

1) A qual padrão de projeto o código abaixo se refere? Explique.

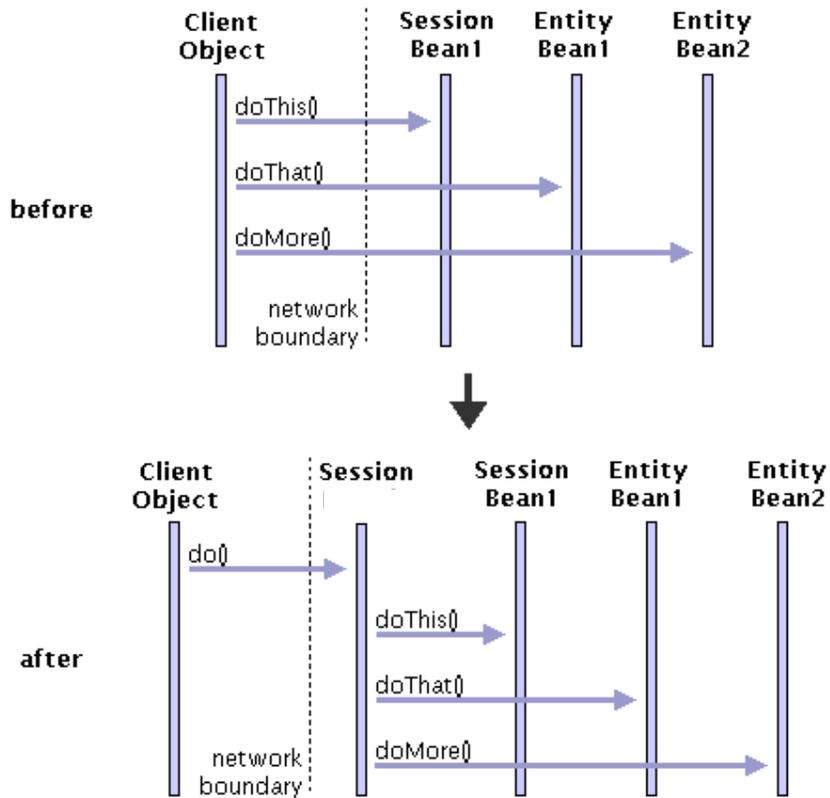
```
class SuperClass {
    public void doAlgorithm() {
        for ( int i = 1; i < 2; i++ )
        {
            print( "Loop #" + i );
            uniqueMethod1();
            commonMethod();
            uniqueMehod2();
        }
    }
    private void commonMethod()
    { print( "commonMethod" ); }
    abstract void uniqueMethod1();
    abstract void uniqueMethod2();
}

class SubClassOne extends SuperClass {
    public void uniqueMethod1()
    { print( "SubClassOne: uniqueMehod1" ); }
    public void uniqueMethod1()
    { print( "SubClassOne: uniqueMehod2" ); }
}

class SubClassTwo extends SuperClass {
    public void uniqueMethod1()
    { print( "SubClassTwo: uniqueMehod1" ); }
    public void uniqueMethod1()
    { print( "SubClassTwo: uniqueMehod2" ); }
}
```

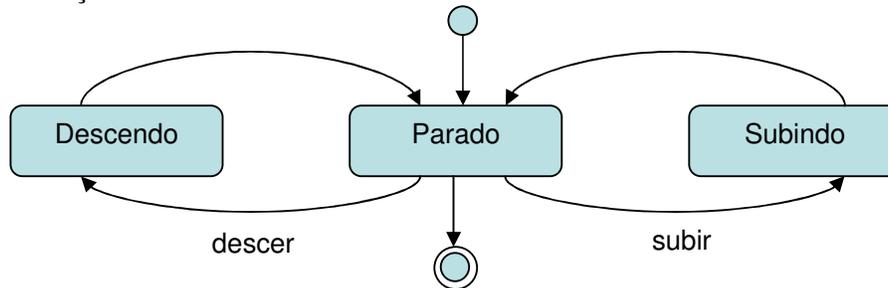
(exemplo retirado da Internet: http://gpwiki.org/index.php/Template_method_pattern)

2) A qual padrão de projeto o diagrama abaixo se refere? Explique.



