

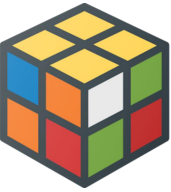


Geek University

Evolua seu lado geek!

www.geekuniversity.com.br

Introdução ao padrão de projeto Command



pythonTM

Design Patterns



Introdução ao padrão de projeto Command

Já aprendemos que os padrões de projetos comportamentais têm como foco as responsabilidades de um objeto, pois eles lidam com a interação entre objetos para alcançar funcionalidades mais complexas.

O padrão Command é um padrão de projeto comportamental, em que um objeto é usado para encapsular todas as informações necessárias para executar uma ação ou disparar um evento posteriormente. Estas informações incluem:

- O nome do método;
- Um objeto que seja dono do método;
- Valores para os parâmetros do método.

Introdução ao padrão de projeto Command

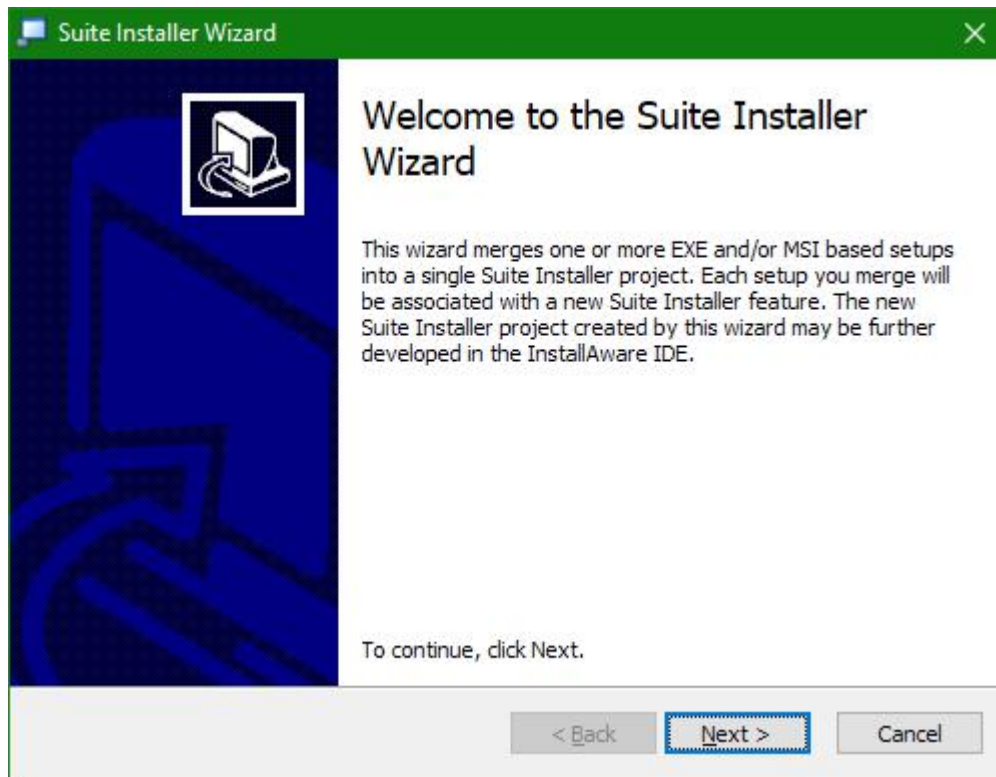


Contextualizando o padrão Command



Introdução ao padrão de projeto Command

Contextualizando o padrão Command



Para entender o padrão Command, considere o caso de um assistente (wizard) de instalação.

Um assistente típico pode ter várias etapas ou telas que capturem preferências de um usuário.

