



Disciplina: ICC043/IEC582 – Paradigmas de Linguagens de Programação
Prof. Arcanjo Miguel Mota Lopes (amml@icomp.ufam.edu.br)

Integrantes:

- Bruna Lucas dos Santos
- David Yan dos Santos Prado
- Lucas Eduardo Souza de Moura
- Pedro Henrique Barros Mendonça
- Raquel da Mota Batista

Lista de Exercícios 3

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 se baseia na lógica formal, usa fatos e regras para dedução de soluções automaticamente. Seu foco é “o que” deve ser resolvido e não o “como”, deixando esse para a inferência. Em relação a programação procedimental, a lógica permite códigos mais compactos, porém com execução mais lenta devido à complexidade da inferência. Já na procedimental temos uma quantidade maior de codificação, porém maior controle e velocidade nos programas. Enquanto a lógica facilita a criação em problemas declarativos, o procedimental é mais simples para manutenção e ajustes práticos. Bratko, I. (2001). Prolog Programming for Artificial Intelligence. Pearson.

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

R = Uma **Cláusula** é a unidade básica de conhecimento, representando uma regra ou fato na forma de uma expressão lógica. Um **Predicado** é uma função lógica que expressa uma relação ou propriedade, associando elementos a valores verdadeiros ou falsos. Já uma **Proposição** é uma afirmação que pode ser verdadeira ou falsa, como um fato. A ligação entre eles ocorre porque as cláusulas são compostas por predicados, que por sua vez formam proposições que descrevem o conhecimento do sistema. Esse relacionamento permite a inferência lógica, onde os predicados aplicados às cláusulas geram conclusões baseadas nas proposições. Bratko, I. (2001). Prolog Programming for Artificial Intelligence. Pearson.



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

R = A **abstração**, que permite modelar elementos do mundo real em objetos com atributos e métodos; o **encapsulamento**, que organiza o acesso aos dados através de métodos, garantindo proteção e modularidade; a **herança**, que possibilita a reutilização de código em classes derivadas de outras, compartilhando atributos e métodos; e o **polimorfismo**, que permite que diferentes objetos respondam de formas específicas a uma mesma mensagem ou método. Sommerville, I. (2019). *Software Engineering* (10th ed.). Pearson.

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

R = O **encapsulamento** organiza o acesso aos dados e ações através de métodos dentro da classe. A importância é devido a segurança e modularidade que ele oferece, atributos da classe tendo acesso controlado e reutilização do código em diferentes trechos. Sommerville, I. (2019). *Software Engineering* (10th ed.). Pearson.

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 restringir o acesso direto aos atributos de um objeto, permitindo sua manipulação apenas por métodos disponibilizados pela classe. Em Python, o encapsulamento é implementado com recursos como o uso de um ou dois underscores antes do nome do atributo (`_atributo` ou `__atributo`) para indicar que ele é privado, aliado ao uso de propriedades (`@property`) para criar métodos acessores e modificadores. Em Java, isso é feito com modificadores de acesso como `private`, que limita o acesso ao atributo dentro da própria classe, e métodos `get` e `set`, usados para leitura e atualização controlada. Horstmann, C. S., & Cornell, G. (2018). *Core Java Volume I—Fundamentals*. Pearson. Lutz, M. (2013). *Learning Python*. O'Reilly Media.

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

R = O estado de um objeto refere-se aos valores atuais dos seus atributos, que definem a condição ou situação do objeto em um determinado momento. O estado de um objeto pode ser alterado ao longo do tempo, conforme os métodos são chamados. Já os métodos não possuem estado propriamente dito, pois eles são funções ou comportamentos definidos para manipular os atributos (estado) de um objeto. Sommerville, I. (2019). *Software Engineering* (10th ed.). Pearson.



7) Nesta atividade, você será responsável por desenvolver um **Sistema Especialista em Prolog** para determinar se uma pessoa está apta a doar sangue com base em critérios preestabelecidos. O objetivo é utilizar conceitos de lógica de programação e regras condicionais para modelar uma base de conhecimento que seja capaz de avaliar a compatibilidade entre doadores e receptores de sangue. Para isso considere:

Github: <https://github.com/davdprad/plp-doacao-sangue-prolog.git>

A. Axiomas

		Receptores							
		A+	A-	B+	B-	AB+	AB-	O+	O-
Doadores	A+	Sim	Não	Não	Não	Sim	Não	Não	Não
	A-	Sim	Sim	Não	Não	Sim	Sim	Não	Não
	B+	Não	Não	Sim	Não	Sim	Não	Não	Não
	B-	Não	Não	Sim	Sim	Sim	Sim	Não	Não
	AB+	Não	Não	Não	Não	Sim	Não	Não	Não
	AB-	Não	Não	Não	Não	Sim	Sim	Não	Não
	O+	Sim	Não	Sim	Não	Sim	Não	Sim	Não
	O-	Sim	Sim	Sim	Sim	Sim	Sim	Sim	Sim

B. Fatos para representar os Axiomas:

1. **compativel/2**: Crie para determinar a compatibilidade entre dois tipos sanguíneos.
2. **rhcomp/2**: crie para avaliar a compatibilidade dos fatores RH entre doador e receptor.

