UNIESP CENTRO UNIVERSITÁRIO

CURSO DE SISTEMAS PARA INTERNET

CURSO SISTEMAS DA INFORMAÇÃO

DISCIPLINA: PADRÃO DE PROJETOS

PROFESSORA: ANGELO FRANCESCOLY DIAS GONÇALVES

TURMA: P3

PROJETO PADRÃO PROTOTYPE (PROTÓTIPO)

Alfredo Guilherme Orlando Gracio

Ricardo Augusto Parêdes do Amaral

Yuri Maia Ribeiro

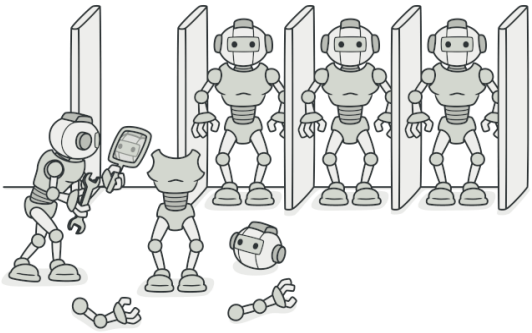
CABEDELO

2023

Introdução ao Padrão de Projeto Prototype

O QUE É

Em sua essência, o Prototype introduz a ideia de clonagem de objetos, permitindo a criação de novas instâncias por meio da duplicação de objetos existentes, os protótipos. Esses protótipos servem como modelos iniciais, definindo as características padrão que novos objetos herdarão sem fazer seu código ficar dependente de suas classes.



QUAL O PROBLEMA

Um dos problemas centrais que o Prototype visa resolver é a necessidade de criar objetos com características comuns, mas com a flexibilidade de adaptação individual.

A utilidade do Prototype revela-se em cenários onde a criação direta de objetos pode ser custosa ou complexa.

*Copiando um objeto "do lado de fora" **nem sempre** é possível.*

QUAL A SOLUÇÃO

A solução essencial proporcionada pelo Prototype reside na introdução do conceito de protótipos. Ao clonar um protótipo, evitamos a sobrecarga de criar objetos do zero, garantindo consistência e eficiência no desenvolvimento.

Esses protótipos são instâncias iniciais, predefinidas com atributos e comportamentos padrão que são compartilhados por todos os objetos que serão criados a partir deles. A chave está na capacidade de clonar esses protótipos, gerando novas instâncias sem a complexidade e a sobrecarga associadas à criação tradicional de objetos.

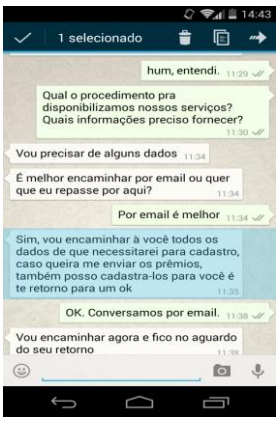
Exemplos de usabilidade do padrão prototype

Editores de Documentos

Considere um editor de documentos como o Microsoft Word. Suponha que você tenha elaborado um documento com um formato específico, fontes e estilos. Se precisar criar um novo documento com a mesma estrutura, seria mais eficiente duplicar o documento existente, mantendo todas as configurações, em vez de começar do zero. Nesse contexto, o documento original funciona como um "protótipo" para os documentos subsequentes.

