

Universidade Federal de Minas Gerais
Ciência da Computação

Linguagens de Programação - Haniel Barbosa

Lista de Exercícios 5

Elaborada por José Wesley Magalhães

1. Considere o seguinte programa em C++:

```
1 #include <stdio.h>
2 int x=0;
3 void p(int,int);
4 void main(){
5     int x = 1;
6     p(x,x);
7 }
8 void p(int y, int z){
9     x = x+1;
10    y = y+1;
11    z = z+1;
12    printf("%d\n",x+y+z);
13 }
```

- (a) Qual o valor impresso pelo programa? Lembre-se que C++ possui escopo estático.
- (b) Suponha que nós modifiquemos a declaração de `p` tal que os parâmetros sejam passados por referência. Isto é, `p(int &y, int &z){...}`. Qual o valor impresso neste caso? Justifique.
2. A passagem de parâmetros por expansão de macros é um mecanismo bastante utilizado em C. Um exemplo é dado abaixo:

```
1 #define SUM(X, Y) (X) + (Y)
2
3 int main(int argc, char** argv) {
4     printf("sum = %d\n", SUM(argc, argv[0][0]));
5 }
```

- (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 capture de variáveis. Explique o que é esse problema.
- (c) Um outro problema é a múltipla avaliação de parâmetros. Essa múltipla avaliação faz com que o programa possa ter um significado diferente da intenção do programador. Escreva um programa em C que prove que parâmetros de macros são avaliados múltiplas vezes.

3. Um dos mecanismos de passagem de parâmetros é chamado de passagem por nome. Nesse tipo de passagem, os parâmetros não são avaliados imediatamente. Cada parâmetro real é avaliado no contexto da chamada da função, no momento em que são usados. Esta ideia foi lançada em Algol, e também teve seu uso em Simula. Apesar de ter sido abandonado por ser muito difícil de implementar, coisas interessantes podem ser feitas utilizando passagem por nome. Considere o programa abaixo, escrito em Simula, onde os parâmetros *k* e *u* são passados por nome.

```

1 Integer Procedure Sigma (k,m,n,u);
2   Name k,u;
3   Integer k,m,n, u;
4 Begin
5   Integer s;
6   k := m;
7   While k <= n Do Begin s := s + u; k := k + 1; End;
8   Sigma := s;
9 End;
```

- (a) Qual o valor de *Z* na chamada abaixo?

```

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

- (b) Explique o que o programa abaixo faz. Não precisa escrever o valor calculado, apenas dizer o que ele calcula.

```

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

4. Acerca de passagem de parâmetros em C/C++, responda:

- (a) É preferível que alguns parâmetros sempre sejam passados por referência, se possível, dependendo do seu tipo. Cite um exemplo de tipo de variável em C/C++ em que a passagem por referência seja a melhor escolha e explique por que esta é a melhor opção.
- (b) Em qual região de memória são armazenados os parâmetros passados por valor?

5. Consider a classe abaixo, implementada em Java:

```

class MyInt {
    int i;
    MyInt(int k) {
        i = k;
    }
}
```

```
void swap1(MyInt j) {  
    MyInt tmp = j;  
    j = new MyInt(i);  
    i = tmp.i;  
}  
void swap2(MyInt j) {  
    MyInt tmp = j;  
    j.i = i;  
    i = tmp.i;  
}  
void swap3(int j) {  
    int tmp = j;  
    j = i;  
    i = tmp;  
}  
}
```

Cada uma das próximas questões é completamente independente uma das outras. Estas questões devem ser respondidas com base nas definições abaixo:

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

- (a) Qual é o valor de `m1.i` e `m2.i` depois da chamada `m1.swap1(m2)`?
- (b) Qual é o valor de `m1.i` e `m2.i` depois da chamada `m1.swap2(m2)`?
- (c) Qual é o valor de `m1.i` e `m2.i` depois da chamada `m1.swap3(m2.i)`?
- (d) Qual é o tipo de passagem de parâmetros que Java adota para tipos primitivos (*int*, *float*, *char*, etc)?
- (e) Qual é o tipo de passagem de parâmetros que Java adota para objetos?