Questionário de PLP

1 - O que motivou o surgimento da programação funcional ?

R - Motivada pela necessidade de pesquisadores de IA e seus subcampos : computação simbólica, prova de teoremas, sistemas baseados em regras e processamento de linguagem natural.

2 - Explique quais as características que diferenciam a programação procedural, programação lógica e programação funcional.

R - Programação Procedural : [Paradigma de programação](https://pt.wikipedia.org/wiki/Paradigma_de_programa%C3%A7%C3%A3o" \o "Paradigma de programação) que especifica os passos que um programa deve seguir para alcançar um estado desejado , mas pode se referir a um paradigma de programação baseado no conceito de chamadas a procedimento. Os Procedimentos, também conhecidos como rotinas, subrotinas, métodos, ou funções simplesmente contêm um conjunto de passos computacionais a serem executados. Um dado procedimento pode ser chamado a qualquer hora durante a execução de um programa, inclusive por outros procedimentos ou por si mesmo.

R - Programação Lógica : Linguagens usadas para a programação lógica são chamadas de linguagens declarativas, porque os programas escritos nelas consistem em declarações em vez de atribuições e sentenças de fluxo de controle. Essas são na verdade sentenças, ou proposições, em lógica simbólica.

R - Programação Funcional : É um [paradigma de programação](https://pt.wikipedia.org/wiki/Paradigma_de_programa%C3%A7%C3%A3o" \o "Paradigma de programação) que trata a computação como uma avaliação de [funções matemáticas](https://pt.wikipedia.org/wiki/Fun%C3%A7%C3%A3o_matem%C3%A1tica" \o "Função matemática) e que evita estados ou dados mutáveis. Ela enfatiza a aplicação de [funções](https://pt.wikipedia.org/wiki/Sub-rotina" \o "Sub-rotina), em contraste da [programação imperativa](https://pt.wikipedia.org/wiki/Programa%C3%A7%C3%A3o_imperativa" \o "Programação imperativa), que enfatiza mudanças no estado do programa.

Uma função, neste sentido, pode ter ou não ter [parâmetros](https://pt.wikipedia.org/wiki/Par%C3%A2metro_(ci%C3%AAncia_da_computa%C3%A7%C3%A3o)" \o "Parâmetro (ciência da computação)) e um simples valor de retorno. Os parâmetros são os valores de entrada da função, e o valor de retorno é o resultado da função. A definição de uma função descreve como a função será avaliada em termos de outras funções.

3 - Quais são os três conceitos básicos para a Programação Orientada a Objetos ? Explique em que consiste cada um.

R - Classe, representa um conjunto de objetos com características afins. Uma classe define comportamento dos objetos através de seus métodos, e quais estados ele é capaz de manter através de seus atributos. Exemplo de classe: Os humanos

R - Objeto/instância, é capaz de armazenar estados através de seus atributos e reagir a mensagens enviadas a ele, assim como se relacionar e enviar mensagens a outros objetos. Exemplo de objetos da classe Humanos: José, Maria e João.

R - Atributos, são características de um objeto. Basicamente a estrutura de dados que vai representar a classe. Exemplos: Funcionário: nome, endereço. Por sua vez, os atributos possuem valores. Por exemplo, o atributo nome pode conter o valor José. O conjunto dos valores de atributos de um determinado objeto é chamado de estado.

4 - Escreva uma descrição EBNF para uma sentença condicional simples em linguagem C.

R - if(numero % 2 == 0){

Printf(“O numero informado é par”);

} else{

Printf(“O número informado não é par”);

}

5 - Quando a gramática é considerada ambígua ? Explique.

R - Uma gramática é dita ser ambígua se existe uma cadeia que pode ser gerada pela gramática em mais de um caminho, ou seja, a cadeia admite mais de uma árvore sintatica ou, equivalentemente, mais de uma derivação mais à esquerda.

6 - Diferencie análise léxica e análise sintática e dê um exemplo de erro léxico e um de erro sintático que o compilador C detecta em um programa fonte em C.

R - Um análisador léxico é essencialmente um casador de padrões, que tenta encontrar uma subcadeia de uma cadeia de caracteres que case com um padrão de caracteres.