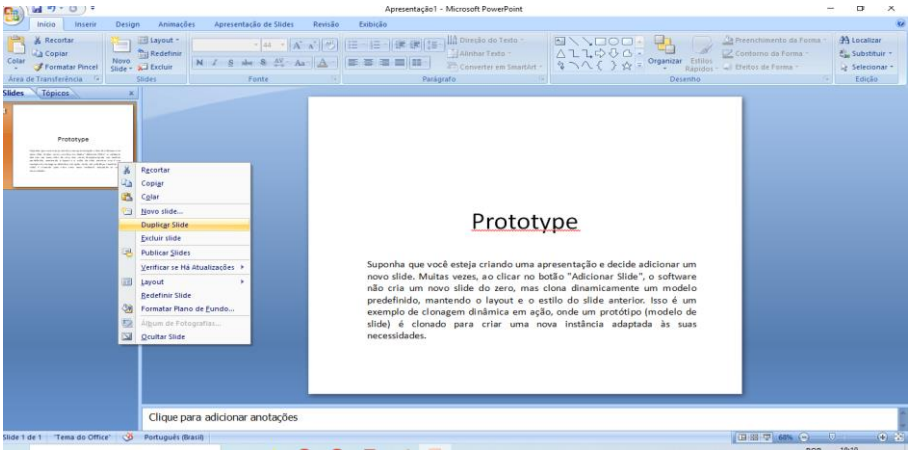
Aplicativos de Mensagens

Se você já enviou uma mensagem semelhante a várias pessoas, pode ter percebido a opção de "Encaminhar" ou "Reutilizar Mensagem". Nesse momento, o aplicativo pode estar usando clonagem dinâmica, permitindo que você envie uma cópia da mensagem original para destinatários diferentes sem reescrever a mensagem toda.



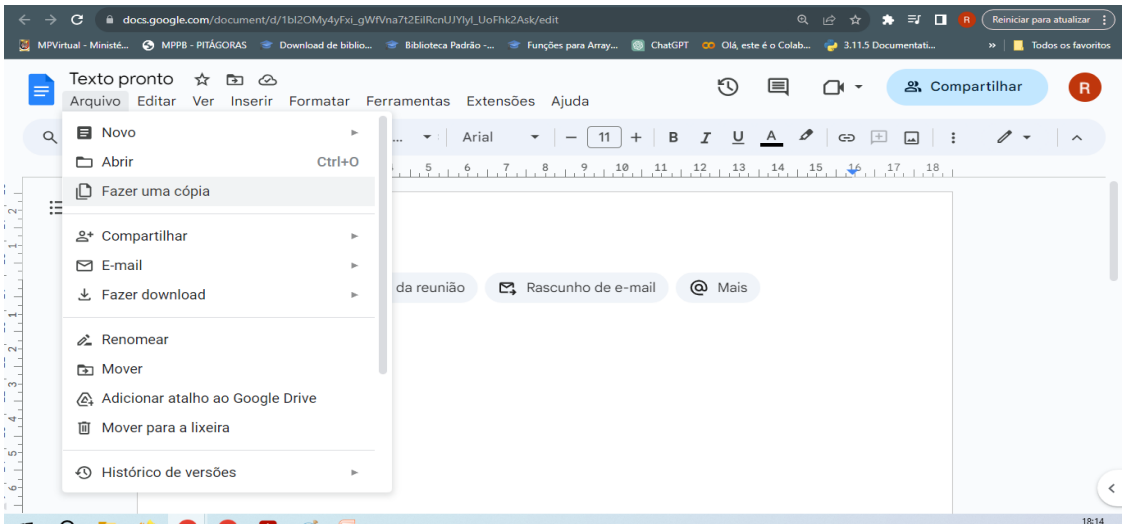
Aplicativos de Edição de Apresentações (como o Microsoft PowerPoint)

Suponha que você esteja criando uma apresentação e decide adicionar um novo slide. Muitas vezes, ao clicar no botão "Adicionar Slide", o software não cria um novo slide do zero, mas clona dinamicamente um modelo predefinido, mantendo o layout e o estilo do slide anterior. Isso é um exemplo de clonagem dinâmica em ação, onde um protótipo (modelo de slide) é clonado para criar uma nova instância adaptada às suas necessidades.



