

QUESTÃO 1

```

public class ContadorAcessos {

    private static ContadorAcessos instancia;

    private int total;

    private ContadorAcessos() {
        this.total = 0;
    }

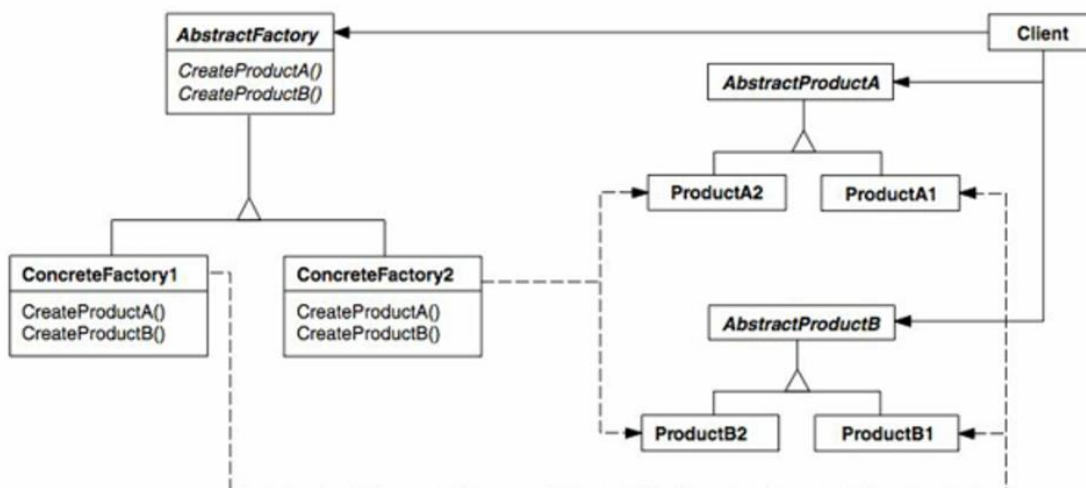
    public static ContadorAcessos getInstancia() {
        if (instancia == null) {
            instancia = new ContadorAcessos();
        }
        return instancia;
    }

    public void registrarAcesso() {
        this.total++;
    }

    public int getTotal() {
        return this.total;
    }

}
  
```

QUESTÃO 2



QUESTÃO 3

SIGLA	PRINCÍPIO	DEFINIÇÃO
S	<i>Single Responsibility Principle</i> — Responsabilidade Única	Uma classe deve ter apenas um motivo para mudar, ou seja, responder a um único ator/responsabilidade.
O	<i>Open/Closed Principle</i> — Aberto/Fechado	Entidades de software devem ser abertas para extensão, mas fechadas para modificação.
L	<i>Liskov Substitution Principle</i> — Substituição de Liskov	Objetos de um subtipo devem poder substituir objetos do supertipo sem alterar a correção do programa.
I	<i>Interface Segregation Principle</i> — Segregação de Interfaces	Clientes não devem ser forçados a depender de métodos que não utilizam; é preferível várias interfaces específicas a uma interface genérica.
D	<i>Dependency Inversion Principle</i> — Inversão de Dependência	Módulos de alto nível não devem depender de módulos de baixo nível: ambos devem depender de abstrações.