R - Análisadores sintáticos constroem árvores de análise sintática para programas informados. Em alguns casos, a árvore de análise sintática é apenas construida implicitamente, significando que talvez apenas um percurso da árvore seja gerado.

Exemplo de erro léxico em C -

Exemplo de erro sintático em C -

7 - Defina : vinculação, tempo de vinculação, vinculação estática e vinculação dinâmica.

R - Vinculação : Vinculação é uma associação, como entre um atributo e uma entidade ou entre uma operação e um símbolo.

R - Tempo de Vinculação : O momento no qual a vinculação ocorre é chamado de tempo de vinculação. A vinculação e os tempos de vinculação são conceitos proeminentes na semântica das linguagens de programação. As vinculações podem ocorrer em tempo de projeto da linguagem, em tempo de implementação da linguagem, em tempo de compilação, em tempo de carga, em tempo de ligação ou em tempo de execução.

R - Vinculação Estática : Uma declaração explícita é uma sentença em um programa que lista nomes de variáveis e especifica que elas são de um certo tipo. Uma declaração implícita é uma forma de associar variáveis a tipos por meio de convenções padronizadas, em vez de por sentenças de declaração.

R - Vinculação Dinâmica : Com a vinculação de tipos dinâmica, o tipo de variável não é especificado por uma sentença de declaração, nem pode ser determinado pelo nome da variável. Em vez disso, a variável é vinculada a um tipo quando atribuído um valor a ela em uma sentença de atribuição. Quando a sentença de atribuição é executada, a variável que está recebendo um valor atribuído é vinculada ao tipo de valor da expressão no lado direito da atribuição.

8 - Defina tempo de vida, escopo, escopo estático e escopo dinâmico.

R - Tempo de Vida : O tempo de vida de uma variável é o durante o qual ela está vinculada a uma posição específica da memória. Então o tempo de vida de uma variável começa quando ela é vinculada a uma célula específica e termina quando ela é desvinculada dessa célula.

R - Escopo : O escopo de uma variável é a faixa de sentenças nas quais ela é visível. Uma variável é visível em uma sentença se ela pode ser referenciada nessa sentença. As regras de escopo de uma linguagem determinam como uma ocorrência em particular de um nome é associada com uma variável.

R - Escopo Estático : O escopo é chamado assim porque o escopo de uma variável pode ser determinado estaticamente - ou seja, antes da execução. Isso permite a um leitor de programas humano e um compilador determinar o tipo de cada variável.

R - Escopo Dinâmico : O escopo dinâmico é baseado na sequência de chamadas de subprogramas, não em seu relacionamento espacial uns com outros. Logo, o escopo pode ser determinado apenas em tempo de execução.

9 - Quais são as questões de projeto para o tipo ponteiro ?

R - As questões de projeto primárias particulares a ponteiros são :

* Qual é o escopo e qual é o tempo de vida de uma variável do tipo ponteiro ?
* Qual é o tempo de vida de uma variável dinâmica do monte ?
* Os ponteiros são restritos em relação ao tipo de valores aos quais eles podem apontar ?
* Os ponteiros são usados para gerenciamento de armazenamento dinâmico, endereçamento indireto ou ambos ?
* A linguagem deveria suportar tipos ponteiro, tipos de referência ou ambos ?

10 - Defina : erro de tipo e fortemente tipada.

R - Erro de tipo : Um erro de tipo é a aplicação de um operador a um operando de um tipo não apropriado. Por exemplo, na versão original do C, se um valor int fosse passado para uma função que esperava um valor float, um erro de tipo ocorreria.

R - Fortemente tipada : Uma linguagem de programação é fortemente tipada se erros de tipos são sempre detectados. Isso requer que os tipos de todos os operandos possam ser determinados, em tempo de compilação ou em tempo de execução.

11 - De que forma a verificação de tipos estática é melhor que a verificação dinâmica ?

R -

12 - Explique o que significa efeito colateral de uma função. Dê exemplo, mostrando o resultado se a avaliação for feita com e sem efeito colateral.

R -