

Data: DD/MM/AAAA
Nome: Fulano de Tal
Assinatura: #####
Matrícula: XX/XXXX-X

EXEMPLO DE PROVA (GABARITO)

Disciplina: Arquitetura & Desenho de Software - Aula de Revisão - Profa. Milene Serrano - Monitores Artur Sousa e Hugo Sobral

Questão Única

Contexto

Um engenheiro de Software foi contratado recentemente para uma empresa de cadastro de pacientes em um hospital.

Para ele, foi conferido um primeiro desafio.

Esse desafio consistiu na aplicação de padrões de projeto, visando solucionar algumas demandas específicas.

Atualmente, no cadastro dos dados do paciente no sistema, os dados numéricos não são formatados adequadamente. Dessa forma, os mesmos são implementados como uma grande string de números.

No caso, são passados os seguintes dados numéricos no cadastro do paciente:

- CPF com formato XXX.XXX.XXX-XX;
- Telefone com formato (XX) XXXXX-XXXX;
- Data de nascimento com formato XX/XX/XXXX;

No backend, o ideal seria que esses dados fossem tratados e formatados com sua própria regra de formatação. Lembrando que para cada dado - seja CPF, Telefone ou Data de Nascimento - há um algoritmo específico para lidar com a formatação necessária. Apenas em tempo de execução, quando o dado é conhecido, é que cabe a execução desse algoritmo.

No intuito de lidar com essa situação, bem como visando isolar os detalhes de implementação e priorizando a escalabilidade, o Engenheiro de Software atuou orientando-se por um padrão de projeto específico para solucionar esse problema.

PARTE I:

O novo funcionário conferiu uma solução bem elegante para isso.

Para tanto, ele estudou bem a demanda, e logo pensou em uma solução orientada a um padrão de projeto da categoria:

() GoFs Criacionais

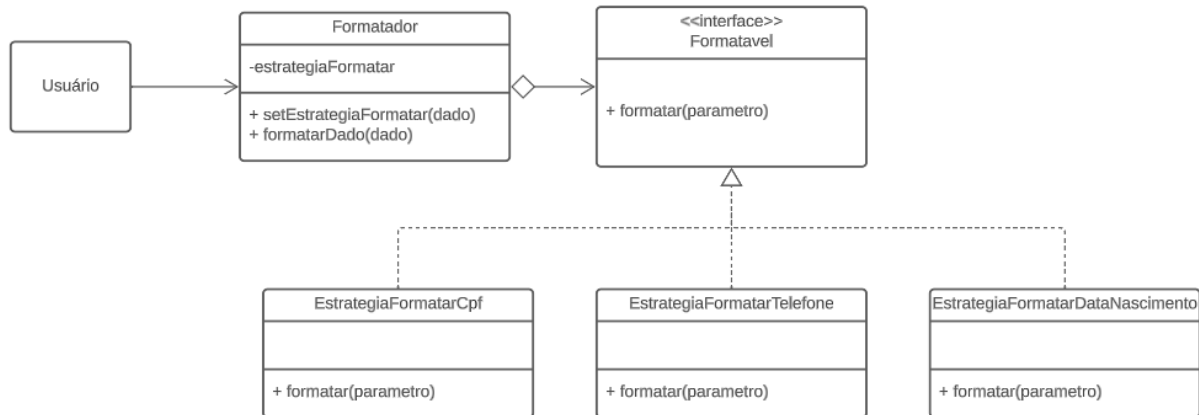
() GoFs Estruturais

(X) GoFs Comportamentais

Qual é o padrão? Strategy

PARTE II:

Além disso, ele pensou na modelagem, e elaborou um diagrama de classes com os principais participantes do padrão de projeto, considerando o Gamma et. al como referência. A seguir (ou no verso), modele o Diagrama de Classes.



PARTE III:

Por fim, ele implementou essa solução. A seguir, escreva o pseudocódigo da programação, SEM OMITIR PASSOS. Caso prefira, pode usar JAVA (exclusivamente essa linguagem).

- **OBS_01:** Não é para usar apenas comentários, omitindo os pontos de complexidade do código.
- **OBS_02:** O código precisa revelar claramente sobre as particularidades do padrão e seus participantes. Caso contrário, infelizmente, a nota será reduzida.

BOA PROVA! :)

```
public interface Formatavel {
    public String formatar(String dado);
}

public class EstrategiaFormatarCPF implements Formatavel {
    public String formatar(String a) {
        // código de formatação para cpf
        // ...
        return aFormatado;
    }
}

public class EstrategiaFormatarTelefone implements Formatavel {
    public String formatar(String a) {
        // código de formatação para telefone
        // ...
        return aFormatado;
    }
}

public class EstrategiaFormatarDataNascimento implements Formatavel {
    public String formatar(String a) {
        // código de formatação para data de nascimento
        // ...
        return aFormatado;
    }
}
```

```

    }
}

public class Formatador {
    private Formatavel formatavel;
    public void setFormatavel(Formatavel formatavel){
        this.formatavel = formatavel;
    }
    public String formatarDado(string a) {
        return formatavel.formatar(a);
    }
}

public class Aplicacao {
    public static void main(String[] args) {

        // Cria um objeto formatador.
        // Lê o dado.
        // Lê o tipo de entrada de dado.

        if (tipo == cpf) {
            formatador.setformatavel(new EstrategiaFormatarCpf());
        }

        if (tipo == telefone) {
            formatador.setformatavel(new EstrategiaFormatarTelefone());
        }

        if (tipo == dataNascimento) {
            formatador.setformatavel(new EstrategiaFormatarDataNascimento());
        }

        var resultado = formatador.formatarDado(dado);

        system.out.print(resultado);
    }
}

```