

Arquitetura de Software

*Padrões de Projetos – Padrões Criacionais
(Continuação)*

Nairon Neri Silva

Prototype

Intenção

- Especificar os tipos de objetos a serem criados usando uma instância protótipo e criar novos objetos pela cópia desse protótipo

Motivação

- Você precisa construir um editor para partituras musicais customizando um framework geral para editores gráficos, acrescentando novos objetos que representam notas, pausas e pentagramas.
- O editor do framework pode ter uma paleta de ferramentas para adicionar estes objetos de música à partitura. A paleta também incluiria ferramentas para selecionar, mover e manipular objetos de música de outra forma.

Motivação

- O usuário clicaria na ferramenta de uma semínima para adicionar semínimas à partitura.
- E também poderia usar a ferramenta de movimentação para mover uma nota para cima ou para baixo nas linhas de pauta, alterando seu registro sonoro.

Motivação

- Vamos considerar que o framework forneça uma classe abstrata **Graphic** para componentes gráficos, como as notas. Além disso fornece uma classe abstrata **Tool** para definir ferramentas como aquelas da paleta.
- O framework também predefine uma subclasse **GraphicTool** para ferramentas que criam instâncias de objetos gráficos e os adicionam ao documento

Motivação

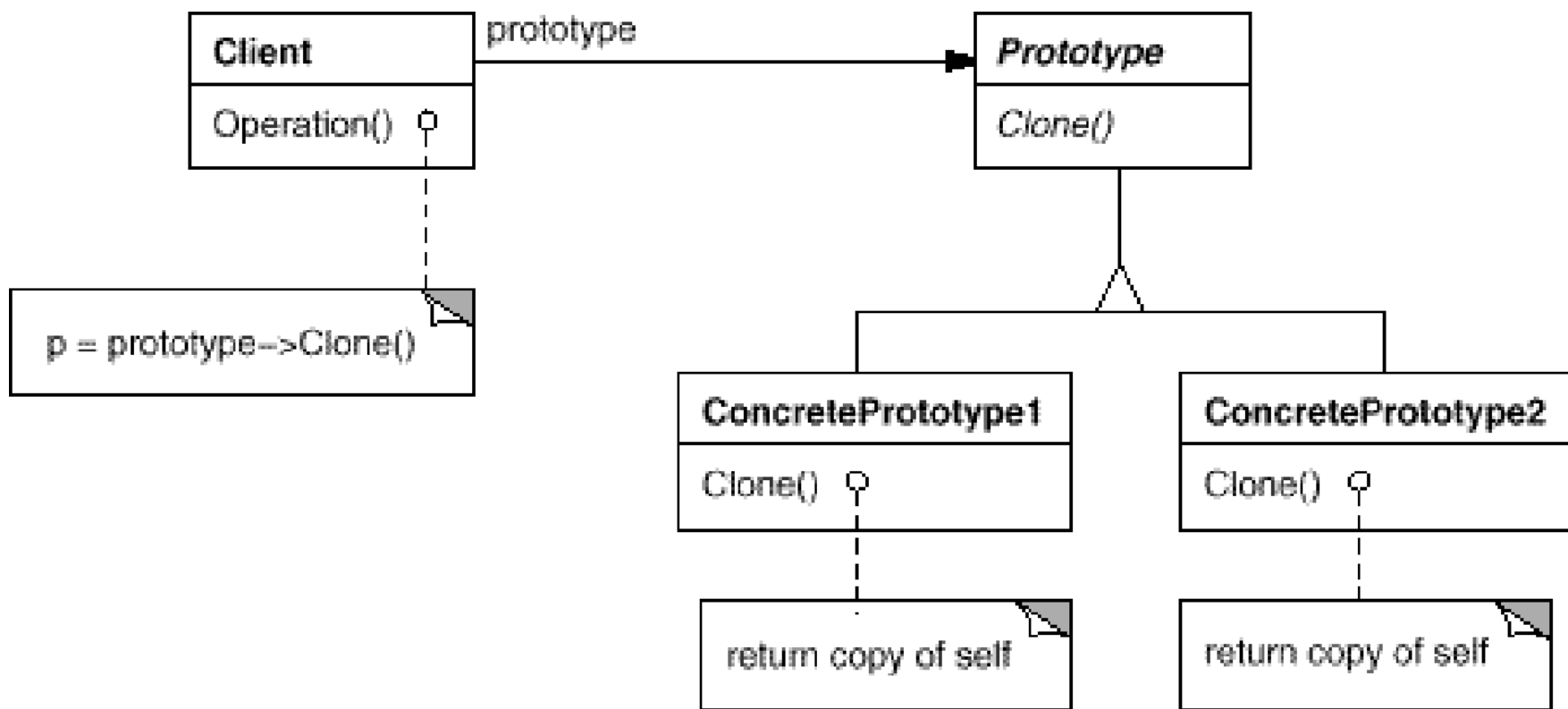
- Mas **GraphicTool** apresenta um problema para o projetista do framework: As classes para notas e pentagramas são específicas da nossa aplicação, mas a classe **GraphicTool** pertence ao framework.
- A classe **GraphicTool** não sabe como criar instâncias das nossas classes musicais para adicioná-las à partitura.
- Poderíamos introduzir subclasses de **GraphicTool** para cada tipo de objeto musical que elas instanciam – muitas classes serão geradas
- A questão, porém, poderia o framework usar composição para parametrizar instâncias de **GraphicTool** pela classe de **Graphic** que se espera que elas criem?

