



Lista de Exercícios 1

Conceitos de Linguagens e Critérios de Avaliação

Instruções: os exercícios podem ser resolvidos com papel e caneta ou computador. Procure treinar sua habilidade de desenvolver algoritmos sem computador. Faça testes de mesa. Não se preocupe demais com a sintaxe; deixe a tarefa de encontrar erros de sintaxe para o compilador.

- 1) O Assembly foi criado para resolver diversas dificuldades existentes na programação de computadores com código de máquina. Cite algumas.
- 2) Por que os pseudocódigos, dentre os quais podemos mencionar o Short Code, não são de fato exemplos de linguagens de programação como conhecemos hoje?
- 3) Por que a expressão "a linguagem de programação X é uma linguagem do paradigma Y " não é uma forma adequada de inserir uma linguagem de programação no contexto dos paradigmas? Qual seria a forma correta de descrever a linguagem?
- 4) O que significa distinguir linguagens entre "dinâmicas" e "estáticas"? Cite um exemplo de uma linguagem que você considera estática e uma que você considera dinâmica.
- 5) Qual a desvantagem em linguagens que possuem muitos recursos?

Considere a tabela de critérios e características para responder as próximas questões.

Características	Critérios		
	Legibilidade	Facilidade de escrita	Confiabilidade
Simplicidade	○	○	○
Ortogonalidade	○	○	○
Tipos de dados	○	○	○
Sintaxe	○	○	○
Abstração		○	○
Expressividade		○	○
Checação de tipos			○
Exceções			○
Restricted Aliasing			○

- 6) Compare duas linguagens de sua escolha em termos de legibilidade. É possível afirmar categoricamente que uma linguagem é "mais legível" que outra? Apresente argumentos para embasar sua visão.
- 7) Como a quebra da ortogonalidade pode ser prejudicial à facilidade de escrita? Dê um exemplo de duas linguagens de programação em que o mesmo fim pode ser atingido de maneira ortogonal e de maneira não ortogonal. Mostre que a ortogonalidade simplifica a escrita de programas.
- 8) Por que tratamento de exceções torna a linguagem mais confiável? Dê um exemplo de bom e um de mau tratamento de exceções em qualquer linguagem de programação.



- 9) Como a simplicidade de uma linguagem de programação pode facilitar a legibilidade? Pode-se afirmar categoricamente que linguagens mais simples são mais legíveis ou vice-versa? Apresente exemplos.
- 10) No contexto de projeto de linguagens de programação, o que são "tempos"? Estresse a diferença que existe entre o conceito de "tempo de execução" quando estamos falando sobre projetos de linguagens e sobre o tempo que um programa leva para executar.
- 11) O que é tipagem dinâmica e o que é tipagem estática?
- 12) Qual é a diferença entre "tipagem dinâmica" e "tipagem fraca"?
- 13) Como a tipagem dinâmica influencia a confiabilidade da linguagem?