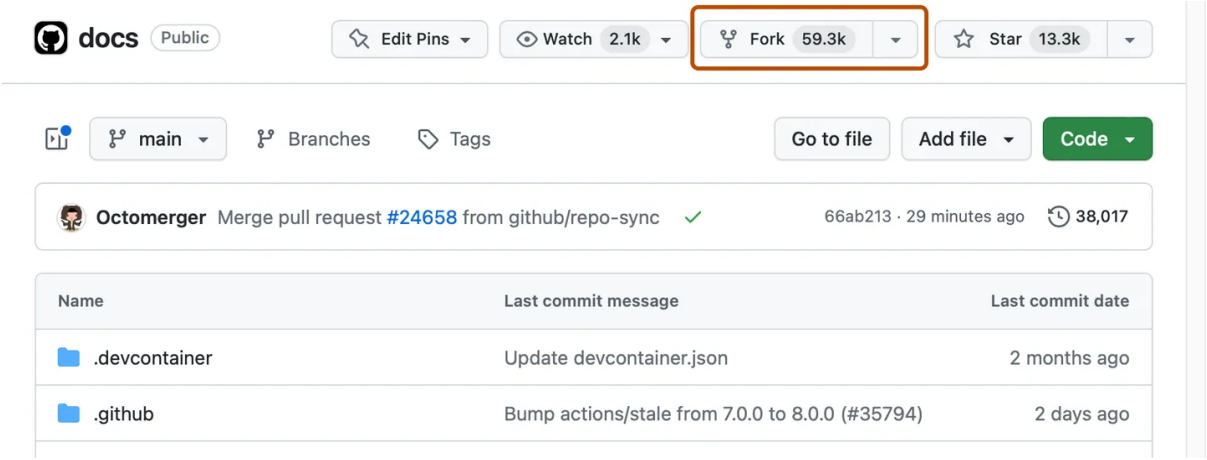
Google Docs

No Google Docs, ao criar um novo documento, você pode perceber que tem a opção de "Fazer uma cópia" de um documento existente. Essa funcionalidade reflete a clonagem dinâmica, onde um protótipo de documento é usado como base para criar uma nova instância.



GitHub

Em plataformas de desenvolvimento colaborativo, como o GitHub, ao criar um novo repositório, você pode ter a opção de "Fork", que cria uma cópia independente de um repositório existente. Isso reflete a ideia de clonagem dinâmica em ambientes de desenvolvimento de software.



Vantagens

Importância da Clonagem Dinâmica

Um aspecto crucial do padrão Prototype é a capacidade de realizar clonagem dinâmica. Isso significa que a clonagem pode ocorrer em tempo de execução, permitindo que novos objetos sejam criados sem a necessidade de especificar suas classes concretas no código. Essa flexibilidade é valiosa, pois possibilita a criação de novas instâncias de objetos durante a execução do programa, adaptando-se dinamicamente às necessidades do sistema.

Em contraste com a abordagem estática da herança, onde as classes dos objetos devem ser conhecidas antecipadamente, a clonagem dinâmica proporciona uma solução mais dinâmica, permitindo que novos tipos de objetos sejam adicionados e utilizados sem a necessidade de alterar o código existente.

Criação de Objetos Complexos de Forma Eficiente

O padrão Prototype permite a criação de novos objetos duplicando-os a partir de objetos existentes. Isso é especialmente útil quando a criação de um objeto é custosa ou complexa, pois em vez de criar um novo objeto do zero, você pode clonar um objeto existente.

Redução de Código Duplicado

Ao utilizar o padrão Prototype, você evita a duplicação de código ao criar objetos sem precisar reescrever a lógica de inicialização de um objeto complexo várias vezes.

Flexibilidade na Criação de Novos Objetos

O padrão Prototype oferece flexibilidade na criação de novos objetos, permitindo alterar e personalizar as propriedades do objeto clonado conforme necessário, mantendo a estrutura básica do objeto original.

Performance Aprimorada

Em comparação com outras abordagens de criação de objetos, o uso de clones pode ser mais eficiente em termos de desempenho, especialmente quando a inicialização de objetos é intensiva em recursos.

Padronização e Organização do Código

Ao adotar um padrão de design como o Prototype, o código tende a ser mais organizado e padronizado, facilitando a manutenção e compreensão do sistema por outros desenvolvedores.

Facilitação de Testes

O padrão Prototype pode facilitar os testes, pois os objetos podem ser facilmente clonados e manipulados em testes unitários ou de integração.

Aplicabilidade em Diversos Contextos

Pode ser utilizado em várias linguagens de programação e em diferentes tipos de sistemas, adaptando-se bem a diversas situações.

Exemplo de código em Python

O código criado como exemplo tenta, de maneira simples, demonstrar o uso do Padrão Prototype. Buscou-se, simular as vantagens de se clonar as características básicas de um usuário, as quais serão clonadas para cada novo usuário e modificadas de acordo com a necessidade ou permissão dada pelo sistema.

