

Il **Command** è un pattern comportamentale che trasforma una richiesta in un oggetto autonomo che contiene tutte le informazioni relative a quella richiesta. Questo permette di parametrizzare i metodi con diverse richieste, ritardare l'esecuzione di una richiesta o eseguire richieste in modo remoto.

Componenti del Command

- 1. **Command (Comando):** Un'interfaccia che dichiara un metodo per eseguire l'operazione.
- 2. **ConcreteCommand (Comando Concreto):** Implementa il comando specificando l'azione da eseguire.
- 3. **Invoker (Invocatore):** Richiede l'esecuzione del comando.
- 4. **Receiver (Ricevitore):** Sa come eseguire l'operazione associata al comando.

Esempio con Pokémon

Immagina di voler rappresentare le mosse dei Pokémon come comandi.

Interfaccia Command

javaCopia codice

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

ConcreteCommand

javaCopia codice

```
public class AttaccoElettrico implements Command {  
    private Pokemon attaccante;  
    private Pokemon difensore;  
  
    public AttaccoElettrico(Pokemon attaccante, Pokemon difensore) {  
        this.attaccante = attaccante;  
        this.difensore = difensore;  
    }  
  
    @Override  
    public void execute() {  
        System.out.println(attaccante.getNome() + " usa Attacco Elettrico su " + difensore.getNome());  
        // Logica di attacco elettrico  
    }  
}
```

Invoker

javaCopia codice

```
public class Allenatore {  
    private Command comando;  
  
    public void setComando(Command comando) {  
        this.comando = comando;  
    }  
  
    public void eseguiComando() {  
        comando.execute();  
    }  
}
```

Receiver

javaCopia codice

```
public class Pokemon {  
    private String nome;  
    public Pokemon(String nome) {  
        this.nome = nome;  
    }  
  
    public String getNome() {  
        return nome;  
    }  
}
```

Applicazione Principale

javaCopia codice

```
public class MainApp {  
    public static void main(String[] args) {  
        Pokemon pikachu = new Pokemon("Pikachu");  
        Pokemon charmander = new Pokemon("Charmander");  
  
        Command attaccoElettrico = new AttaccoElettrico(pikachu, charmander);  
  
        Allenatore ash = new Allenatore();  
        ash.setComando(attaccoElettrico);  
        ash.eseguiComando(); // Output: Pikachu usa Attacco Elettrico su Charmander  
    }  
}
```