

Linguagens de Programação - Professor Haniel Barbosa

Lista de exercícios 3

Luís Felipe Ramos Ferreira
2019022553

May 28, 2023

1. (b) O programa em si e os estados que ele mantém

- 2. a V
- b F
- c F
- d F
- e V
- f V
- g V
- h V

- 3. a
 - valor_inicial : Memória estática
 - valor_intermediario : Memória estática
 - taxa : Memória dinâmica
 - valores : Memória dinâmica
- b
 - valores[0] = 25
 - valores[1] = 75
 - valores[2] = 15

- 4.
 - *Mark and sweep*
 - *Copying collector*
 - *Reference counting*

O sistema de coleta de lixo *Reference counting* é ideal para sistemas em que o coletor de lixo deve atuar durante a execução do programa. Isso devido ao fato de que o sistema não precisa estar "parado" para que o coletor de lixo seja executado, uma vez que ele maneja a coleta de lixo em tempo de execução. Além disso, esse tipo de coleta de lixo permite que o *heap* não precise ser lido completamente para que se possa desalocar a memória, permitindo que seja feita a coleta de lixo de sistemas com grande quantidades de acesso à memória. Nesse sentido, nota-se que o algoritmo de coleta de lixo *Reference counting* é uma boa escolha para sistemas embarcados, como os utilizados em aviões e carros, onde a manutenção da memória deve ser feita de forma mais eficiente do que a de algoritmos como o *Mark and sweep*. Em contrapartida, é importante citar que o algoritmo *Reference counting* também possui suas desvantagens, como o alto custo de manutenção dos contadores de referência, além do monitoramento do uso desses contadores, de modo a evitar dependências cíclicas, ponteiros cegos, etc.

5. VPL

6. VPL

- 7. a Polimorfismo paramétrico, na definição do tipo genérico T, e polimorfismo de sobrecarga, quando se define a sobrecarga de operadores nas linhas 6 e 7.

- b Neste caso, não ocorre erro de memória uma vez que o destrutor da classe *auto_ptr* é chamado ao final da função, o que consequentemente desaloca o ponteiro alocado no contexto local. Com o uso da ferramenta *Valgrind* foi possível confirmar esse fato.
 - c Da mesma forma que na questão anterior, não irá acontecer problemas relacionados à memória uma vez que ao final da execução da função, o destrutor da classe *auto_ptr* é chamado e a memória será desalocada da maneira correta.
 - d Ao contrário dos casos anteriores, nesta função irá sim ocorrer um erro de memória, uma vez que após o lançamento da exceção, a referência para o ponteiro *p* será perdida, e consequentemente a memória não será desalocada corretamente.
8. VPL
9. VPL
10. VPL
11. a O contrato que deve ser garantido é a existência dos métodos *isEmpty()* e *remove()* nos objetos *s*, independente do seu tipo.
- b O termo *duck typing* é um termo comumente utilizado para descrever linguagens de programação tipadas dinamicamente. De modo geral, o que a expressão quer dizer é: não importa se a coisa em questão é um pato ou não, o que importa é que ela faça **quack** (onomatopeia dos sons que os patos emitem). Em outras palavras, se aquela coisa grasna, anda e come como um pato, então se trata de um pato. Essa expressão se relaciona com esse contexto uma vez que não nos importa o tipo do parâmetro *s*, desde que ele implemente os métodos definidos no contrato especificado.
12. (a) Pavao is an animal
- (b) Tigre is an mammal
- (c) Krypto is a dog
- (d) Pavao, which is an animal, is eating.
- (e) Tigre, which is a mammal, is sucking milk.
- (f) Tigre, which is an animal, is eating.
- (g) Krypto is barking rather loudly.
- (h) Krypto, which is a mammal, is sucking milk.
- (i) Krypto barks when it eats.
- (j) Runtime error
- (k) Krypto is barking rather loudly.
13. O problema do diamante, em um contexto de orientação a objetos, se refere a uma ambiguidade que ocorre quando duas classes herdam de uma mesma classe, e posteriormente uma nova classe herda dessas duas classes filhas. Dando nome aos bois, seria o caso de duas classes B e C herdarem de uma classe A, e posteriormente uma classe D herdar de B e C. Nesse cenário, ambas classes B e C reimplementam um método M de A. Qual então deve ser a implementação de M que a classe D irá utilizar, a da classe B ou da classe C?
- O nome problema do diamante vem do fato de que o formato das heranças entre as classes é similar ao de um losângulo, ou então de um diamante. Existem diversas maneiras de resolver esse problema, sendo algumas delas a reimplementação do código de modo que isso não ocorra ou a priorização de uma das classes B ou C em cima uma da outra, com base na heurística desejada.
14. VPL
15. VPL