C. Regra Principal

1. **podedoar/2**: Crie essa regra para determinar se um indivíduo X pode doar sangue para Y, levando em conta os critérios de:
 - a. Compatibilidade do tipo sanguíneo (**compativel(X, Y)**).
 - b. Compatibilidade do fator RH (**rhcomp(X, Y)**).
 - c. Idade do doador entre 18 e 65 anos.
 - d. Peso do doador maior que 50 kg.



D. Elabore as seguintes perguntas:

1. Quem está apto a doar sangue para alguém?
2. Para quem 'Fulano' pode doar ou receber sangue?
3. Quem possui determinado tipo sanguíneo?
4. Quem é doador de fator RH + ou RH -?

E. Desenvolva uma interface gráfica para o sistema especialista de doação de sangue usando a linguagem de sua preferência (por exemplo, wxPython, JavaSwing ou Qt/C++). A interface deve permitir que os usuários:

1. Visualize os resultados das consultas disponíveis, como:
 - a. Determinar quem está apto a doar para um receptor específico;
 - b. Verificar para quem uma pessoa pode doar ou de quem pode receber o sangue;
 - c. Identificar pessoas com um determinado tipo sanguíneo ou fator RH.

Base de Conhecimento:

tiposanguineo(joao,a).	fatorrh(joao,+).	peso(joao,75.7).	idade(joao,41).
tiposanguineo(davi,a).	fatorrh(davi,+).	peso(davi,50).	idade(davi,24).
tiposanguineo(maria,a).	fatorrh(maria,-).	peso(maria,49).	idade(maria,51).
tiposanguineo(ana,a).	fatorrh(ana,-).	peso(ana,80).	idade(ana,17).
tiposanguineo(julia,o).	fatorrh(julia,+).	peso(julia,47).	idade(julia,15).
tiposanguineo(alice,a).	fatorrh(alice,+).	peso(alice,30).	idade(alice,56).
tiposanguineo(pedro,a).	fatorrh(pedro,-).	peso(pedro,20).	idade(pedro,10).
tiposanguineo(laura,b).	fatorrh(laura,+).	peso(laura,54).	idade(laura,18).
tiposanguineo(manuela,b).	fatorrh(manuela,-).	peso(manuela,61).	idade(manuela,66).
tiposanguineo(vitoria,b).	fatorrh(vitoria,+).	peso(vitoria,70).	idade(vitoria,12).
tiposanguineo(manuel,o).	fatorrh(manuel,+).	peso(manuel,130).	idade(manuel,56).
tiposanguineo(jose,ab).	fatorrh(jose,+).	peso(jose,65).	idade(jose,100).
tiposanguineo(carlos,ab).	fatorrh(carlos,-).	peso(carlos,48).	idade(carlos,67).
tiposanguineo(telma,o).	fatorrh(telma,-).	peso(telma,79).	idade(telma,48).

**** Todo código deve estar disponível no GitHub. O envio deve ser realizado via ColabWeb, inserindo link do repositório do código. No repositório deve conter o README para execução do sistema.**



Orientações Importantes:

- **Referências Acadêmicas:** Utilize artigos científicos, livros acadêmicos e outras fontes confiáveis para embasar suas respostas. As referências devem ser citadas corretamente, conforme as normas da ABNT ou APA. O uso de informações sem a devida citação será considerado plágio.
- **Ética Acadêmica:** É imperativo que o trabalho seja de sua própria autoria. Plágio, isto é, a cópia de textos de outras fontes sem a devida atribuição, é uma prática antiética e será penalizada segundo as normas institucionais.
- **Uso de Inteligência Artificial:** Caso utilize ferramentas de IA para auxiliar na escrita ou pesquisa, como ChatGPT ou similares, indique claramente onde e como essas ferramentas foram utilizadas. O uso de IA para a redação completa de respostas ou para evitar a elaboração própria será considerado inadequado e antiético.