usuario_prototype.py

```
class UsuarioPrototype

    def __init__(self):
        self.idade = 18
        self.nome = "Visitante"
        self.tipo_acesso = "Padrão"
        self.limite_download = "10GB"
        self.email = None

    def clone(self):
        return UsuarioPrototype()
```

print_utils.py

```
def imprimir_usuario(usuario, mensagem):
    print(f"{mensagem}:")
    print(f"\nNome: {usuario.nome}\nIdade:
{usuario.idade}\nLimite_download:{usuario.limite_download}\nEmail: {usuario.email}\n")
```

main.py

```
from usuario_prototype import UsuarioPrototype
from print_utils import imprimir_usuario

# Criando uma instância do protótipo
usuario_padrao = UsuarioPrototype()

# Clonando o protótipo para criar um novo usuário
novo_usuario = usuario_padrao.clone()

# Modificando características do novo usuário conforme necessário
novo_usuario.nome = "Novo Visitante"
novo_usuario.limite_download = "5GB"
novo_usuario.email = "novovisitante@example.com"

# Exibindo informações sobre os usuários
```

```
imprimir_usuario(usuario_padrao, "Usuário Padrão (antes da modificação)")
imprimir_usuario(novo_usuario, "Novo Usuário")

# Modificando o usuário padrão após a criação do novo usuário
usuario_padrao.nome = "Visitante Modificado"
usuario_padrao.limite_download = "2GB"
usuario_padrao.email = "visitante_modificado@example.com"

# Exibindo informações sobre os usuários novamente
imprimir_usuario(usuario_padrao, "Usuário Padrão (após a modificação)")
imprimir_usuario(novo_usuario, "Novo Usuário (sem alterações)")
```

Passo a passo do código

O primeiro passo é identificar o objeto que servirá como protótipo. Este objeto base contém as características essenciais que desejamos replicar em novas instâncias.

Definindo a Classe do Protótipo.

Começamos definindo a classe 'UsuarioPrototype' que servirá como nosso protótipo. Ao encapsular as características comuns e o processo de clonagem em uma classe, facilitamos a criação de novos usuários com base no protótipo.

usuario_prototype.py

```
class UsuarioPrototype:
    def __init__(self):
        self.idade = 18
        self.nome = "Visitante"
        self.tipo_acesso = "Padrão"
        self.limite_download = "10GB"
        self.email = None

    def clone(self):
        return UsuarioPrototype()
```

Criando uma instância do protótipo

O protótipo padrão serve como um modelo inicial para novos objetos, fornecendo valores padrão para suas propriedades.

Definir uma interface clara é crucial para a aplicação bem-sucedida do padrão Prototype. A interface determina os métodos e propriedades que as instâncias concretas (os clones) devem implementar.

usuario_padrao = UsuarioPrototype()

Clonando o protótipo para criar um novo usuário

A clonagem evita a necessidade de criar novos objetos a partir do zero, economizando esforço e garantindo consistência nas propriedades.

O passo central do padrão Prototype é a clonagem do protótipo para criar novas instâncias. Em vez de criar novos objetos a partir do zero, o sistema utiliza métodos especializados para duplicar o protótipo, garantindo que as características essenciais sejam preservadas.

novo_usuario = usuario_padrao.clone()

Modificando características do novo usuário conforme necessário

A flexibilidade de personalização é mantida, permitindo que cada usuário seja único, apesar de começar como uma cópia do protótipo.

Após a clonagem, é possível personalizar as novas instâncias conforme necessário. Isso implica ajustar detalhes específicos, como o conteúdo de um documento, o texto de uma postagem, o destinatário de uma mensagem ou os detalhes de um evento. A flexibilidade para personalizar cada instância é uma das vantagens-chave do padrão Prototype.

```
usuario_padrao.nome = "Visitante Modificado"
usuario_padrao.limite_download = "2GB"
usuario_padrao.email = "visitante_modificado@example.com"
```

Exibindo informações sobre os usuários

Demonstrando que os novos usuários foram criados a partir do protótipo e podem ser personalizados.

```
imprimir_usuario(usuario_padrao, "Usuário Padrão (após a modificação)")
imprimir_usuario(novo_usuario, "Novo Usuário (sem alterações)")
```

Conclusão

Em suma, o padrão de projeto Prototype é uma ferramenta poderosa no mundo do desenvolvimento de software, permitindo a criação de objetos complexos de forma eficiente, flexível e escalável. Sua capacidade de clonar objetos existentes para gerar novas instâncias economiza recursos computacionais e simplifica a criação de novos elementos, promovendo a reutilização e a manutenção do código. Ao possibilitar a criação de objetos sem depender de classes específicas, o Prototype oferece uma abordagem versátil e dinâmica para a construção de sistemas, impulsionando a inovação e a adaptação rápida às mudanças.