



Lógica de Programação

Paradigmas de Programação

Os paradigmas de programação servem como diferentes filosofias para estruturar a **lógica** e o **fluxo** de um programa. Cada paradigma oferece uma maneira distinta de abordar a resolução de problemas, definindo as ferramentas e técnicas disponíveis para o programador.

Programação Procedural

A programação **procedural**, também conhecida como programação imperativa, é o paradigma clássico, onde o foco está na sequência de instruções que modificam o estado do programa. O programador define um conjunto de passos a serem executados, um após o outro, manipulando variáveis e estruturas de controle para alcançar o resultado desejado.


Programação Procedural

Características:

- Ênfase na sequência de instruções e modificação do estado do programa.
- Uso de variáveis para armazenar dados e estruturas de controle para direcionar o fluxo do programa.
- Foco em instruções que descrevem "como fazer" as coisas, em vez de "o que fazer".

Programação Orientada a Objetos (POO)

A **POO** introduz o conceito de objetos como entidades autônomas que encapsulam dados (atributos) e comportamentos (métodos). Os objetos interagem entre si através de mensagens, promovendo modularidade, reuso de código e organização complexa em sistemas maiores.



P00



Características:

- Encapsulamento: Os dados e comportamentos de um objeto são agrupados e protegidos dentro do próprio objeto.
- Abstração: O programador interage com os objetos através de interfaces, ocultando detalhes internos e focando na funcionalidade principal.
- Herança: Permite que novos objetos sejam criados a partir de classes existentes, herdando atributos e métodos, promovendo reuso de código.
- Polimorfismo: Objetos de diferentes classes podem responder à mesma mensagem de maneiras distintas, proporcionando flexibilidade e adaptabilidade.

Programação Funcional

Na programação funcional, as funções assumem o papel central, sendo tratadas como unidades básicas de código que operam em dados sem modificar o estado do programa. O foco está na imutabilidade e na manipulação de dados através de funções puras e sem efeitos colaterais.

Programação Funcional

Características:

- **Imutabilidade:** Os valores dos dados não são modificados diretamente, mas sim passados como argumentos para funções que geram novos valores.
- **Funções puras:** Uma função sempre retorna o mesmo resultado para os mesmos argumentos, sem depender de estados externos ou efeitos colaterais.
- **Recursão:** Funções podem chamar a si mesmas para resolver problemas de forma elegante e concisa.

Programação Interativa

A programação interativa coloca o programador em um ciclo de feedback contínuo com o computador. O programa é executado de forma incremental, permitindo que o usuário interaja com ele e forneça feedback em tempo real. Isso é útil para prototipagem rápida, desenvolvimento de jogos e interfaces interativas.

Programação Interativa

Características:

- Interação em tempo real entre o programa e o usuário.
- Resposta dinâmica à entrada do usuário.
- Possibilidade de adaptar o programa com base nas interações.

Programação Estruturada

A programação estruturada enfatiza a organização clara e lógica do código, utilizando blocos de código bem definidos e estruturas de controle como **if-else**, **for** e **while** para controlar o fluxo do programa. Essa abordagem visa melhorar a legibilidade, a manutenabilidade e a confiabilidade do código.

Programação Estruturada

Características:

- Uso de blocos de código para organizar o programa em seções lógicas.
- Estruturas de controle bem definidas para direcionar o fluxo do programa.
- Ênfase na clareza e na legibilidade do código.

Escolhendo o Paradigma Ideal

A escolha do paradigma de programação mais adequado depende das características do problema a ser resolvido, dos objetivos do programador e das restrições do projeto.

Escolhendo o Paradigma Ideal

- **Para problemas simples e lógicos:** A programação procedural pode ser suficiente.
- **Para sistemas complexos e modulares:** A POO oferece organização e reuso de código.
- **Para manipulação de dados e transformações:** A programação funcional pode ser mais eficiente e elegante.

Escolhendo o Paradigma Ideal

- **Para interação com o usuário e interfaces dinâmicas:** A programação interativa é ideal.
- **Para programas com foco na clareza e legibilidade:** A programação estruturada é fundamental.

Conclusão

Ao dominar os diferentes paradigmas de programação, você se torna um programador mais versátil e capaz de escolher a ferramenta ideal para cada desafio. Explore diferentes linguagens de programação e pratique os conceitos aprendidos para aprimorar suas habilidades e construir software robusto e eficiente.

ATIVIDADE

PDF DAS AULAS: bit.ly/3QLKmJ1

Faça um programa em python que solicite ao usuário o seu ano de nascimento e retorne a idade da pessoa em anos.

EX: Você tem 28 anos