기를 예약

    // 파일 작업 수행
}
```

- **Java** 예제

```
import java.io.FileInputStream;
import java.io.IOException;

public class Main {
    public static void main(String[] args) {
        try (FileInputStream file = new FileInputStream("example.txt")) {
```

```
            // 파일 작업 수행
        } catch (IOException e) {
            System.out.println("Error: " + e.getMessage());
        }
        // try 블록을 벗어나면 file이 자동으로 닫힘
    }
}
```

Revision #4
Created 26 July 2024 08:33:04 by Yeonwoo Kim
Updated 26 July 2024 08:38:49 by Yeonwoo Kim