Padrões de Projeto

Prototype

Gustavo Neri - 4º Período

Intenção

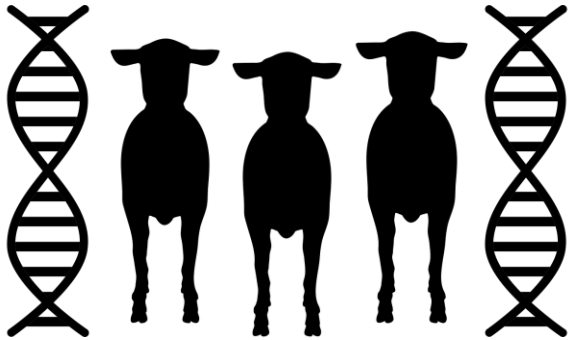
Prototype

- “Especificar tipos de objetos a serem criados usando uma instância protótipo e criar novos objetos pela cópia desse protótipo.”[1]

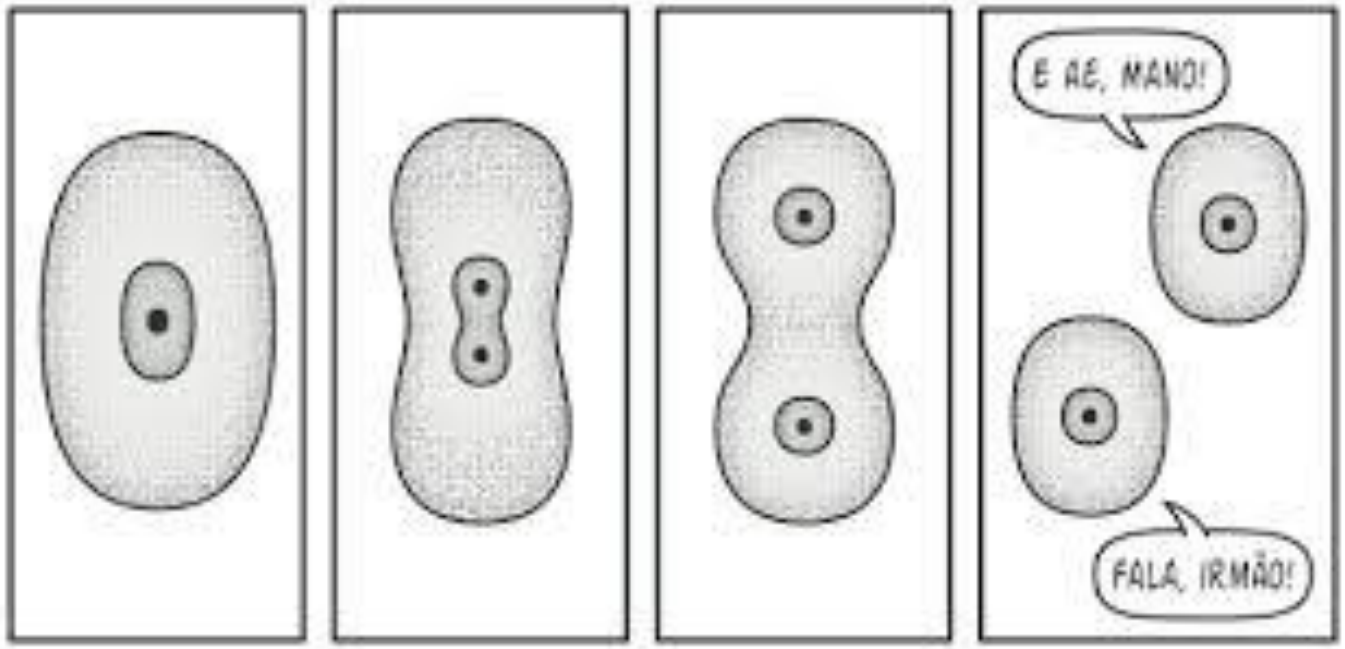
Ou seja

- A partir de um objeto já criado (instância) é possível criar novos objetos apenas clonando-o, sendo uma função do próprio objeto.

Motivação



- Imagine um cientista trabalhando em um laboratório que deseja poder clonar uma ovelha, copiando todo o seu material genético (DNA)
- Ele conseguiu obter o DNA de uma ovelha escolhida e quer cloná-la.
- Abstraindo para sistemas de softwares, o ideal seria que o cientista não necessitasse conhecer todo o banco de dados do DNA escolhido, ele apenas diz para a classe ovelha, por exemplo, que ele deseja clonar a ovelha escolhida.