QUESTÃO 4

O padrão é o **Observer** e sua intenção é: definir uma dependência um-para-muitos entre objetos, de modo que, quando um objeto muda de estado, todos os seus dependentes sejam notificados automaticamente. Há um evento (a nota foi gerada) e vários interessados independentes — relação um-para-muitos. Todas as ações são executadas, não uma escolhida entre elas. A lista de ações muda com o tempo; a geração da nota, não. O Observer isola o que varia do que é estável. O gerador não precisa saber quem são os interessados, nem quantos são.

```
public interface AcaoAposGerarNota {
    void executa(NotaFiscal nf);
}

public class GeradorDeNotaFiscal {

    private final List<AcaoAposGerarNota> acoes = new ArrayList<>();

    public void adicionaAcao(AcaoAposGerarNota acao) {
        this.acoes.add(acao);
    }

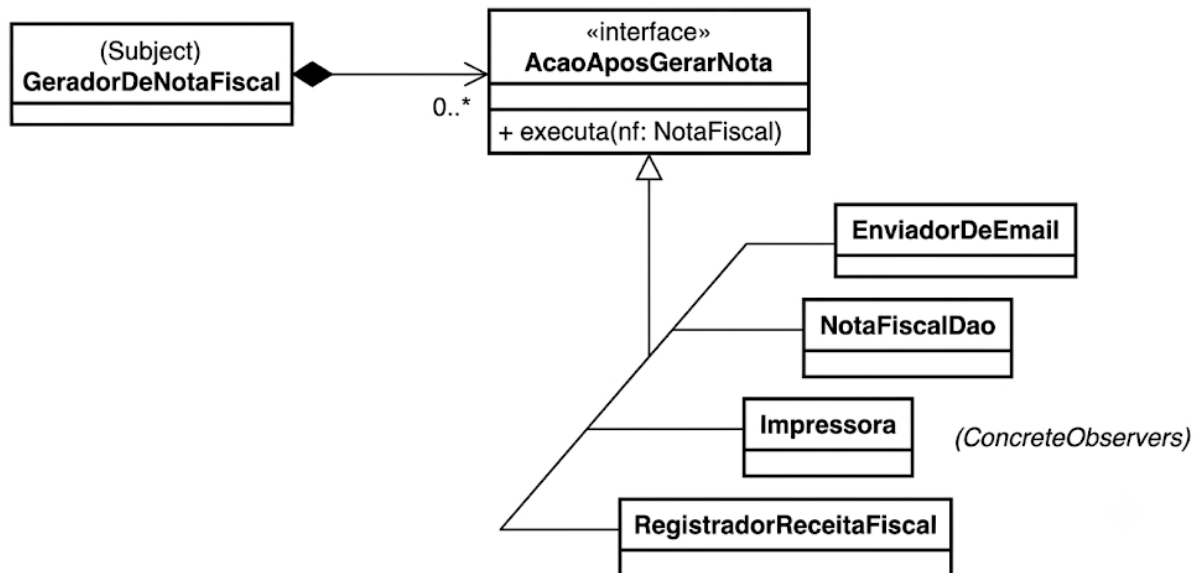
    public NotaFiscal gera(Orcamento orcamento) {
        NotaFiscal nf = new NotaFiscal(orcamento.getCliente(),
                                         orcamento.getValor(),
                                         calculaImpostos(orcamento));

        for (AcaoAposGerarNota acao : acoes) {
```

```

    acao.executa(nf);
  }
  return nf;
}

```



Membros da Banca:

Avaliador 1 (nome e assinatura)

Avaliador 2 (nome e assinatura)

Avaliador 3 (nome e assinatura)

Presidente da Banca (nome e assinatura)



Assinaturas do documento



Código para verificação: **EOU0B833**

Este documento foi assinado digitalmente pelos seguintes signatários nas datas indicadas:



MARÍLIA GUTERRES FERREIRA (CPF: 011.XXX.530-XX) em 17/08/2026 às 12:41:19

Emitido por: "SGP-e", emitido em 03/05/2019 - 13:37:04 e válido até 03/05/2119 - 13:37:04.

(Assinatura do sistema)



MATTHEUS DA HORA FRANÇA (CPF: 788.XXX.735-XX) em 17/08/2026 às 13:56:08

Emitido por: "SGP-e", emitido em 04/09/2019 - 14:18:37 e válido até 04/09/2119 - 14:18:37.

(Assinatura do sistema)



ADILSON VAHLICK (CPF: 994.XXX.439-XX) em 17/08/2026 às 14:36:46

Emitido por: "SGP-e", emitido em 30/03/2018 - 12:43:42 e válido até 30/03/2118 - 12:43:42.

(Assinatura do sistema)

Para verificar a autenticidade desta cópia, acesse o link <https://portal.sgpe.sea.sc.gov.br/portal-externo/conferencia-documento/VURFU0NfMTlwMjJfMDAwMzE3NDRfMzE3NTFfMjAyNI9FT1UwQjgzMw==> ou o site <https://portal.sgpe.sea.sc.gov.br/portal-externo> e informe o processo **UDESC 00031744/2026** e o código **EOU0B833** ou aponte a câmera para o QR Code presente nesta página para realizar a conferência.