

Prototype

Análise e Desenvolvimento de Sistemas - 6º Período

Cenário

- Em alguns casos desejamos fazer **cópias** de outros objetos
- Além disso, fazer **mudanças** nesses objetos em **tempo de execução**
- Isso pode ser feito por meio de **subclasses**. Mas já sabemos as desvantagens

Cenário

- Objetos "pesados" podem trazer uma complexidade ainda maior, quando precisamos de cópias dos mesmos

Cenário

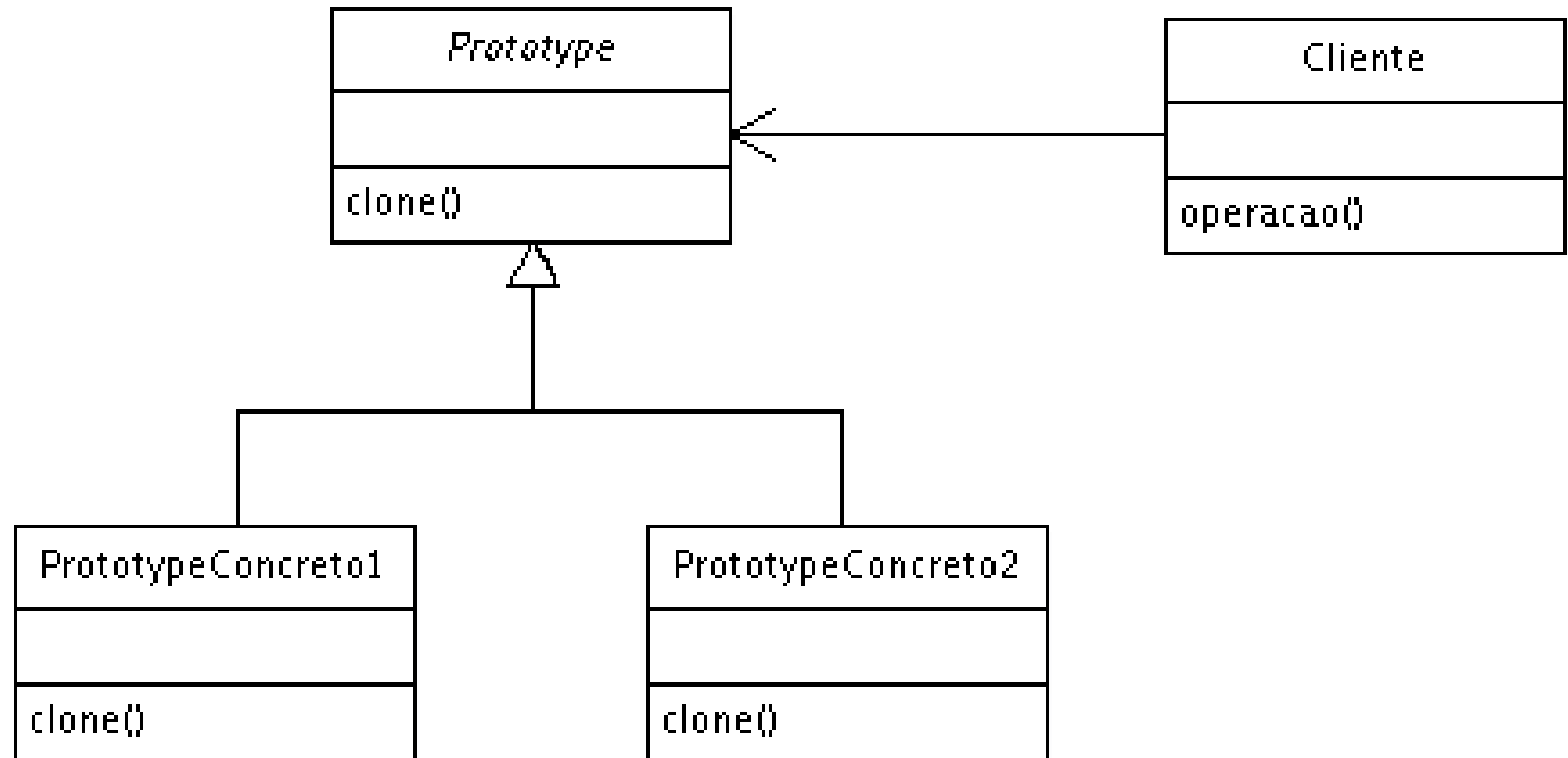
- Imagine que queremos implementar uma função no Portal para exportar relatórios
- Cada aluno pode gerar relatórios de vários tipos: Notas, Faltas, Atividades etc
- Dependendo do formato do relatório, pode ser **caro inicializar o objeto**
- Além disso, o **número de requisições** pode ser grande

Prototype

- Com o Prototype podemos ter **apenas um objeto instanciado que servirá como protótipo para outros**
- Os outros objetos serão **cópias** desses protótipo
- Isso permite que, nesse caso, **apenas uma inicialização seja feita**

Prototype

- Essa abordagem **elimina a necessidade de criação de subclasses**
- Usamos um **registro dos protótipos**
- O client tem **acesso apenas a esse registro e não conhece a classe que implementa o objeto**



Prototype vs Proxy

- A **motivação** muda
- Perceba que o diagrama de classe entre os dois padrões é o mesmo
- No entanto, o **Proxy** nós entrega a capacidade de **controle de acesso**
- Já o **Prototype**, a inicialização de apenas um objeto