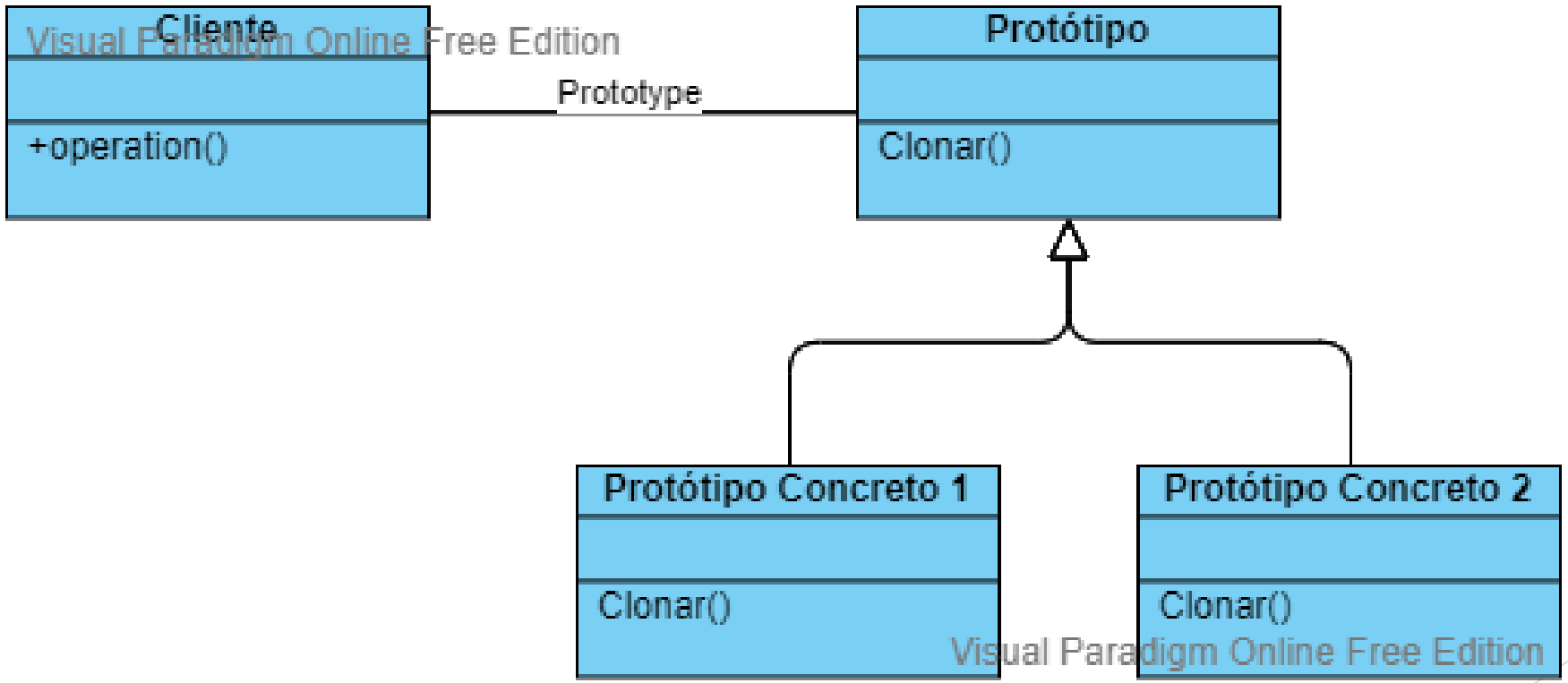
Motivação

A célula original age como um protótipo e tem um papel ativo na criação da cópia

Aplicabilidade

- Quando é necessário a recriação de objetos "caros" (complexos ou que demandam poder de processamento).
- É aplicado quando houver a necessidade de criação de objetos idênticos em tempo real.
- Evitar o conhecimento das classes por parte do cliente.

Estrutura


Cliente
+operation()
Prototype
Protótipo
Clonar()
Protótipo Concreto 1
Clonar()
Protótipo Concreto 2
Clonar()
Visual Paradigm Online Free Edition

Participantes

- **Cliente** -> código que requisita a clonagem do objeto protótipo
- **Prototype** -> uma classe que declara uma interface para objetos capazes de clonar a si mesmo.
- **Prototype Concreto** -> implementação de um *prototype*. *Um objeto protótipo*

Consequências

Vantagem

- Menor custo de processamento em tempo real e menor complexidade
- Oculta classes de outras pessoas
- Flexibilidade

Desvantagem

- Necessita de uma função (clonar) em todos os objetos protótipos
- Pode ser complexo ao clonar objetos que dependem de outros objetos

Implementação

O padrão Prototype deve ser instanciado com um objeto do mesmo tipo em seu construtor.

```
public FiatPrototype(FiatPrototype fiatPrototype) {
    //Cópia dos dados do objeto
}
```

A função clonar deve ser do tipo do objeto para que possa retornar um novo objeto clonado.

```
public FiatPrototype clonar() {
        return new FiatPrototype(this);
}
```

Exemplo de código

https://github.com/gutneri/Padr-o-de-Projeto---Prototype

Curiosidades

- Evita que um cliente ou usuário descubra as classes que criam os objetos.
- Único padrão que utiliza objetos para criar os produtos ao invés de classes.
- Os padrões **Abstract Factory**, **Builder**, e **Prototype** podem todos ser implementados como **Singleton**.
- É possível usar o **Prototype** para compor métodos do padrão **Abstract Factory**

Referências

- [1] GAMMA, Erich et al. Padrões de Projeto: Soluções reutilizáveis de software orientado a objetos.
- https://github.com/luizomf/design-patterns-typescript
- https://github.com/MarcosX/Padr-es-de-Projeto
- https://medium.com/@gbbigardi/arquitetura-e-desenvolvimento-de-software-parte-5-prototype-f010238fbf48