

Universidade Federal de Lavras

GAC107 – Paradigmas de Linguagens de Programação

Profa. Ana Paula Piovesan Melchiori

- 1-** Explique o motivo de existirem tantas linguagens de programação, sendo que não existe uma linguagem geralmente considerada melhor que as outras.
- 2-** O tratamento de erros é algo que dificulta a escrita de programas. Quais são as vantagens de se usar manipulação de exceções?
- 3-** Qual a diferença entre uma linguagem tipificação forte e uma com tipificação fraca. Cite um exemplo de linguagem de programação de cada categoria.
- 4-** Explique o que é o recurso de sobrecarga.
- 5-** Subprogramas não são valores. Ainda assim, é útil que sejam passados como parâmetros ou armazenados em variáveis. Explique.
- 6-** Algumas linguagens têm um recurso de tipos genéricos para criação de subprogramas genéricos ou estruturas de dados genéricas. Explique as situações em que tal recurso ajuda na programação.
- 7-** Discuta a passagem de parâmetros de subprogramas em linguagens de programação, apresentando elementos de diferenciação entre as mesmas.
- 8-** Discuta sobrecarga de subprogramas em linguagens de programação, apresentando elementos de diferenciação entre as mesmas.
- 9-** Utilizando uma linguagem imperativa, escreva expressões algébricas correspondentes aos seguintes comandos:
 - a) A soma dos 5 primeiros inteiros positivos.
 - b) A idade média de Sara (idade 23), Mark (idade 19) e Fátima (idade 31).
 - c) O número de vezes que 73 cabe em 403.
 - d) O resto de quando 403 é dividido por 73.
 - e) 2 à 10ª potência.
 - f) O valor absoluto da distância entre a altura de Sara (54 polegadas) e a altura de Mark (57 polegadas).

10- Com relação ao trecho de código (Código 1), a seguir, discuta:

- a) A linguagem xyz não é fortemente tipada, ou tipificada.
- b) Na linha 2, ocorre uma conversão de tipo por alargamento.
- c) Sem conhecer a precedência dos operadores, definida na linguagem xyz, na expressão aritmética da linha 4, a melhor alternativa é o uso dos parênteses para garantir a ordem das operações esperada pelo programador.
- d) Imagine o caso em que a linguagem xyz não realiza conversão de tipos por estreitamento, então a variável c, na linha 6, receberá apenas a parte inteira do resultado da expressão aritmética da linha 4.

```
1 a = 10.5
2 b = int(a)
3 def calc_c(x, y)
4     c = x**2*y/4
5     return c
6 c = calc_c(a, b)
7 print(c)
```

Código 1 em linguagem xyz.

11- Com relação ao uso de subprogramas, discuta as seguintes afirmações:

- a) Subprogramas podem desviar o fluxo de execução de um código.
- b) Subprogramas agrupam trechos de código, tornando o programa mais legível.
- c) Subprogramas é uma técnica que causa problemas na confiabilidade do código, mas é necessária.
- d) Subprogramas geralmente, podem definir o tempo de vida e escopo de variáveis.
- e) Subprogramas podem receber parâmetros por valor ou por referência. Na passagem de parâmetro por referência, a variável referenciada poderá ter seu valor alterado pelo subprograma.

12- Cite pelo menos uma linguagem de programação de cada um dos paradigmas:

- a) Imperativo
- b) Funcional
- c) Lógico
- d) Orientado a Objetos

13- Sobre os tipos de dados, discuta as seguintes afirmações:

- a) *Ponto Flutuante* e *Decimal* aceitam, em geral, conversão por alargamento.
- b) *Decimal* é definido com um número fixo de casas decimais, *Ponto Flutuante* não.
- c) *Vetor* é um aglomerado de valores homogêneos.
- d) *String* é, geralmente, representada como um vetor de *Caracteres*.
- e) O tipo *Booleano* é suportado em todas as linguagens de programação, como um tipo primitivo.

14- A expressão $a = (b == 0) ? a-- : a++$ está na forma contraída. Reescreva a expressão na forma não contraída.

15- Considere o pseudo-código a seguir:

```
1  arquivo = abrir("c:\abc.txt", "leitura")
   linhas = arquivo.ler_linhas()
2  para_cada linha em linhas
3  faça
4      numero = int(linha)
5      print(numero)
```

Código 2.

Discuta a confiabilidade do código e o qual estrutura de controle poderia ser adicionada para tornar o código mais confiável e resistente a erros. Altere o código com a estrutura de controle.

16- Considere o trecho de código a seguir (Código 3). O que pode dar errado se a avaliação da expressão, na linha 3, não for feita em *curto-circuito*?

```
1  vetor = [a, b, c, d, e]
2  index = 0
3  while (index < len(vetor)) and (vetor[index] != f):
4      print("Não achou")
```

Código 3.

17- Utilizando uma linguagem imperativa, considere a seguinte lista de inteiros:

lista = [3,6,23,67,-98,8,90,-8,0]

Faça um programa que percorra todos os elementos da lista e exiba apenas os número maiores que a média dos números da mesma lista.

18- Por que existem vários paradigmas computacionais? Liste os principais, apresentando suas características gerais e dê exemplos de aplicações para cada paradigma.