Motivação

- A questão, porém, é: como pode um framework usar composição para parametrizar instâncias de **GraphicTool** pela classe de **Graphic** que se espera que elas criem?
- A solução é fazer **GraphicTool** criar um novo **Graphic** copiando ou clonando uma instância de uma subclasse de **Graphic**. Chamamos esta instância de **protótipo**.
- A **GraphicTool** é parametrizada pelo protótipo que ela deveria clonar e adicionar ao documento. Se todas as subclasses de **Graphic** suportam uma operação Clone, então **GraphicTool** pode clonar qualquer tipo de **Graphic**.

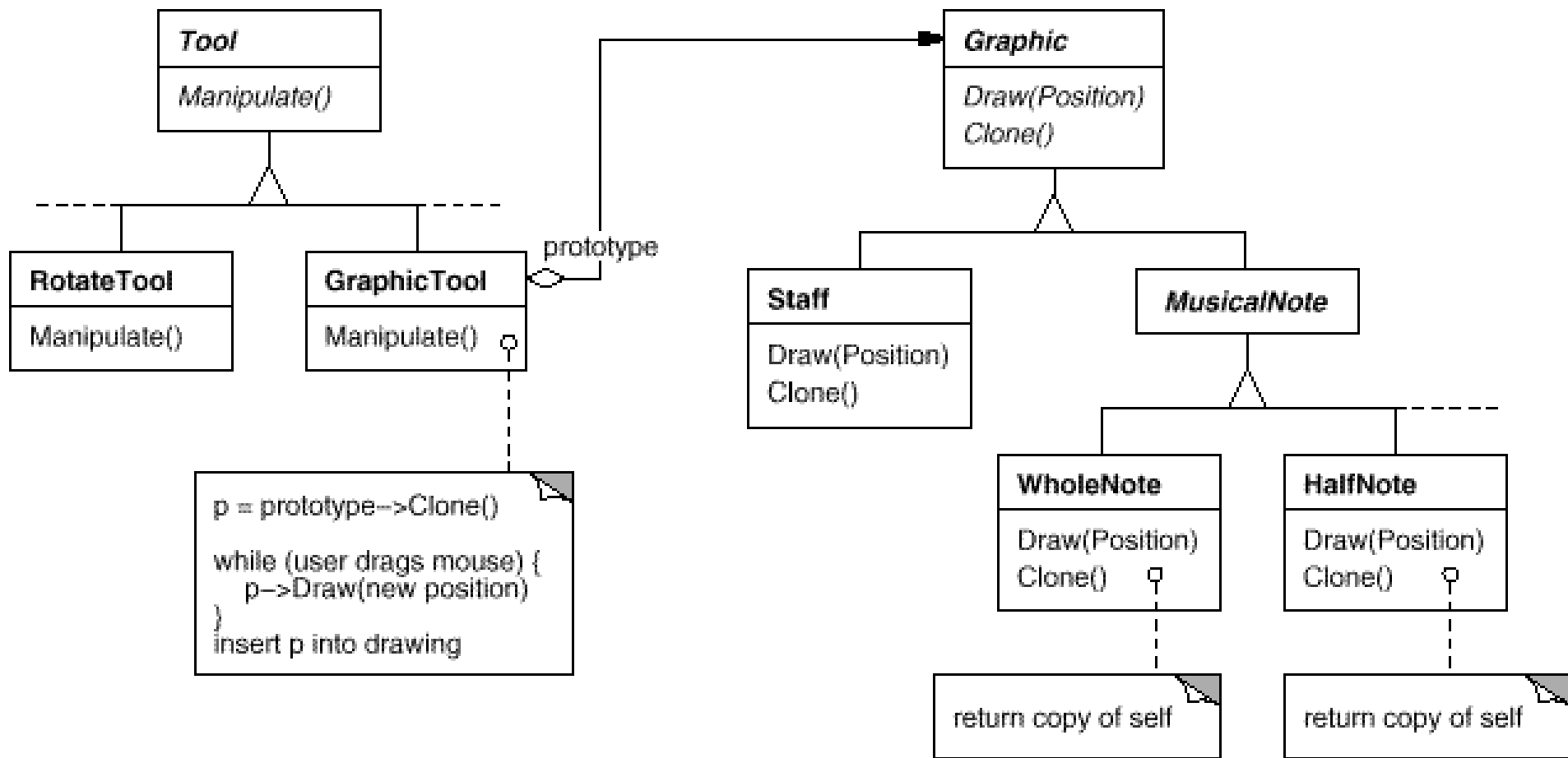
Aplicabilidade

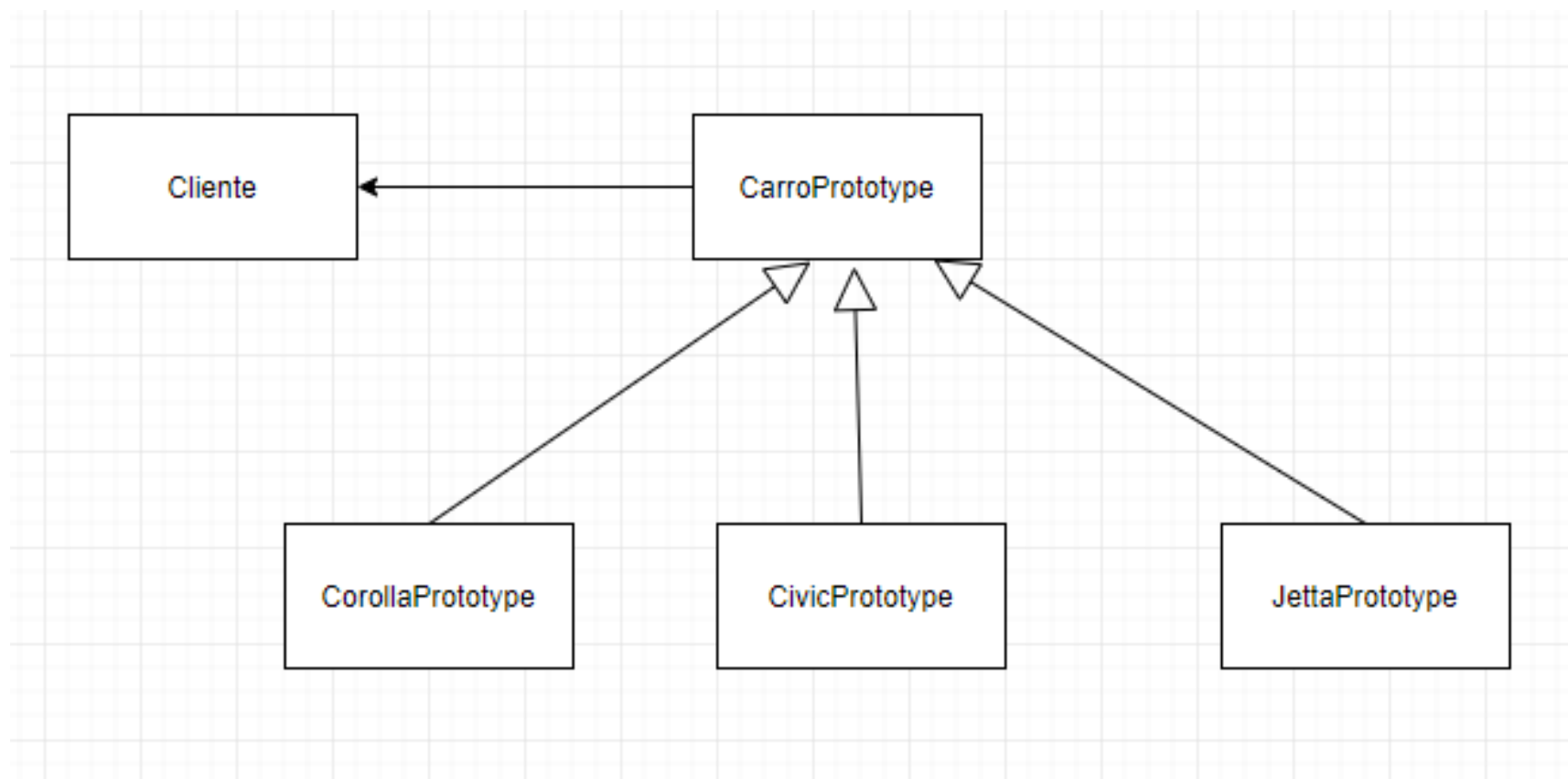
1. Quando as classes a instancia são especificadas em tempo de execução, por exemplo, por carga dinâmica
