

## Lista de Linguagens de Programação – 8

Nome: \_\_\_\_\_ Matrícula: \_\_\_\_\_

1. Considere uma linguagem desconhecida, com tipos primitivos inteiro e *string* tal que as seguintes igualdades sejam válidas:
  - (a)  $1 + 2 * 3 = 7$
  - (b)  $"1" + "2" + "3" = "123"$
  - (c)  $"1" + 2 + 3 = "123"$
  - (d)  $1 + "2*3"$  é um erro de tipo.

Descreva um sistema de precedência, associatividade, sobrecarga e coerção implícita que suporte esta linguagem. Neste sistema, qual o resultado da avaliação de  $"1" + 2*3$ ?

2. Considere três tipos de dados:

- X: inteiros divisíveis por 3.
- Y: inteiros divisíveis por 12.
- Z: inteiros.

Considere que as variáveis  $x$ ,  $y$  e  $z$  possuem os tipos X, Y e Z, respectivamente. Para cada atribuição abaixo, responda se o compilador pode informar se a atribuição é segura:

(a)  $x \leftarrow y$

(b)  $x \leftarrow x$

(c)  $y \leftarrow y + 1$

(d)  $z \leftarrow x$

(e)  $x \leftarrow z$

(f)  $x \leftarrow x + 3$

(g)  $x \leftarrow x + z$

3. Esta questão refere-se ao programa abaixo, escrito na linguagem C:

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 typedef struct ThreeDPointType { int x; int y; int z; } ThreeDPoint;
4 typedef struct DateType { int day; int month; int year; } Date;
5 int main() {
6     ThreeDPoint *p = (ThreeDPoint*)malloc(sizeof(ThreeDPoint));
7     Date *d;
8     char a[] = {7, 0, 0, 0, 7, 0, 0, 0, 30, 7, 0, 0};
9     p->x = 7;
10    p->y = 7;
11    p->z = 1822;
12    d = (struct ThreeDPoint*)p;
13    printf("%d, %d, %d\n", d->day, d->month, d->year);
14    d = (struct ThreeDPoint*)a;
15    printf("%d, %d, %d\n", d->day, d->month, d->year);
16 }
```

- (a) Implemente o programa acima, e reporte o que será impresso.
- (b) Esse programa evidencia que C é uma linguagem fracamente tipada. Por que?
- (c) O resultado impresso depende do compilador utilizado para compilar o programa? Justifique a sua resposta.
- (d) Esse tipo de construção sem dúvida torna os programas escritos em C mais difíceis de serem entendidos. Ainda assim essas coerções inseguras são úteis. Descreva uma situação em que você poderia usar esse tipo de padrão de programação.

4. Essa questão é similar à questão anterior, mas dessa vez estaremos usando um programa escrito em Java:

```
class ThreeDPoint { int x; int y; int z; }
class DateType { int day; int month; int year; }
public class Coersion {
    public static void main(String args[]) {
        ThreeDPoint p = new ThreeDPoint();
        p.x = 7; p.y = 7; p.z = 1822;
        Object o = p;
        DateType d = (DateType)o;
        System.out.println(d.day + ", " + d.month + ", " + d.year);
    }
}
```

- (a) Escreva esse programa em um arquivo chamado `Coersion.java`, compile-o usando `javac`, e execute-o usando o interpretador `java`. O que será impresso?
- (b) A sua resposta para a questão anterior ilustra o fato de que Java é uma linguagem fortemente tipada. Essa tipagem forte tem um custo em termos de desempenho. Que custo é esse?
- (c) Considere agora esse mesmo programa, dessa vez escrito em C++:

```
class ThreeDPoint { public: int x; int y; int z; };
class DateType { public: int day; int month; int year; };
main() {
    ThreeDPoint *p = new ThreeDPoint();
    p->x = 7; p->y = 7; p->z = 1822;
    DateType *d = (DateType*)p;
    std::cout << d->day << ", " << d->month << ", " << d->year << std::endl;
}
```

Compile esse programa usando `g++` e responda, com base na execução do mesmo: C++ é uma linguagem fortemente tipada, ou fracamente tipada?

5. Considere a seguinte seção de SML:

```
~/fernando$ sml
Standard ML of New Jersey v110.68 [built: Tue Aug 18 23:43:29 2009]
- type ThreeDPoint = int * int * int;
type ThreeDPoint = int * int * int
- type Date = int * int * int;
type Date = int * int * int
- fun days (d:Date) = #1 d;
val days = fn : Date -> int
- val point:ThreeDPoint = (7, 7, 1822);
val point = (7,7,1822) : ThreeDPoint
- days point;
#####

```

- (a) O que será impresso no lugar de #####?
- (b) A sua resposta para a questão anterior mostra que o sistema de tipos de SML é nominal ou estrutural? Justifique a sua resposta.
- (c) Considere, agora, um programa semelhante, escrito em C:

```
#include <stdio.h>
typedef struct ThreeDPointType { int x; int y; int z; } ThreeDPoint;
typedef struct DateType { int day; int month; int year; } Date;
int days(Date d) { return d.day; }
int main() {
    ThreeDPoint p;
    p.x = 7; p.y = 7; p.z = 1822;
    printf("%d\n", days(p));
}
```

Implemente e compile esse programa. Qual o resultado encontrado?

- (d) A sua resposta para a questão anterior demonstra que a linguagem C possui um sistema de tipos estrutural ou nominal?

6. O objetivo deste exercício é resolver o problema 18 do projeto *Euler*. Este problema consiste em encontrar a maior soma em um percurso descendente em um triângulo de números. Por exemplo, dado o triângulo abaixo:

```
3  
7 4  
2 4 6  
8 5 9 3
```

O maior percurso é 23: começamos pelo 3 (primeira linha), somamos 7 (segunda linha), somamos 4 (terceira linha) e então somamos 9 (quarta linha). Em um percurso descendente, podemos mover-nos somente para um número na linha imediatamente abaixo, que esteja imediatamente à esquerda ou imediatamente à direita da posição corrente.

Bom, sem mais delongas, escreva um programa em SML que encontre o percurso de maior soma no triângulo abaixo:

```
75  
95 64  
17 47 82  
18 35 87 10  
20 04 82 47 65  
19 01 23 75 03 34  
88 02 77 73 07 63 67  
99 65 04 28 06 16 70 92  
41 41 26 56 83 40 80 70 33  
41 48 72 33 47 32 37 16 94 29  
53 71 44 65 25 43 91 52 97 51 14  
70 11 33 28 77 73 17 78 39 68 17 57  
91 71 52 38 17 14 91 43 58 50 27 29 48  
63 66 04 68 89 53 67 30 73 16 69 87 40 31  
04 62 98 27 23 09 70 98 73 93 38 53 60 04 23
```

Como existem “somente” 16.384 percursos possíveis, pode-se até tentar resolver este problema por força-bruta. Claro que nós não vamos fazer isto, afinal, nossa universidade tem um nome a zelar. Uma boa dica seria usar um pouquinho de *programação dinâmica* neste problema. Outra coisa: não se preocupe com entrada e saída:

“Os lírios do campo se vestem garbosamente, e eles não não se preocupam com entrada e saída, no entanto nem Salomão em toda a sua glória se vestiu como eles”.

Simplesmente inicialize sua estrutura de dados com os números de nosso triângulo. Se não quiser digitar os números, um por um, você pode copiá-los no site do projeto *Euler*.