

Type to search



Paradigmas de Programação: Uma Ab...

Paradigmas de Programação: Uma Ab...

1. 1. Introdução

1.1. 1.1 Motivação

1.2. 1.2 Paradigmas de Programação

1.3. 1.3 Objetivos

1.4. 1.4 Organização

2. 2 Paradigmas Imperativo e Declarativo

2.1.

2.2. 2.1 Paradigma Imperativo

2.3. 2.2 Paradigma Declarativo

3. 3 Paradigma Procedimental

3.1. 3.1 Paradigma Estrutural

3.2. 3.2 Paradigma Procedimental

4. 4 Paradigma Orientado a Objetos

4.1. 4.1 Classe

4.2. 4.2 Objeto

4.3. 4.3 Encapsulamento

4.4. 4.4 Herança

4.5. 4.5 Polimorfismo



Powered by GitBook

2.2 Paradigma Declarativo

A ideia do paradigma declarativo, por sua vez, é que um programa pode ser visto como um teorema em uma lógica apropriada. Por essa linha de raciocínio, a programação é levada a um nível mais alto de abstração, onde o programador pode concentrar-se em descrever o resultado o que deve ser computado, e não necessariamente como esse resultado deve ser computado. Dessa forma, a programação declarativa define "o que" é para ser computado, mais do que "como" computar, entrando assim em contraste a programação imperativa.

Por exemplo, o programa a seguir, em linguagem Prolog, que é uma linguagem declarativa, resolve o problema de obter todos os números binários de três dígitos. Note que esse programa especifica "o que" deve ser computado como resultado. O programa declara os dígitos que podem ser usados na composição de um número binário, e define um binário de três dígitos como uma tupla (A,B,C), em que cada componente representa um dígito binário. Para ver o resultado computado pelo programa, o usuário deve fazer a consulta: `?-binário(N)`.

Exemplo 2:

`dígito(0).`

`dígito(1).`

`binário(N) :- N=(A,B,C), dígito(A), dígito(B), dígito(C).`

Na programação declarativa, não há a ideia de estado do programa, como ocorre na programação imperativa. Nesta primeira há apenas declarações de verdades que, após serem declaradas, são imutáveis. Por consequência, a programação se torna sem efeitos colaterais, isto é, dadas algumas declarações e realizada alguma interação entre elas, o resultado será sempre o mesmo para aquelas declarações, enquanto que, na programação imperativa, um mesmo trecho de código nem sempre retornará o mesmo resultado (pois o resultado depende do estado em que ele é computado).