



Programação Funcional

Kleber Jacques F. de Souza

Programação Funcional

- É um paradigma que trata a computação como uma **avaliação de funções matemáticas** e que evita estados ou dados mutáveis.
- Ela enfatiza a aplicação de funções, em contraste da programação imperativa, que enfatiza mudanças no estado do programa

Programação Funcional

- O **programa é uma função** (ou grupo de funções), composta de funções mais simples
 - **Tudo é função!** Sem procedimentos!
 - Não há o conceito de variáveis nem comandos
- Programação Funcional **Pura**

Programação Funcional

- Uma das principais características de uma linguagem **funcional pura** é que nem expressões, nem funções, tem **efeitos colaterais**.
- Efeito colateral funcional, ocorre quando a **função** modifica um de seus **parâmetros** ou uma **variável global**.

Programação Funcional

```
int a = 5;
int funcao(){
    a = 10;
    return 3;
}
void main(){
    a = a + funcao();
}
```

Programação Funcional

- Um aspecto importante na programação funcional é que as funções são tratadas como valores de primeira classe.
- Um nome de uma função pode ser passado como parâmetro, e uma função pode retornar outra função como valor.

Vantagens

- Resultam em programas mais legíveis
- Mais confiáveis
- Imutabilidade
- Facilidade de realizar testes

Desvantagens

- Esforço inicial no desenvolvimento
- Recursividade (custo de memória)
- Dificuldade em estimar o tempo e os recursos necessários

Funcional x Imperativa

- Função que computa a soma dos cubos dos primeiros n positivos inteiros.

```
int sum_cubes(int n){  
    int sum=0;  
    for(int i=1;i<=n;i++)  
        sum += i*i*i;  
    return sum;  
}
```

```
sum_cubes n = sum(map ^3) [1...n]
```

Exemplos

- LISP
- Scheme
- ML
- Haskell

Referências Bibliográficas

SEBESTA, Robert W. **Conceitos de linguagens de programação**. Porto Alegre, Bookman, 2011.

TUCKER, Allen. **Linguagens de programação princípios e paradigmas**. Porto Alegre, AMGH, 2014.