

Lista de Exercícios 12

Parâmetros

Exercício 01) Considere o programa abaixo escrito em Java.

01: static void f(int x, int y) {	05: static void m() {
02: x = 1;	06: int A[] = { 0, 2 };
03: y = 3;	07: f(A[0], A[A[0]]);
04: }	08: }

Indique os valores do arranjo A após o término da chamada de f caso a passagem de parâmetros seja feita:

- a) Por valor
- b) Por valor-resultado
- c) Por referência
- d) Por expansão de macros
- e) Por nome

Exercício 02) Considere o programa em Java a seguir.

01: class MyInt {	12: void swap2(MyInt j) {
02:	13: MyInt tmp = j;
03: int i;	14: j.i = i;
04: MyInt(int k) { i = k; }	15: i = tmp.i;
05:	16: }
06: void swap1(MyInt j) {	17: void swap3(int j) {
07: MyInt tmp = j;	18: int tmp = j;
08: j = new MyInt(i);	19: j = i;
09: i = tmp.i;	20: i = tmp;
10: }	21: }
11:	22: }

Cada um dos itens a seguir são completamente independentes uns dos outros. Estes devem ser respondidas com base nas seguintes definições.

```
MyInt m1 = new MyInt(3);  
MyInt m2 = new MyInt(4);
```

- a) Qual o valor de m1.i e m2.i depois da chamada de m1.swap1(m2)?
- b) Qual o valor de m1.i e m2.i depois da chamada de m1.swap2(m2)?
- c) Qual o valor de m1.i e m2.i depois da chamada de m1.swap3(m2.i)?
- d) O que se pode dizer sobre a passagem de parâmetros de tipos primitivos de Java (int, float, ...)? E a passagem de objetos?

Exercício 03) A passagem de parâmetros por expansão de macros é um mecanismo bastante utilizado em C. Um exemplo é dado a seguir.

01: #include <stdio.h>
02: #define SUM(X, Y) (X) + (Y)
03: void main(int argc, char ** argv) {
04: printf("sum = %d\n", SUM(argc, argv[0][0]));
05: }

- a) Macros são expandidas por um componente do compilador chamado pré-processador. Escreva o código do programa acima, após o pré-processamento.
- b) Um dos problemas com expansão de macros é a chamada captura de variáveis. Ilustre esse problema com um exemplo diferente do visto nos slides.
- c) Um outro problema é a múltipla avaliação de parâmetros, isto é, o fato de parâmetros serem avaliados múltiplas vezes pode dar ao programa um significado diferente da intenção do programador. Escreva um exemplo em C, diferente do visto nos slides, que prove que parâmetros de macros são avaliados múltiplas vezes.

Exercício 04) Considere código seguinte com sintaxe semelhante a C++:

01: <code>#include <iostream></code>	13: <code>f(x);</code>
02: <code>using namespace std;</code>	14: <code>cout << x << ' ' << y << endl;</code>
03:	15: <code>}</code>
04: <code>int x = 3, y = 4;</code>	16: <code>int main() {</code>
05:	17: <code>int x = 1;</code>
06: <code>void f(int y) {</code>	18: <code>int y = 2;</code>
07: <code>y = y + x;</code>	19: <code>cout << x << ' ' << y << endl;</code>
08: <code>cout << x << ' ' << y << endl;</code>	20: <code>f(x);</code>
09: <code>}</code>	21: <code>g();</code>
10: <code>void g() {</code>	22: <code>cout << x << ' ' << y << endl;</code>
11: <code>int x = 5;</code>	23: <code>return 0;</code>
12: <code>cout << x << ' ' << y << endl;</code>	24: <code>}</code>

Execute esse código supondo:

- a) **Escopo estático** e passagem de parâmetro por **valor-resultado**
- b) **Escopo estático** e passagem de parâmetro **por nome**
- c) **Escopo dinâmico** e passagem de parâmetro **por valor**
- d) **Escopo dinâmico** e passagem de parâmetro **por referência**

Exercício 05) Um dos mecanismos de passagem de parâmetros chama-se passagem por nome. Na passagem por nome os parâmetros não são avaliados imediatamente. A avaliação acontece quando esses parâmetros são utilizados. Um detalhe importante, contudo, é que a avaliação acontece no contexto da chamada da função, e não no contexto da função chamada. Isto faz com que não ocorra o problema da captura de variáveis, como ocorre em expansão de macros. A ideia de passagem por nomes foi lançada em Algol e viu uso também em Simula, aquela mesma linguagem que foi a precursora das linguagens orientadas por objeto. Por outro lado, este mecanismo acabou abandonado, pois era difícil de ser implementado e complicava o código dos programas. E claro, entretanto, é que se pode fazer coisas maravilhosas com a passagem de parâmetros por nome. Considere, por exemplo, a função abaixo, escrito em Simula, e responda as perguntas a seguir.

```

01: Real Procedure Sigma (k, m, n, u);
02:   Name k, u;
03:   Integer k, m, n;
04:   Real u;
05: Begin
06:   Real s:=0;
07:   k:= m;
08:   While k <= n Do
09:     Begin
10:       s:= s + u;
11:       k:= k + 1;
12:     End;
13:   Sigma:= s;
14: End;

```

a) Qual o valor de Z na chamada a seguir?

```
Z:= Sigma (i, 1, 4, i ** 2);
```

b) O que o programa abaixo faz? Não precisa escrever o valor calculado, somente explicar o que ele calcula.

```

a := io.read_integer();
Z := Sigma (i, 1, 100, 1 / (i + a) ** 2);

```