

DOCUMENTAÇÃO DOS PADRÕES DE PROJETO NO SISTEMA DE GESTÃO HOSPITALAR INTRODUÇÃO

Brena Monike Laurindo - 202220304920 | Edson Luiz Zuchi - 202220302295

Mateus da Silva Reinert - 202010005320

O documento apresenta a documentação dos padrões de projeto Singleton, Builder e Prototype utilizados no sistema de gestão hospitalar. São analisados os métodos e códigos das classes Hospital, Doctor & Nurse, fornecendo a aplicabilidade de cada padrão e justificando sua utilização.

1. Padrão Singleton

O padrão Singleton é utilizado na classe Hospital para garantir que apenas uma instância da classe seja criada e fornecer um ponto global de acesso a essa instância. O objetivo é evitar a criação de múltiplas instâncias do hospital e garantir que todas as partes do sistema acessem a mesma instância.

Justificativa de uso:

Em um sistema de gestão hospitalar, é desejável ter apenas uma instância do hospital em execução, pois todos os dados e recursos relacionados ao hospital devem ser compartilhados e consistentes.

O uso do padrão Singleton garante que as informações dos médicos, enfermeiros e demais entidades relacionadas sejam acessadas de forma coerente em todo o sistema.

Através do método estático *getInstance()*, é possível obter a instância única do hospital e realizar operações sobre ela, como adicionar médicos e enfermeiros, obter informações do hospital, entre outras.

2. Padrão Builder

O padrão Builder é utilizado nas classes Doctor e Nurse para facilitar a construção desses objetos complexos, fornecendo uma interface fluente para definir os atributos dessas classes e criar as instâncias correspondentes.

Justificativa de uso:

As classes Doctor & Nurse possuem atributos que podem variar e exigem a criação de objetos com diferentes combinações desses atributos.

O uso do padrão Builder simplifica a criação desses objetos, pois separa a construção do objeto da sua representação final.

Através dos métodos *name()* e *specialization()* no caso de Doctor, e *name()* e *experienceYears()* no caso de Nurse, é possível definir os valores dos atributos individualmente.

O método *build()* é responsável por criar a instância final da classe, utilizando os valores definidos no processo de construção.

3. Padrão Prototype

O padrão de projeto Prototype é utilizado para criar novos objetos através da clonagem de um objeto existente, evitando a necessidade de criar novas instâncias a partir do zero. No contexto do sistema de gestão hospitalar, o padrão Prototype pode ser aplicado para criar cópias de objetos que representam profissionais da área da saúde, como médicos e enfermeiros.

Justificativa de uso:

A classe **HealthcareProfessional**, apresentada no código fornecido, é uma classe abstrata que serve como base para as classes Doctor & Nurse. Essa classe implementa a interface Cloneable, que indica que objetos dessa classe podem ser clonados.

O método *clone()* na classe HealthcareProfessional é responsável por realizar a clonagem do objeto. Ele chama o método *clone()* da superclasse Object e faz um cast para o tipo HealthcareProfessional. Esse método retorna uma cópia do objeto, preservando o estado atual dos atributos.

Conclusão:

No sistema de gestão hospitalar, foram aplicados três padrões de projeto: Singleton, Builder e Prototype.

O padrão Singleton garante que exista apenas uma única instância da classe Hospital, proporcionando acesso global e controlado aos seus métodos e atributos. Isso evita inconsistências e conflitos de estado, centralizando o acesso a uma instância compartilhada.

O padrão Builder foi utilizado na construção dos objetos Doctor & Nurse, simplificando a criação de classes complexas. Ele oferece uma sintaxe fluente e legível, permitindo a especificação dos atributos desejados e fornecendo um ponto centralizado para a construção dos objetos. Isso melhora a clareza do código e facilita a manutenção, especialmente quando há muitos atributos opcionais ou mutáveis.

Por fim, o padrão Prototype foi empregado para permitir a clonagem dos objetos HealthcareProfessional, representados pelas classes Doctor & Nurse. Através do método clone(), é possível criar cópias dos profissionais de saúde existentes, preservando o estado atual de seus atributos. Isso traz flexibilidade e melhora o desempenho do sistema, evitando a necessidade de recriar instâncias do zero.

Esses padrões de projeto contribuem para um código estruturado, flexível e de fácil manutenção no sistema de gestão hospitalar.