

## Design Pattern

### Strategy

---

```
public interface IStrategy {
    int doOperation(int number1,int number2);
}

public class OperationAdd implements IStrategy {
    public int doOperation(int number1,int number2) {
        return number1 + number2;
    }
}

public class OperationMultiply implements IStrategy {
    public int doOperation(int number1,int number2) {
        return number1 * number2;
    }
}

public class OperationSubtract implements IStrategy {
    public int doOperation(int number1,int number2) {
        return number1 - number2;
    }
}

public class Context {
    private IStrategy strategy;

    public Context(IStrategy strategy) {
        this.strategy = strategy;
    }

    public int executeStrategy(int number1,int number2) {
        return strategy.doOperation(number1,number2);
    }
}

public class Application {
    public static void main(String... args) {
        Context context = new Context(new OperationAdd());
        System.out.println("10 + 5 = " + context.executeStrategy(10,5));

        context = new Context(new OperationSubtract());
        System.out.println("10 - 5 = " + context.executeStrategy(10,5));

        context = new Context(new OperationMultiply());
        System.out.println("10 * 5 = " + context.executeStrategy(10,5));
    }
}
```