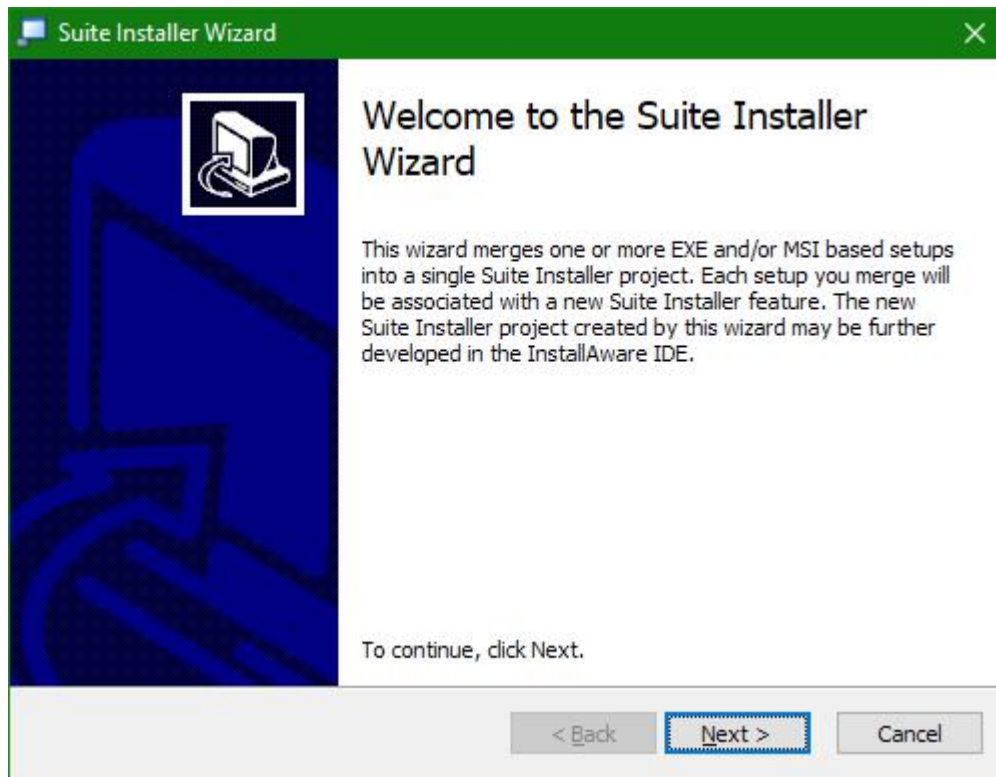
Enquanto navega pelo assistente, o usuário faz determinadas escolhas.

Os assistentes geralmente são implementados com o padrão Command.



Introdução ao padrão de projeto Command

Contextualizando o padrão Command



Um assistente é iniciado com um objeto chamado Command.

As preferências ou escolhas feitas pelo usuário nas várias etapas do assistente são armazenadas no objeto Command.

