THAMIRIS ADRIANO

PDS TECH

FULL STACK DEVELOPMENT

TYPESCRIPT

AULA 05

SUMÁRIO

O QUE VEM POR AÍ?

Você já se perguntou como os programas de computador podem se tornar mais eficientes, flexíveis e fáceis de manter? A resposta pode estar na Inversão de Controle (IoC) e na Injeção de Dependência.

Nessa aula, exploraremos como a Inversão de Controle transforma a programação tradicional, passando o controle das tarefas para um sistema externo como um maestro que rege uma orquestra. Aprenderemos como IoC permite que um "container" ou sistema externo gerencie o ciclo de vida dos componentes, ao invés da pessoa desenvolvedora precisar definir rigidamente o fluxo do programa. Além disso, vamos mergulhar na Injeção de Dependência, uma técnica onde as dependências são fornecidas externamente.

Ao final, você entenderá como esses conceitos não apenas simplificam o gerenciamento de código, mas também abrem portas para sistemas mais adaptáveis e sustentáveis no mundo da programação.

HANDS ON

Nesse vídeo, exploramos a Injeção de Dependência de uma maneira envolvente e aplicada. Vamos analisar exemplos reais dessa técnica em uso, identificando como ela pode ser integrada em diferentes tipos de projetos. O foco será entender a Injeção de Dependência em um contexto prático, observando seus impactos na melhoria da estrutura e eficiência dos projetos. Será uma oportunidade valiosa para visualizar os benefícios da Injeção de Dependência em ação.

SAIBA MAIS

Inversão de controle

Inversão de Controle (IoC) é uma maneira de organizar programas de computador nos quais não é a pessoa desenvolvedora quem determina diretamente a ordem e o modo como os métodos são chamados. Em vez disso, essa responsabilidade é passada para uma parte específica do software, geralmente chamada de "container", ou outro componente, que gerencia quando e como certas partes do programa são executadas. Isso é diferente da programação tradicional, onde o(a) programador(a) tem controle total sobre o fluxo do programa.

Imagine um programa de computador como uma série de tarefas. Normalmente, seu código controla a ordem e o modo como essas tarefas são executadas. A Inversão de Controle muda isso: ao invés de seu código gerenciar tudo, um sistema externo toma as decisões de controle. Isso é como ter um maestro regendo uma orquestra, onde cada instrumentista (parte do seu código) só toca quando o maestro indica.

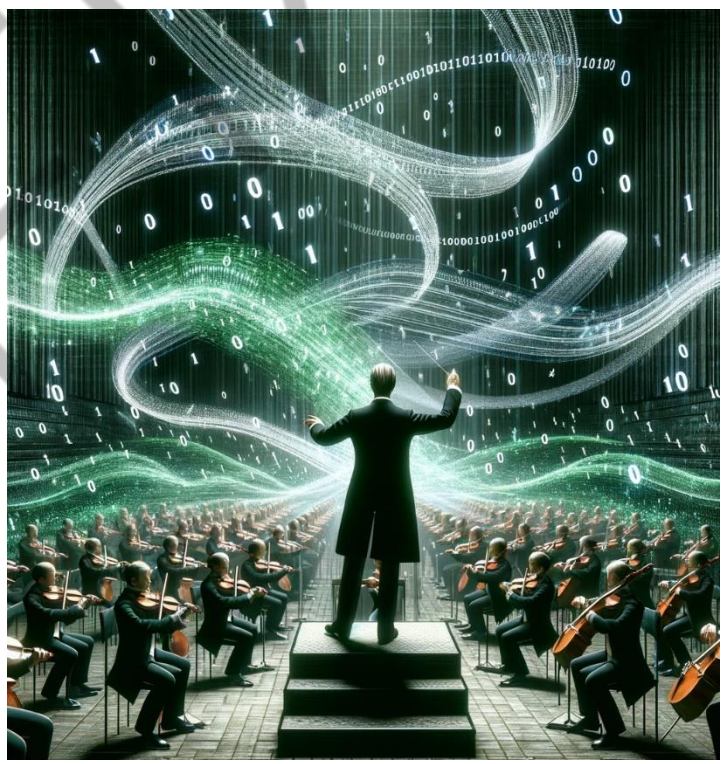

Figura 1 — Maestro regendo orquestra

Fonte: Dall-e (2023)

Em Programação Tradicional, normalmente um programa chama outros programas ou componentes conforme necessidade, controlando quando eles começam e terminam. A pessoa desenvolvedora escreve o código especificando exatamente o que acontece e quando. Com a Inversão de Controle, em vez de especificar todas as chamadas e gerenciar o ciclo de vida dos componentes, o(a) desenvolvedor(a) entrega parte desse controle a um sistema externo. Esse sistema decide quando chamar certas partes do programa.