2. Para evitar a construção de uma hierarquia de classes de fábricas paralela à hierarquia de classes de produto
3. Quando as instâncias de uma classe puderem ter dentre poucas combinações diferentes de estados. Pode ser mais conveniente instalar um número correspondente de protótipos e cloná-los ao invés de instanciar a classe manualmente, cada vez com um estado apropriado



Participantes

- **Prototype** (Graphic)
 - Declara uma interface para clonar a si próprio
- **ConcretePrototype** (Staff, WholeNote, HalfNote)
 - Implementa uma operação para clonar a si próprio
- **Client** (GraphicTool)
 - Cria um novo objeto solicitando a um protótipo que clone a si próprio





Consequências

1. Acrescenta e remove produtos em tempo de execução
2. Especifica novos objetos pela variação de valores
3. Especifica novos objetos pela variação da estrutura
4. Reduz o número de subclasses
5. Configura uma aplicação com classes dinamicamente

Padrões Relacionados

- Prototype e [Abstract Factory](#) competem entre si em várias situações, porém podem ser usados em conjunto
 - Uma [Abstract Factory](#) pode armazenar um conjunto de protótipos a partir dos quais podem ser clonados e retornados os objetos-produto
- Projetos que utilizam os padrões [Composite](#) e [Decorator](#) também podem se beneficiar do uso do Prototype

Saiba Mais...

- <https://refactoring.guru/pt-br/design-patterns/prototype>
- <https://www.youtube.com/watch?v=Z- smcjkdwM>