

Lista de Exercícios 3

Grupo : Guilherme Teles da Costa Moura,
José Santarem Gomes,
Letícia Souza de Araújo,
Sarah Paulina Cruz de Arruda,
Victor da Silva Dantas,
Vinícius Santana Gato.

1) Explique o que é programação lógica. Sua explicação deve dizer qual a principal característica da programação lógica. Fale sobre suas vantagens e desvantagens fazendo uma comparação com programação procedimental (incluindo os assuntos: tamanho do código fonte de um programa, velocidade de execução e facilidade de criação/manutenção do programa).

R: A programação lógica é um paradigma onde os programas são compostos por declarações ou proposições, em vez de comandos de controle e atribuições. Em vez de descrever como realizar uma tarefa passo a passo, o programador define relações, e o computador utiliza essas declarações para deduzir os resultados desejados. Uma das principais características das linguagens de programação lógica é a **semântica declarativa**, que permite que o significado de cada sentença seja determinado diretamente a partir da própria sentença, sem a necessidade de examinar o contexto de execução.

- **Tamanho do código fonte:** Como são usadas proposições e declarações, o código tende a ser mais compacto. Assim, em comparação com as linguagens imperativas, a programação lógica geralmente resulta em códigos mais curtos.
- **Velocidade de execução:** A velocidade de execução em programação lógica é mais lenta porque envolve provar que uma declaração lógica é verdadeira, utilizando processos de inferência e resolução que exploram várias possibilidades.
- **Facilidade de criação/manutenção do programa:** Os programas em programação lógica são mais fáceis de modificar em comparação com a programação procedimental, pois eles descrevem o que deve ser feito, sem se preocupar com os detalhes de como executar as tarefas.

2) Na Programação Lógica, explique o que é Cláusula, Predicado e Proposição. Explique a ligação entre eles.

R:

- **Cláusula** : É uma sentença que pode ser verdadeira ou falsa. Podendo ser uma cláusula positiva, como uma disjunção de literais afirmativos (por exemplo, $P \vee Q \vee R$ ou $Q \vee R$), ou uma cláusula negativa, que é uma implicação de uma proposição falsa (como $\neg P \vee \neg Q \vee \neg R$). Assim, pode ser vista como uma coleção de proposições lógicas que utilizam predicados.
- **Proposição** : É uma afirmação sobre o mundo, que pode ser verdadeira ou falsa. Uma expressão lógica que pode ser vista como uma coleção de proposições lógicas, como uma regra ou fato, que utilizam predicados;
- **Predicado** : É a estrutura lógica que define a relação ou a propriedade associada à proposição. Segundo Sterling e Shapiro (1994), "os predicados em Prolog são usados para descrever relações entre entidades, permitindo a definição de regras lógicas para manipulação de dados" (p. 72);

Logo, esses três conceitos utilizam a lógica de predicados para criar relações e inferir novas informações.

3) Quais são as principais características do paradigma orientado a objetos?

R: Abstração, Encapsulamento, Herança e Polimorfismo

4) O que significa encapsulamento e por que ele é importante?

R: Encapsulamento é um dos conceitos fundamentais da programação orientada a objetos (POO) e refere-se a técnica de agrupar dados (atributos) e os métodos (funções) que operam sobre esses dados em uma única unidade, ou seja, dentro de uma classe, e de controlar o acesso a esses dados, tornando-os privados ou protegidos.

É importante pois oferece vantagens como a proteção dos dados, aumentando a segurança e a confiança no sistema. Também contribui para a redução da complexidade, facilita a manutenção e modificação do código, pois como posto por Booch (2007) "o encapsulamento permite a modificação da implementação sem alterar a interface pública, o que facilita a evolução do sistema sem causar impacto nos consumidores da classe" (p. 33). Da mesma forma, promove a reusabilidade e modularidade, permitindo a criação de sistemas mais flexíveis e fáceis de adaptar.

5) Como é possível preservar o encapsulamento dos objetos? Explique os recursos de, pelo menos, duas linguagens de programação.

R: O encapsulamento é preservado ao garantir que os dados internos de um objeto sejam acessados e modificados apenas por meio de métodos específicos, limitando a visibilidade e alterando o comportamento de atributos de forma controlada. Como na linguagem Java, o encapsulamento é preservado através dos modificadores de acesso:

- **private:** Os atributos e métodos marcados como private não podem ser acessados fora da classe, preservando o encapsulamento ao impedir que outras classes modifiquem diretamente os dados internos do objeto.
- **public:** Permite que os atributos e métodos sejam acessados de qualquer parte do código.
- **protected:** Acesso restrito à classe e suas subclasses, além de classes no mesmo pacote.
- **default:** Acesso limitado ao pacote em que a classe está definida.

Em C++, o encapsulamento também é preservado usando modificadores de acesso semelhantes aos do Java:

- **private:** Atributos e métodos privados só podem ser acessados dentro da própria classe.
- **public:** Permite o acesso aos membros da classe de qualquer parte do código.
- **protected:** Permite o acesso dentro da classe e suas subclasses.

Com isso, em ambas as linguagens, é comum criar métodos públicos, chamados de **getters**, para obter o valor de um atributo, e **setters**, para definir o valor de um atributo, garantindo que as alterações sejam feitas de maneira controlada. Além disso, existem os **construtores**, usados para inicializar objetos de maneira controlada.

6) O que significa estado no conceito de orientação a objetos? Os métodos podem possuir estados?

R: No conceito de orientação a objetos (OO), o estado de um objeto refere-se aos dados internos ou atributos que ele possui em um determinado momento. Esses atributos armazenam informações que definem o comportamento e as características do objeto. O estado pode mudar com o tempo, dependendo das interações do objeto com seu ambiente, geralmente por meio de métodos.

Nesse sentido, os métodos em OO são responsáveis por modificar ou consultar o estado de um objeto. Mas não, métodos não possuem estado próprio, porém, eles podem afetar o estado de um objeto ou depender do estado atual para calcular seus resultados.

Referências:

- <https://www.devmedia.com.br/os-4-pilares-da-programacao-orientada-a-objetos/9264>
- **SEBESTA, Robert W.** *Conceitos de linguagens de programação*. Tradução técnica de Eduardo Kessler Piveta. 9. ed. São Paulo: Bookman, 2011. ISBN 9780136073475.
- Bratko, I. (2001). *Prolog Programming for Artificial Intelligence* (3rd ed.). Addison-Wesley.
- STROUSTRUP, Bjarne. *Programming Principles and Practice Using C++*. 2. ed. Boston: Addison-Wesley, 2014.
- Sterling, L., & Shapiro, E. (1994). *The Art of Prolog: Advanced Programming Techniques* (2nd ed.). MIT Press.
- Booch, G. (2007). *Object-Oriented Analysis and Design with Applications* (3rd ed.). Addison-Wesley.
- MARTIN, Robert C. *Clean Code: A Handbook of Agile Software Craftsmanship*. 1. ed. Upper Saddle River, NJ: Prentice Hall, 2009.