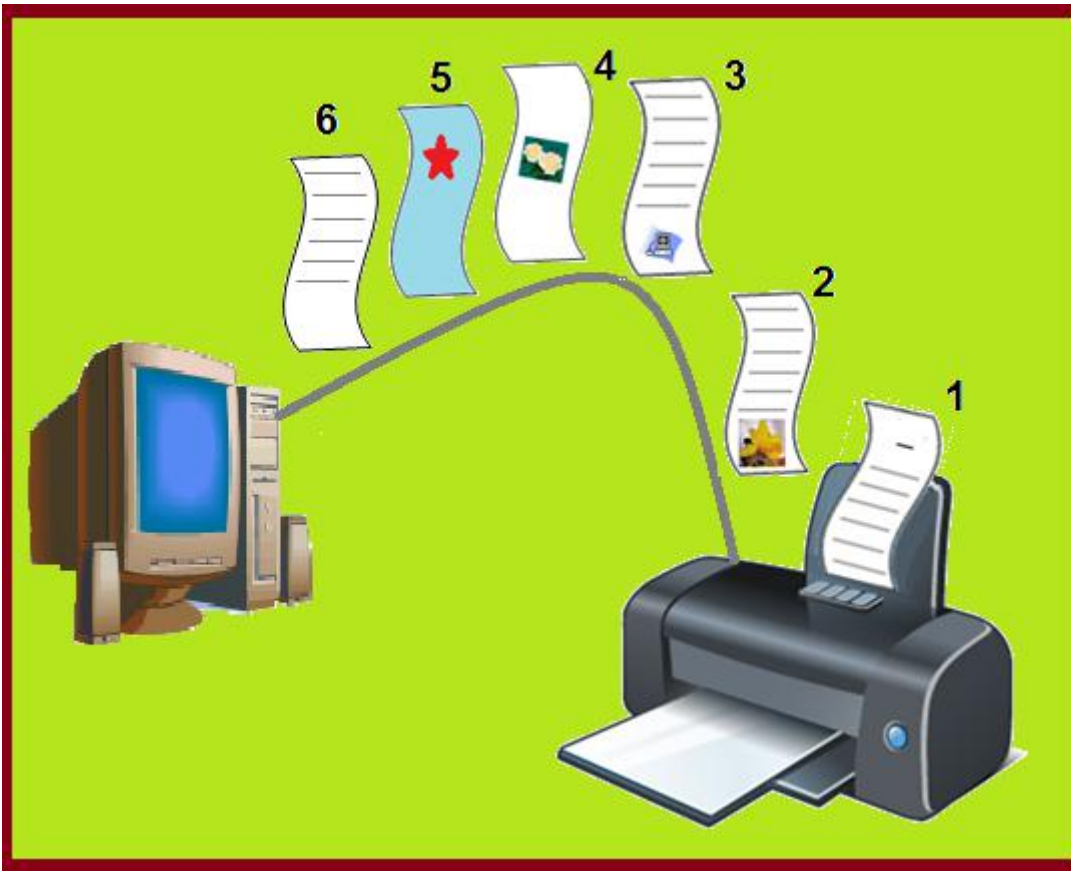
Quando o usuário clica no botão “Fim” (Finish) na última tela do assistente, o objeto Command executa um método *execute()* que considera todas as opções armazenadas e faz o procedimento de instalação apropriado.

Assim todas as informações relacionadas às escolhas são encapsuladas em um objeto que pode ser usado mais tarde para executar uma ação.



Introdução ao padrão de projeto Command

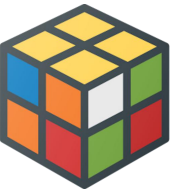
Contextualizando o padrão Command



Outro exemplo simples é o Spooler de impressão.

Um spooler pode ser implementado como um objeto Command que armazena informações como tipo de página (A4, A5, etc), retrato/paisagem, agrupado/não agrupado, etc.

Quando o usuário imprimir algo, por exemplo um documento ou imagem, o spooler executará o método *execute()* do objeto Command e o documento ou imagem será impressa com as preferências definidas.



Introdução ao padrão de projeto Command

Compreendendo o padrão Command

O padrão Command trabalha com os seguintes termos:

- **Command** -> Um objeto Command conhece os objetos Receiver e invoca um método deste objeto;
- **Receiver** -> Os valores dos parâmetros do método receptor (receiver) são armazenados no objeto Command;
- **Invoker** -> O chamados ou invocador (invoker) sabe como executar um comando;
- **Client** -> O cliente cria um objeto Command e define o seu receptor;



Introdução ao padrão de projeto Command

Os principais objetivos do padrão Command são:

- Encapsular uma requisição como um objeto;
- Possibilitar a parametrização dos clientes com diferentes requisições;
- Permitir salvar as requisições em uma fila;
- Oferecer uma callback orientada a objetos;



Introdução ao padrão de projeto Command

O padrão Command pode ser usado em vários casos, incluindo:

- Parametrizar objetos de acordo com a ação a ser executada;
- Adicionar ações em uma fila e executar as requisições em pontos diferentes;
- Criar uma estrutura para operações de alto nível baseadas em operações menores;



Introdução ao padrão de projeto Command

O padrão Command pode ser usado em vários casos, incluindo:

- Parametrizar objetos de acordo com a ação a ser executada;
- Adicionar ações em uma fila e executar as requisições em pontos diferentes;
- Criar uma estrutura para operações de alto nível baseadas em operações menores;

Vamos ao código fazer uma implementação simples deste padrão...



Geek University

Evolua seu lado geek!

www.geekuniversity.com.br