(exemplo retirado da Internet: <http://java.sun.com/blueprints/patterns/SessionFacade.html>)

- 3) Um elevador pode estar parado, subindo ou descendo. Quando o elevador está parado, uma pessoa pode acionar o botão de subir ou o de descer e informar o próximo andar para o qual deseja ir. O diagrama abaixo representa as possíveis transições entre os estados.



- Qual é o padrão de projeto mais adequado para implementar o funcionamento do elevador? Por quê?
 - Faça o diagrama de classe que irá representar o funcionamento do elevador.
 - Faça o diagrama de seqüência que represente a seguinte operação: o elevador está parado e a pessoa seleciona a opção de subir. Note que após subir até o andar desejado, o elevador volta a ficar parado automaticamente.
- 4) Por que o padrão Singleton é normalmente utilizado quando o padrão Fachada é aplicado?

5) O código abaixo se refere ao padrão Strategy. Quais são os benefícios provenientes do uso deste padrão? Qual é a dificuldade para criação de uma nova estratégia?

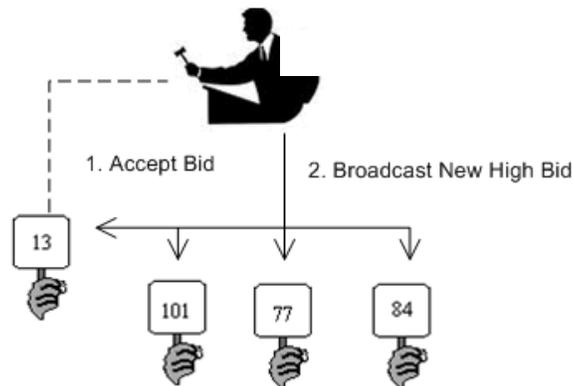
```
class StrategyExample {  
  
    public static void main(String[] args) {  
  
        Context context;  
  
        // Three contexts following different strategies  
        context = new Context(new ConcreteStrategyAdd());  
        int resultA = context.executeStrategy(3,4);  
  
        context = new Context(new ConcreteStrategySubtract());  
        int resultB = context.executeStrategy(3,4);  
  
        context = new Context(new ConcreteStrategyMultiply());  
        int resultC = context.executeStrategy(3,4);  
    }  
}  
  
interface Strategy {  
    int execute(int a, int b);  
}  
  
class ConcreteStrategyAdd implements Strategy {  
  
    public int execute(int a, int b) {  
        return a + b; // Do an addition with a and b  
    }  
}  
  
class ConcreteStrategySubtract implements Strategy {  
  
    public int execute(int a, int b) {  
        return a - b; // Do a subtraction with a and b  
    }  
}  
  
class ConcreteStrategyMultiply implements Strategy {  
  
    public int execute(int a, int b) {  
        return a * b; // Do a multiplication with a and b  
    }  
}  
  
class Context {  
    private Strategy strategy;  
  
    public Context(Strategy strategy) {  
        this.strategy = strategy;  
    }  
  
    public int executeStrategy(int a, int b) {  
        return strategy.execute(a, b);  
    }  
}
```

(exemplo retirado da Internet: http://en.wikipedia.org/wiki/Strategy_pattern)

6) Em um determinado tipo de leilão, cada participante do leilão é identificado por uma placa contendo um número, conforme a figura abaixo. Quando o participante levanta a placa, ele indica ao leiloeiro que aceita o lance atual. Neste momento o leiloeiro informa aos outros participantes que a proposta atual foi aceita por um dos participantes e indica qual é o novo lance.

Caso você tenha que implementar um sistema para a simulação deste tipo de leilão onde os participantes não tem informação direta sobre os demais participantes mas somente informação vinda do leiloeiro, qual padrão de projeto você utilizaria para implementar a relação entre o leiloeiro e os participantes?

(Dica: como o leiloeiro sabe quais são os participantes do leilão?)



(exemplo retirado da internet: http://sourcemaking.com/design_patterns/observer)