



Linguagens Compiladas e Interpretadas

Conceitos, Diferenças e Aplicações

Objetivo da Apresentação:

- Explicar conceitos de linguagens compiladas e interpretadas.
- Análise das principais diferenças entre linguagens compiladas e interpretadas.
- Demonstração de aplicações práticas das linguagens compiladas e interpretadas.
- Compreensão essencial para a escolha da linguagem mais adequada para projetos específicos.





Conceitos Básicos - Linguagens Compiladas

Tópico	Descrição
Definição	Tradução de código-fonte para código de máquina via compilador. Produz arquivo executável antes da execução.
Exemplos	C, C++, Rust, Go.
Processo de Compilação	Escrita do Código: Desenvolvedor escreve código-fonte. Compilação: Compilador verifica e traduz o código. Execução: Executável rodado pelo SO.
Vantagens	Alta performance. Erros detectados antes da execução. Segurança do código executável.
Desvantagens	Desenvolvimento mais lento. Menos flexibilidade para mudanças rápidas.

Conceitos Básicos - Linguagens Interpretadas

Tópico	Descrição
Definição	Código-fonte executado por um interpretador, linha por linha, em tempo de execução, sem gerar arquivo executável previamente.
Exemplos	Python, JavaScript, PHP, Ruby.
Processo de Compilação	Escrita do Código: Desenvolvedor escreve código-fonte. Interpretação: Interpretador lê e executa o código linha por linha. Execução: Código é executado imediatamente após a interpretação.
Vantagens	Maior flexibilidade para mudanças rápidas. Desenvolvimento ágil. Facilidade para depuração.
Desvantagens	Performance geralmente inferior às linguagens compiladas. Dependência do interpretador para a execução.



Compiladas VS Interpretadas

Tópico	Linguagens Compiladas	Linguagens Interpretadas
Performance	Código convertido em código de máquina otimizado antes da execução, resultando em melhor performance e execução mais rápida.	Código traduzido em tempo real pelo interpretador, o que pode causar lentidão devido ao overhead durante a execução.
Flexibilidade e Debugging	Erros são detectados durante a compilação, tornando o ciclo de desenvolvimento mais rígido, mas garantindo código livre de certos erros.	Maior flexibilidade e facilidade para modificar e testar código sem recompilação, facilitando a depuração em tempo real.
Portabilidade	Código compilado é específico para a plataforma de compilação, exigindo recompilação para rodar em diferentes plataformas.	Mesmo código pode rodar em diferentes plataformas com a presença de um interpretador compatível, oferecendo maior portabilidade.
Exemplos de Aplicações	Usadas em sistemas onde performance é crítica: sistemas embarcados, softwares de tempo real, jogos, sistemas operacionais.	Preferidas para desenvolvimento web, scripts de automação, e prototipagem rápida, valorizando flexibilidade e rapidez no desenvolvimento.

Aplicações Práticas

- Quando escolher uma linguagem compilada ou interpretada?
- **Linguagens Compiladas:**
 - Preferíveis quando a performance é uma prioridade.
 - Ideais quando o controle sobre o hardware é necessário.
- **Linguagens Interpretadas:**
 - Ideais para projetos que exigem rapidez no desenvolvimento.
 - Preferíveis em cenários que requerem flexibilidade ou portabilidade.
- **Impacto no Desenvolvimento e Manutenção de Projetos**
- A escolha entre linguagem compilada ou interpretada pode afetar:
 - **Eficiência do desenvolvimento:** Agilidade no processo de codificação e testes.
 - **Facilidade de manutenção:** Simplicidade na depuração e atualização do código.
 - **Escalabilidade do projeto:** Capacidade de crescimento e adaptação do sistema.
- **Consideração do Contexto:** Avaliar o contexto do projeto e os requisitos específicos para tomar a melhor decisão.



Recapitulando

- **Conceitos principais discutidos:**

- Definição de linguagens compiladas e interpretadas.
- Diferenças em performance, flexibilidade, e portabilidade.
- Exemplos de aplicações práticas.

- **Escolha da Linguagem:**

- Deve ser baseada nas necessidades do projeto.
- Considerar os trade-offs entre performance, flexibilidade e portabilidade.



Encerramento

