

Aluno(a): GABARITO

1. Com relação ao software dos computadores, é correto afirmar:

- (a) A Linguagem de Máquina é universal e única, conseqüentemente é entendida por todos os computadores.
- (b) Os interpretadores das linguagens de programação geram programas FONTE para processamento pelos computadores.
- (c) Os interpretadores e compiladores das linguagens de programação geram programas FONTE para processamento pelos computadores.
- (d) Os compiladores das linguagens de programação geram programas OBJETOS para processamento pelos computadores.
- (e) Os interpretadores das linguagens de programação geram programas FONTE e OBJETOS para processamento pelos computadores.

2. Para a programação do software, a equipe de TI contratada pelo Sr. Hiroshito pretende adotar um **paradigma de programação** e uma linguagem que suporte tal paradigma. Para isso, conduziu uma pesquisa sobre os principais **paradigmas** e linguagens de programação. A pesquisa revelou diversos paradigmas, mas foram selecionados apenas o **imperativo**, o **funcional** e o **orientado a objetos**. Selecionou-se, então, uma ou mais linguagens que suportam cada paradigma.

Estão relacionados corretamente uma ou mais linguagens de programação ao respectivo paradigma de programação suportado em:

- (a) Imperativo - Cobol e Lua, Funcional - C#, Orientado a objetos - Java
- (b) Imperativo - Pascal e Cobol, Funcional - LISP, Orientado a objetos - Java e C#
- (c) Imperativo - C++, Funcional - LISP e Cobol, Orientado a objetos - Java e C
- (d) Imperativo - Java e C++, Funcional - Lua e Prolog, Orientado a objetos - Pascal e Java
- (e) Imperativo - Pascal e C++, Funcional - Lua, Orientado a objetos - Java e C#

3. Programar o computador significa "dar ordens a ele", as quais são executadas sequencialmente. Em tal **paradigma**, "representar" a solução de um problema para ser resolvido pelo computador envolve escrever uma série de ações que, se executadas sequencialmente, levam à solução. Trata-se do paradigma:

- (a) Lógico.
- (b) Funcional.
- (c) Orientado a Objetos.
- (d) Imperativo.
- (e) Orientado a Símbolos.

4. Considere a seguinte base de conhecimento em **Prolog**:

comprou(joao, honda).	comprou(joao, uno).	valor(honda, 20000).
ano(honda, 1997).	ano(uno, 1998).	valor(uno, 7000).

- (a) Crie uma regra **pode_vender** onde o primeiro argumento é a pessoa, o segundo o carro e o terceiro é o ano atual (não especificar "homem" ou "carro" nas regras), onde a pessoa só pode vender o carro se o carro for comprado por ela nos últimos 10 anos e se seu valor for menor do que 10000 Euros.

Resolução Esperada

```
%pode_vender/3
pode_vender(Pessoa, Carro, AnoAtual) :- comprou(Pessoa,Carro), ano(Carro, AnoVenda),
                                         AnoAtual-AnoVenda <= 10,
                                         valor(Carro, Valor), Valor < 10000.
```

5. Defina duas relações (fatos) em **Prolog**:

```
empr(Nome, Depto, Salario)
depto(Departamento, Gerente)
```

Escreva uma consulta ao sistema Prolog respondendo "Que empregados possuem salário superior ao de seu gerente?"

Resolução Esperada

```
% BASE DE CONHECIMENTO}
%empr/3
% empr(NomeEmpregado, Departamento, Salario)
empr(joao,financeiro,6500).
empr(charles,financeiro,10000).
empr(geraldo,financeiro, 2000).
empr(carlos,rh,3900).
empr(arilma, administrativo, 2900).

%depto/2
%depto(Departamento, NomeGerente)
depto(financeiro, joao).
depto(administrativo,rita).
depto(rh,amelia).

% CONSULTA
?- empr(Empregado, Depto, Salario), depto(Depto, Gerente), Empregado \== Gerente,
empr(Gerente, Depto, SalarioGerente), Salario > SalarioGerente.
```

6. Suponha os seguintes fatos em **Prolog**:

```
nota(joao,5.0).
nota(maria,6.0).
nota(joana,8.0).
nota(mariana,9.0).
nota(cleuza,8.5).
nota(jose,6.5).
nota(jaoquim,4.5).
nota(mara,4.0).
nota(mary,10.0).
```

Considerando que:

Nota de 7.0 a 10.0 = Aprovado.

Nota de 5.0 a 6.9 = Recuperação.

Nota de 0.0 a 4.9 = Reprovado.

Escreva uma regra para identificar a situação de um determinado aluno.

Resolução Esperada

```
%situacao/2
%situacao(NomeAluno, Situacao)
situacao(NomeAluno, reprovado) :- nota(NomeAluno, Nota), Nota >= 0, Nota <= 4.9.
situacao(NomeAluno, recuperacao) :- nota(NomeAluno, Nota), Nota >= 5, Nota <= 6.9.
situacao(NomeAluno, aprovado) :- nota(NomeAluno, Nota), Nota >= 7, Nota <= 10.
```