# Padrões de design estruturais

### *Bridge*

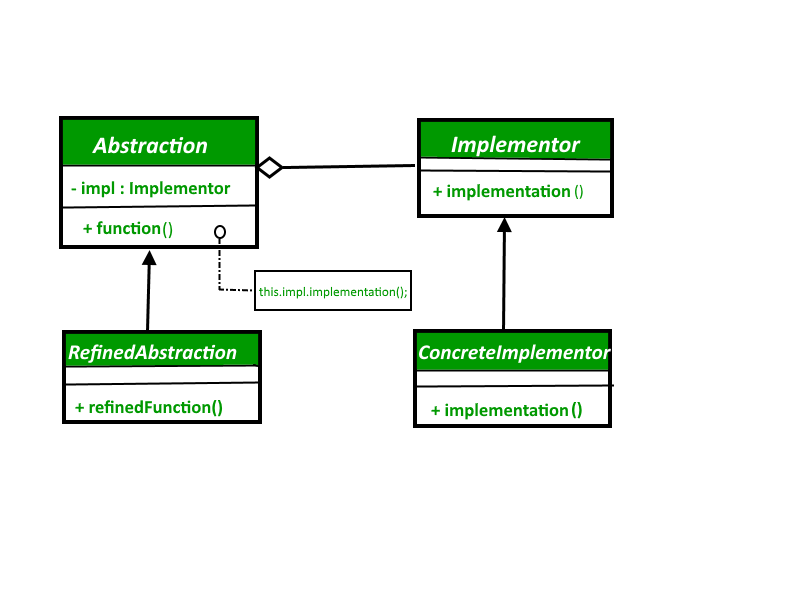
**Sobre Bridge**

* Bridge é um padrão de projeto estrutural que tem o intuito separar uma abstração de suas implementações.
* Na maioria das vezes, são criadas as implementações específicas necessárias de uma abstração usando o conceito de hierarquia. Mas esta aproximação pode não ser adequada quando o número de subclasses começa a aumentar, deixando muito métodos semelhantes a serem reescritos.
* Usa-se então uma *agregação* entre duas interfaces ou classes abstratas, uma parte representando a abstração e a outra a implementação abstrata, quebrando o problema em duas camadas.
* Esta divisão permite que as duas hierarquias sejam diferentes, podendo ser alteradas de formas independentes, garantindo uma maior flexibilidade na implementação de entidades concretas diferentes.
* A implementação se torna, pois, uma parte da abstração, podendo ser expandida para cobrir as possíveis demanda de seu todo, sem que seja necessário criar uma nova estrutura para cada alteração no sistema.

**Aplicações de bridge**

* A técnica de bridge é recomendada quando uma classe deve suportar alga variabilidade não só no conceito de o que é, mas também no seu comportamento, o que implicaria em um grande número de subclasses. Algumas das vantagens de bridge são:
  + Maior flexibilidade ao conceito de herança, evitando demasiadas repetições na criação de novas subclasses.
  + A possibilidade de expandir tanto a abstração quanto a implementação, fornecendo diferentes combinações entre essas subclasses por agregação.
  + Independência entre o desenvolvimento das duas hierarquias de generalização diferentes.
  + Dispensa da necessidade de reescrever métodos abstratos para cada subclasse nova sem que seja realmente preciso.

**Diagrama UML de bridge**

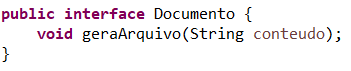


**Diagrama de classes de bridge**

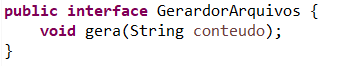
* *Abstraction:* descreve o comportamento geral da interface da abstração. Contém uma referência para um objeto da implementação.
* *Implementor:* definição da interface da classe de implementação. Não necessáriamente deve corresponder a interface da abstração.
* *RefinedAbstraction:* extensão da interface da abstração, garantindo mais uma camada de especialização.
* *ConcreteImplementor:* definição concreta da interface de implementação.

**Exemplo de bridge**

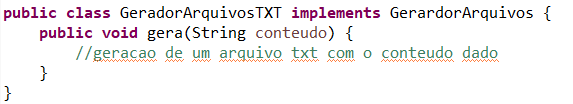
* A seguir, é apresentado um exemplo de bridge, no contexto de um gerador de documentos em diferentes formatos.
  + Cria-se a interface da parte de abstração, Documento

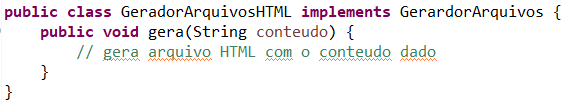


* + Depois, é definida a interface da implementação de um gerardor de arquivos nos formatos desejados, dividindo as tarefas da criação de documentos.



* + Podem ser criadas diferentes subclasses da ferramenta de implementação





* + Elementos mais específicos da implementação podem são definidos então, contendo um objeto da implementação.

