Paradigmas de Linguagens de Programação
Exame Escrito
Centro de Informática – UFPE, 7 de julho de 2015

Questão 1 [2,0] Defina as funções ***front*** e ***last*** sobre listas. A primeira retorna todos os elementos de uma lista, exceto o último; a segunda retorna o último elemento. Por exemplo, ***front [1,2,3]*** = ***[1,2]*** e ***last [1,2,3]*** = ***3.*** Ambas as funções estão definidas apenas para listas não vazias (listas com pelo menos um elemento).

Questão 2 [2,0] Prove, por indução, para listas não vazias ***xs***, a seguinte propriedade:

xs = front xs ++ [last xs]

onde **++** é o operador de concatenação de listas, como visto em sala. **[Dicas: o caso base deve considerar uma lista de tamanho 1 e, na prova do passo indutivo, será necessário usar uma equação da definição de concatenação de listas]**

Questão 3 [2,0]

a) Considere duas variáveis (***x*** e ***y***) cujos valores são referências a objetos de um tipo ***Conta*** em uma linguagem orientada a objetos como Java. Considere ainda que a classe ***Conta*** tem um atributo público ***balance***. É sempre possível simplificar a sequência de atribuições:

x.balance = y.balance + 100; x.balance = y.balance + x.balance

para uma única atribuição: ***x.balance = y.balance + (y.balance + 100)***? Caso considere a simplificação sempre válida, justifique a sua resposta. Caso contrário, fornece um contraexemplo.

b) Explique o conceito de *aliasing* em programação e a relação entre este conceito e passagem de parâmetro por referência.

Questão 4 [1,0] Assinale com V (para verdadeiro) ou F (para falso):

( ) A introdução de passagem de parâmetro por resultado, em uma linguagem funcional, viola o princípio da transparência referencial.

( ) Enquanto um programa (sequencial) em uma linguagem imperativa ou funcional é sempre determinístico, um programa em uma linguagem lógica pode ser não-determinístico, dado que o modelo computacional do paradigma lógico é uma relação e não uma função.

Questão 4 [3,0] Estenda a Linguagem Imperativa 2 (veja BNF em anexo) permitindo a passagem de parâmetros por valor resultado. A implementação deve considerar os métodos de avaliação e checa tipo, bem como as classes novas ou modificadas, impactadas por esta mudança. Particularmente:

1) Defina a BNF para a linguagem redefinida, destacando apenas o que mudar.
2) Explique se é necessária alguma mudança nos ambientes de compilação e execução.
3) Implemente novas classes, se for o caso, e indique todas as classes que seriam afetadas (ilustre a modificação em pelo menos uma classe impactada).

Boa Sorte

Apêndice 1. BNF de LI2.

Programa ::= Comando

Comando ::= Atribuicao | ComandoDeclaracao
| While | IfThenElse
| IO | Comando ";" Comando
| Skip | ChamadaProcedimento
Atribuicao ::= Id ":=" Expressao
Expressao ::= Valor | ExpUnaria | ExpBinaria | Id
Valor ::= ValorConcreto
ValorConcreto ::= ValorInteiro | ValorBooleano | ValorString
ExpUnaria ::= "-" Expressao | "not" Expressao | "length" Expressao
ExpBinaria ::= Expressao "+" Expressao
| Expressao "-" Expressao
| Expressao "and" Expressao
| Expressao "or" Expressao
| Expressao "==" Expressao
| Expressao "++" Expressao

ComandoDeclaracao :: = "{" Declaracao ";" Comando "}"

Declaracao ::= DeclaracaoVariavel | DeclaracaoProcedimento
| DeclaracaoComposta
DeclaracaoVariavel ::= "var" Id "=" Expressao
DeclaracaoComposta ::= Declaracao "," Declaracao
DeclaracaoProcedimento ::= "proc" Id "(" [ListaDeclaracaoParametro] ")" "{" Comando "}"
ListaDeclaracaoParametro ::= Tipo Id | Tipo Id "," ListaDeclaracaoParametro
Tipo ::= "string" | "int" | "boolean"
While ::= "while" Expressao "do" Comando
IfThenElse ::= "if" Expressao "then" Comando "else" Comando
IO ::= "write" "(" Expressao ")" | "read" "(" Id ")"
ChamadaProcedimento ::= "call" Id "(" [ListaExpressao] ")"
ListaExpressao ::= Expressao | Expressao, ListaExpressao

Classes Auxiliares

AmbienteCompilacaoImperativa2

AmbienteExecucaoImperativa2
ContextoCompilacaoImperativa2
ContextoExecucaoImperativa2
ListaValor
Procedimento
DefProcedimento
ProcedimentoJaDeclaradoException
ProcedimentoNaoDeclaradoException