

Padrão de Projeto State

Preâmbulo

O padrão State é empregado para implementar diagramas de transição de estados. Em tais diagramas existem estados e transições entre eles. Eventos provocam transições que, por sua vez, disparam a execução de ações. Tanto as transições quanto as ações dependem do estado ativo no momento em que o evento em questão é gerado.

A implementação sugerida deste padrão faz uso de uma interface que deverá definir todas as reações possíveis para todos os estados a serem contemplados. Seja **Estado** esta interface. Cada reação pode ser definida na interface através de operações que representam os eventos possíveis. Para todo e qualquer possível evento haverá uma operação correspondente nesta interface. Será a execução destas operações que irão provocar mudanças do estado corrente. Cada estado possível do diagrama será representado por uma classe que implementa a interface Estado e, naturalmente, possui uma implementação para cada operação (evento). É cada implementação específica que identificará como é que o estado em questão reage ao evento gerado.

1. Quando devemos aplicar o padrão State? O que você vê de positivo e negativo nesse padrão?
2. Uma conta bancária realiza operações de saque e depósito diferente de acordo com seu estado. Uma conta que está negativo, por exemplo, não aceita saques, e deposita apenas 95% do valor total de um depósito efetuado. Uma conta que está com saldo positivo, aceita saques, e o banco deposita 98% do valor do depósito. Use o State para representar os possíveis estados da conta, e a sua transição de estado. Se uma conta passa a ter

saldo positivo, fica com status POSITIVO. Caso contrário, possui saldo NEGATIVO.

3. Refatore a classe Pedido para aplicar o design pattern State.

```
public class Pedido {

    private enum Status {
        PROCESSANDO, CANCELADO, ENVIADO
    };
    private Status status;
    // outros atributos

    public void alterar() {
        switch (status) {
            case PROCESSANDO:
                // código para alteração do pedido
                break;
            case CANCELADO:
                String msg1 = "Não é possível realizar alterações" +
                    "pois o pedido foi cancelado";
                throw new IllegalStateException(msg1);
            case ENVIADO:
                String msg2 = "Não é possível realizar alterações" +
                    "pois o pedido já foi enviado";
                throw new IllegalStateException(msg2);
        }
    }

    public void cancelar() {
        switch (status) {
            case PROCESSANDO:
                // Código para cancelamento do pedido
                status = Status.CANCELADO;
                break;
            case CANCELADO:
                String msg1 = "O pedido já foi cancelado";
                throw new IllegalStateException(msg1);
            case ENVIADO:
                String msg2 = "Não é possível cancelar" +
                    "pois o pedido já foi enviado";
                throw new IllegalStateException(msg2);
        }
    }

    public void enviar() {
```

```

switch (status) {
    case PROCESSANDO:
        // código para envio do pedido
        status = Status.ENVIADO;
        break;
    case CANCELADO:
        String msg1 = "Não é possível enviar" +
            "pois o pedido foi cancelado";
        throw new IllegalStateException(msg1);
    case ENVIADO:
        String msg2 = "O pedido já foi enviado";
        throw new IllegalStateException(msg2);
}
}
}

```

4. Faça uso do padrão State para implementar o comportamento registrado no diagrama abaixo. Este comportamento corresponde ao comportamento esperado para toda instância da classe Processo. Crie a classe Processo e implemente o comportamento de tal forma que, dada a ocorrência de um evento, possivelmente ocorre a transição para um dado estado, juntamente com a execução das ações julgadas oportunas. Neste caso, apenas uma ação deve ser executada como resultado da saída do estado “Em avaliação”. Implemente a ação “avisar interessado” por meio de uma simples mensagem produzida na saída padrão. Um cenário real poderia exigir o envio de correspondência eletrônica para o email do interessado, o que não é exigido neste exercício. Por fim, observe que o estado “Avaliado” é uma composição dos estados “Indeferido” e “Deferido”. São estes dois últimos que deverão ser tratados. Ou seja, “Avaliado” é apenas uma abstração que não precisa ser tratada da perspectiva de implementação. Observe que, transcorrido o prazo legal após avaliação de um processo, este é conduzido ao estado “Fechado”, independente se o estado é “Indeferido” ou “Deferido”.

