



Lógica de Programação

Resolução de Problemas

Nesta Aula

- Identificação do Problema
- Análise de Cenários
- Análise Crítica
- Linguagens de Programação
- Frameworks, Bibliotecas e APIs
- Documentação e Modularização
- Características e Semântica



Resolução de Problemas

1. Identificação do Problema

- **Definição:** Identificar e definir claramente o problema é o primeiro passo crucial na resolução de problemas em programação.
- **Exemplo:** Suponha que temos um problema de calcular a média de notas de alunos em uma turma.

Resolução de Problemas

2. Análise de Cenários

- **Definição:** Analisar diferentes situações e contextos relacionados ao problema para entender suas ramificações e possíveis soluções.
- **Exemplo:** Considerar diferentes formatos de entrada de dados, como notas individuais ou um arquivo de texto com todas as notas.

Resolução de Problemas

3. Análise Crítica

- **Definição:** Avaliar rigorosamente as soluções propostas para determinar sua eficácia e identificar possíveis melhorias.
- **Exemplo:** Avaliar se a solução proposta para calcular a média das notas é eficiente e se pode ser otimizada de alguma forma.

Introdução à Programação

Linguagens de Programação

O que é linguagem de programação?

É uma linguagem formal que, através de uma série de instruções, permite que um programador escreva um conjunto de ordens, ações consecutivas, dados e algoritmos para criar programas que controlam o comportamento físico e lógico de uma máquina.



Frameworks, Bibliotecas e APIs

Na programação, o **framework** é como uma caixa de ferramentas pronta. Ele tem pedaços de código já feitos para coisas comuns, como mostrar textos na tela ou organizar botões. O programador pega essas ferramentas prontas e encaixa com o seu próprio código para fazer o programa funcionar, sem precisar escrever tudo do zero. Isso deixa o trabalho mais rápido e evita erros.

Uma **biblioteca** de programação é como uma biblioteca de livros, mas em vez de livros, ela tem código pronto para você usar em seus programas.

Uma **API (Application Programming Interface)** é como um garçom para software. Ela permite que dois programas se comuniquem entre si. Um programa faz um pedido à API, e a API retorna o que o programa precisa.

Documentação e Modularização

A **documentação** é como um mapa para o seu código. Ela ajuda você e outros programadores a entenderem o que o código faz, como ele funciona e como usá-lo. Uma boa documentação pode:

Ajudar a entender o código: A documentação explica o que o código faz, por que ele foi escrito e como ele funciona.

Ajudar na manutenção do código: A documentação pode facilitar a manutenção do código, pois fornece informações sobre como o código funciona e como ele pode ser modificado. Isso pode economizar tempo e esforço no futuro.

Facilitar o uso do código: A documentação pode fornecer instruções sobre como usar o código, incluindo quais funções e classes estão disponíveis, como usá-las e quais parâmetros elas esperam. Isso pode economizar muito tempo e frustração para você e outros programadores.

Melhorar a qualidade do código: O processo de escrever documentação pode ajudá-lo a pensar mais claramente sobre o seu código e a identificar áreas que podem ser melhoradas. Isso pode levar a um código mais limpo, eficiente e robusto.

Como Modularizar o Código em Módulos Independentes

Modularizar o código significa dividi-lo em partes menores e independentes, chamadas de módulos. Cada módulo deve ter uma única responsabilidade e ser bem documentado. Isso pode tornar o código mais fácil de entender, manter e reutilizar.

Características e Semântica



Definição: Identificação das características únicas de cada linguagem de programação e compreensão da semântica das linguagens.

Exemplo: Python é conhecido por sua sintaxe limpa e legível, enquanto C é conhecido por sua eficiência e controle de baixo nível.

Conclusão



Nesta aula, exploramos o processo de resolução de problemas em programação e introduzimos os alunos a diferentes linguagens de programação e suas características. É essencial entender como identificar problemas, analisar soluções e escolher a melhor linguagem e ferramentas para implementar essas soluções de forma eficiente e eficaz. Na próxima aula, continuaremos nossa jornada explorando os diferentes paradigmas de programação.