

Lista de Linguagens de Programação – 11

Nome: _____ Matrícula: _____

1. Considere o programa abaixo, escrito em SML/NJ:

```
fun g x =  
  let val inc = 1  
    fun f y = y + inc  
    fun h z =  
      let val inc = 2  
        in  
          f z  
        end  
      in  
        h x  
      end  
    end  
  end
```

- (a) Desenhe um círculo em torno de cada bloco deste programa, e numere estes blocos.
- (b) Quais são os nomes definidos neste programa?
- (c) Para cada definição, descreva o escopo desta definição fornecendo seu número de bloco.
- (d) Para cada ocorrência de um nome, além da definição daquele nome, mostre a qual definição este nome está associado, desenhando uma seta entre o nome e a definição.
- (e) Com base no resultado da questão anterior, qual o valor de `g 5`?

2. Considere a classe `Reuse` abaixo:

```
1  class Reuse {
2      Reuse Reuse (Reuse Reuse) {
3          Reuse:
4          for (;;) {
5              if (Reuse.Reuse(reuse) == Reuse)
6                  break Reuse;
7          }
8          return Reuse;
9      }
10 }
```

(a) A definição de `Reuse` na linha 3 é um rótulo. Caso o comando da linha 6 seja executado, então o fluxo do programa será desviado para a linha 8. Qual o significado de cada uma dentre as outras definições do nome `Reuse`?

(b) A ocorrência de `Reuse` na linha 6 é definida na linha 3. Para cada outra ocorrência da palavra `Reuse`, *que não seja uma definição*, mostre onde este nome é definido.

3. As funções deste exercício devem ser escritas sem que sejam utilizadas as funções `foldr`, `foldl` e `map`.

(a) Defina a função `mymap`, que tenha o mesmo tipo e comportamento de `map`.

(b) Qual o tipo de `mymap`?

(c) Defina a função `myfoldr`, que tenha o mesmo tipo e comportamento de `foldr`.

(d) Qual o tipo de `myfoldr`?

(e) Defina a função `myfoldl`, que tenha o mesmo tipo e comportamento de `foldl`.

4. Considere o programa abaixo, escrito em JavaScript:

```
var x=1;
if (true) {
  var x=2;
  if (true) {
    var x=3;
    alert(x);
  }
  alert(x);
}
alert(x);
```

Você pode executar esse programa em algum interpretador JavaScript disponível on-line, por exemplo, <http://mochi.github.com/mochikit/examples/interpreter/>. Se o fizer, você verá que o número 3 será impresso 3 vezes! De acordo com essa saída, como é o escopo de variáveis em JavaScript?

5. Considere o programa abaixo, escrito em bash script:

```
#!/bin/bash
x=1
function g () { echo $x ; x=2 ; }
function f () { local x=3 ; g ; }
f # serah impresso 1 ou 3?
echo $x # serah impresso 1 ou 2?
```

Executar esse programa é muito fácil: basta salvá-lo em um arquivo, por exemplo, `scope.bash`, adicionar ao arquivo a permissão para execução, e invocá-lo em linha de comando, conforme feito logo abaixo:

```
~/fernando$ chmod u+rx scope.bash
~/fernando$ ./scope.bash
```

Com base no que for impresso, responda: bash script possui escopo dinâmico ou estático?

6. *Scheme* é uma das linguagens mais populares entre os afeccionados por linguagens de programação (não que isto queira dizer muito). O objetivo deste exercício é conhecer um pouco mais sobre esta linguagem.
- (a) O “pai” de *Scheme*, Guy Steele, foi também um dos idealizadores de uma das mais populares linguagens de programação atualmente em uso. Que linguagem foi esta?
 - (b) *Scheme* foi criada a partir de LISP, mas ao contrário desta linguagem, *Scheme* possuía escopo léxico. O que é isto?
 - (c) Escreva um programa “Hello, World!” em *Scheme*.
 - (d) Investigue o significado dos seguintes construtos de *Scheme*: `let`, `let*` e `letrec`.
 - (e) *Scheme* é famosa pelas *macros limpas* (tradução livre de *hygienic macro*). C é uma linguagem que também possui macros, mas estas macros não são *limpas*. Por que as macros de C não são limpas?