KB금융그룹 | 국민의 평생 금융파트너

2025년 상반기 K-디지털 트레이닝

Command - 명령을 클래스로 표현한다

[KB] IT's Your Life

KB 국민은행

Command 패턴

Command 패턴

Command 클래스

- '이 일을 처리 하시오'라는 '명령'을 표현하는 클래스
- 처리하고 싶은 일을 '메소드 호출'이라는 동적인 처리로 표현하지 않고, 명령을 나타내는 클래스의 인스턴스라는 하나의 '객체'로 표현

명령의 집합을 저장해 둠

- 같은 명령을 재실행 가능
- 여러 명령을 모아서 새로운 명령으로 재사용 가능

이벤트 구동형 프로그래밍(GUI 프로그래밍)

- Command를 Event라고도 하며,
- 해당 이벤트가 발생했을 때 해당 객체를 큐에 저장하고, 순서대로 실행

Command 패턴 예제 프로그램

예제 프로그램

- CLI 메뉴 기반의 간단한 어플리케이션
  - 사용자가 선택한 메뉴를 실행

클래스	설명
Command	명령 인터페이스
AddCommand	두 수를 입력받아 합계를 구하는 명령
OpenCommand	파일 열기를 실행하는 명령
PrintCommand	프린터 출력을 실행하는 명령
ExitCommand	어플리케이션을 종료하는 명령
Main	명령 집합을 운영하는 어플리케이션 클래스

Command 패턴 예제 프로그램

Command.java

```
public interface Command {
  void execute();
}
```

Command 패턴 예제 프로그램

AddCommand.java

```
public class AddCommand implements Command {

  @Override
  public void execute() {
    Scanner scanner = new Scanner(System.in);
    System.out.print("숫자1: ");
    int num1 = scanner.nextInt();

    System.out.print("숫자2: ");
    int num2 = scanner.nextInt();

    System.out.printf("%d + %d = %d\n", num1, num2, num1 + num2);
  }
}
```

Command 패턴 예제 프로그램

OpenCommand.java

```
public class OpenCommand implements Command {

  @Override
  public void execute() {
    System.out.println("===============");
    System.out.println("Open command");
    System.out.println("===============");
    System.out.println();
  }
}
```

Command 패턴 예제 프로그램

PrintCommand.java

```
public class PrintCommand implements Command {

  @Override
  public void execute() {
    System.out.println("===============");
    System.out.println("Print command");
    System.out.println("===============");
    System.out.println();
  }
}
```

Command 패턴 예제 프로그램

ExitCommand.java

```
public class ExitCommand implements Command {

  @Override
  public void execute() {
    Scanner scanner = new Scanner(System.in);
    System.out.print("종료할까요?(Y/n) ");
    String answer = scanner.nextLine();

    scanner.close();
    if(answer.isEmpty() || answer.equalsIgnoreCase("Y") ) {
      System.exit(0);
    }
  }
}
```

Command 패턴 예제 프로그램

파일명

```
public class Main {
  public static void main(String[] args) {
    Command[] commands = {
        new AddCommand(),
        new OpenCommand(),
        new PrintCommand(),
        new ExitCommand()
    };

    while(true) {
      Scanner scanner = new Scanner(System.in);
      System.out.println("1: Add, 2: Open, 3: Print, 4: Exit");
      System.out.print("선택: ");
      int sel = scanner.nextInt();

      commands[sel-1].execute();
    }
  }
}
```

메서드가 아닌 객체로 명령어의미를 저장. 각 객체 안에 같은 인터페이스를 가진 execute(실제 동작)메서드를 가지고 있음

Command 패턴

Command 클래스 다이어그램