Exemplos de Inversão de Controle

- **Programação orientada a eventos:** o programa responde a eventos externos (como cliques de botão) em vez de seguir um fluxo pré-definido.
- **Injeção de dependência:** os componentes do programa recebem suas dependências de um sistema externo, em vez de criá-las por conta própria.
- **Programação funcional:** enfatiza o uso de funções onde o controle de execução é frequentemente gerenciado pelo próprio framework de programação.

Em resumo, a Inversão de Controle muda a forma como o fluxo do programa é gerenciado, passando algumas responsabilidades, que normalmente seriam do(a) programador(a), para uma parte externa do software. Isso pode ajudar a tornar os programas mais flexíveis e mais fáceis de manter.

Injeção de dependência

Se imagine preparando o café da manhã em sua cozinha. Você precisa de vários itens: torradeira, cafeteira, suco, pão, café, etc. Agora, pense na cozinha como um sistema e em cada um desses itens como componentes desse sistema.

- **Sem Injeção de Dependência**: neste cenário, você tem uma cozinha onde todos os componentes estão fixos. A torradeira só funciona com um tipo específico de pão, a cafeteira só faz um tipo de café, etc. Você não poderá ter um tipo diferente de pão ou café, pois sua cozinha não está configurada para aceitar diferentes "dependências". Isso é

semelhante a ter um código rígido, onde componentes dependem diretamente de implementações específicas.

- **Com Injeção de Dependência**: agora, imagine uma cozinha modular, onde você pode usar qualquer torradeira, cafeteira ou ingredientes que preferir, escolhendo o que usar com base em suas necessidades ou preferências do dia. Neste caso, sua cozinha é configurada para aceitar diferentes "dependências" (tipos de torradeiras, cafeteiras, ingredientes). Isso é análogo a um sistema de código onde as dependências (como serviços ou componentes) são injetadas. Você pode facilmente substituir uma torradeira por outra sem alterar o restante da sua cozinha.

Aplicando no mundo da programação

Da mesma forma, em programação, a Injeção de Dependência permite que você plugue diferentes componentes (como serviços de log, conexões de banco de dados, etc.) em suas classes ou módulos, sem alterar o código dessas classes ou módulos.

Isso proporciona grande flexibilidade e facilita a manutenção, pois você pode mudar as dependências sem necessidade de modificar o código que as utiliza.

Assim, a Injeção de Dependência, tanto na cozinha quanto na programação, é sobre criar sistemas flexíveis e facilmente modificáveis, permitindo a substituição de componentes sem afetar o funcionamento geral.

O QUE VOCÊ VIU NESTA AULA?

Nesta aula, exploramos como a inversão de controle (IoC) e a injeção de dependência (DI) redefinem a programação, promovendo uma abordagem mais modular e flexível. Ao invés dos componentes do programa gerenciarem suas próprias dependências e fluxos, a IoC transfere essa responsabilidade para um “container” ou framework externo. A injeção de dependência, um dos métodos de implementar IoC, envolve fornecer os componentes do programa com suas dependências de forma externa, geralmente através de construtores, métodos ou propriedades. Isso contrasta com a abordagem tradicional, onde o fluxo do programa e a criação de dependências são rigidamente controlados pela pessoa programadora, resultando em um acoplamento mais forte e menor flexibilidade. Com esses conceitos, agora você pode tornar seus códigos mais modulares, testáveis e fáceis de manter, observando melhorias significativas na estrutura e eficiência dos seus projetos de programação.

REFERÊNCIAS

FREECODECAMP**. Uma rápida introdução à injeção de dependências: o que é e quando usá-la.** Disponível em: https://www.freecodecamp.org/portuguese/news/uma-rapida-introducao-a-injecao-de-dependencias-o-que-e-e-quando-usa-la/. Acesso em: 24 jan. 2024.

TREINAWEB. **Entendendo injeção de dependência.** Disponível em: https://www.treinaweb.com.br/blog/entendendo-injecao-de-dependencia. Acesso em: 24 jan. 2024.

DEVMEDIA. **Introdução ao padrão Injeção de Dependências.** Disponível em: https://www.devmedia.com.br/introducao-ao-padrao-injecao-de-dependencias/28121. Acesso em: 24 jan. 2024.

PALAVRAS-CHAVE

Palavras-chave: Typescript. Injeção de dependência. IoC.