Universidade Federal de Roraima
Departamento de Ciência da Computação
Linguagens de Programação

DISCIPLINA: Linguagens de Programação – DCC206

2ª Lista
Prazo de entrega: 05/08/2016

ALUNO(A): Bruno Rodrigues Caputo **NOTA:_____**

ATENÇÃO: Descrever as soluções com o máximo de detalhes possível, inclusive a forma como os testes foram feitos. Para as questões que requisitarem a escrita/implementação de programas, deve ser enviado código fonte do programa. Na resposta para a questão deve ser apresentado: o modo de compilar/executar o programa; a linha de comando para executar o programa; e um exemplo de entrada/saída do programa.

[Questão 01] Descreva o que é programação funcional, bem como, apresente suas vantagens e desvantagens.

R:Programação funcional é um paradigma(Modelo recomendado a ser seguido por uma metodologia) onde os programas são executados por meio de avaliação de funções,na maioria das vezes usados para verificar a aplicação de argumentos a funções matemáticas, sem haver mudança de estado. Dessa forma, não existem sentenças de atribuição e nem variáveis no sentido imperativo, e como iterações são regidas pela mudança e atribuição das variáveis elas não são postas nesse paradigma.
Vantagens:Paralelização e concorrência de código simplificado,Composição de funções é feita mais intuitivamente, como uma função matemática.Fácil de extrair modelos, fazendo com que códigos funcionais tenham mais garantias durante o processo de compilação.Garantias de correção por provas matemáticas.
Desvantagens: Difícil prever o tempo e o espaço necessário para a execução,pode ser mais complexo o começo do aprendizado que as linguagens imperativas.

[Questão 02] Apresente as principais características das linguagens funcionais. Adicionalmente, para cada característica apresente um exemplo de código em haskell.
R: **-**Função puras,a função retornará o mesmo resultado,caso tenha sido chamada com os mesmos argumentos.
EX: exemplo1.hs A função “divide” sempre retornará o mesmo valor

-Funções de ordem maior,funções podem retornar funções e receber funções como argumentos
EX:exemplo2.hs A função “resultado” retorna a função soma

-Estados imutáveis, Uma vez que o valor de uma variável for definido ele não pode ser modificado,assim, o programa diminui a chance de ter efeitos colaterais.

-Composição de funções: A partir de duas funções é possível gerar uma nova função que execute as duas de forma simultânea
EX:exemplo2.hs A função resultado é um composição das funções “som3” e “soma”

[Questão 03] Pesquise e apresente o nome de 03 empresas que utilizam linguagens funcionais, bem como, o nome da linguagem e em qual domínio a linguagem de programação é utilizada.

R: -ABN AMRO é um banco internacional estabelecido em Amsterdã,para as atividades de investimento do banco é preciso analisar a parte de risco nos portfolios de finanças derivativas,utiliza haskell.

-Facebook usa um pouco de haskell internamente para as ferramentas,lex-pass é uma ferramenta que manipula o código base de php via haskell

-Haskell é usado e alguns projetos internos do Google, para infraestrutura de suporte interna e o open-source projeto Ganeti(Ferramenta para gerenciamento de grupos em redes virtuais.)

Bibliografia:https://wiki.haskell.org/Haskell_in_industry

[Questão 04] Defina Orientação a Objetos (OO) e apresente benefícios.

R: Orientação a Objetos é um paradigma que foi criado para tentar aproximar o mundo real do mundo virtual,a ideia principal tenta simular os problemas do mundo real no computador,utilizando-se de objetos,tais quais existentes no mundo real.Neste paradigma tem-se que o programador é responsável por moldar o universo dos objetos, e explicar como tais objetos apresentam características que podem ser agrupadas e de que modo possam interagir com outros objetos.Tem-se firmamente ligada a ideia dos blocos e escopos de cada objeto.Um carro por exemplo poder ser uma classe que agrupa informações como quantidade de rodas,cor,ano de fabricação e número de portas.Utilizando este classe podem-se criar quantos objetos se queiram(desde que haja memória suficiente).Sendo cada objeto criado uma entidade diferente da outra,mesmo que os objetos compartilhem características semelhantes ainda assim serão objetos distintos,tal como acontece na vida real.dentre os benefícios deste paradigma temos o encapsulamento que junta códigos e as datas que estão sendo manipuladas, o que garante maior segurança por interferência por algo mais externo e uso incorreto, visto que como cada objeto deve estar encapsulado e agrupado têm-se de ser defininido como será o acesso a estes objetos, como public ou private.além do polimorfismo que atua para permitir que uma interface possa ser usada para acessar uma classe mais geral de ações.E por fim herança, que nada mais é, o processo ao qual um objeto pode adquirir propriedades de outros objetos.

[Questão 05] No programa em java abaixo, apresente e determine o escopo e o tipo (primitivo ou objeto) de cada variável/atributo.

```
1.public class Car 2.{

3.      int year;/*primitivo,atributo,pertence a toda classe Car 2.*/
4.      String make;/*primitivo,atributo,pertence a toda classe Car 2.*/
5.      double speed;/*primitivo,atributo,pertence a toda classe Car 2.*/

6.
7.      public Car(int y, String m, double beginningSpeed)
8.      {
9.          year = y;

10.     }
11.     /* int y,primitivo,parâmetro,pertence somente ao método Car,que está dentro da classe Car 2.*/
12.     /* String m,primitivo,parâmetro,,pertence somente ao método Car,que está dentro da classe Car 2.*/
13.     /* double beginningSpeed,primitivo,parâmetro,pertence somente ao método Car,que está dentro da
        classe Car 2.*/
11.
12.     public int getYear()

13.     {
14.         int tmp = year;
15.         Roda r = new Roda(tmp);
16.         return year;

15.     }
17.}
```

/***int tmp**, *primitivo,,local,só existe dentro do método getYear*/
/***Roda r = new Roda(tmp)**, *Objeto da classe Roda,inicializado com temp,que fica dentro do método getYear, que por sua vez está na classe Car 2.*/

Universidade Federal de Roraima
Departamento de Ciência da Computação
Linguagens de Programação

[Questão 06] No programa escrito na linguagem de programação C abaixo, determine se no programa ocorre algum erro (bug) relacionado à referência de memória ou gerenciamento de memória. Caso a resposta seja sim, justifique a sua resposta.

```
1.  #include <stdlib.h>
2.  #define BLOCK_SIZE 128
3.
4.  int *a, *b;
5.  int n;

6.
7.  void foo ()
8.  {
9.      int i;
10.     for (i = 0; i < n; i++)
          a[i] = -1;
11.     for (i = 0; i < BLOCK_SIZE - 1; i++)
12.         b[i] = -1;

14.}
15.
16.  int main ()
17.  {
18.      n = BLOCK_SIZE;
19.      a = malloc (n * sizeof(*a));
20.      b = malloc (n * sizeof(*b));

21.      *b++ = 0;
22.      foo ();
23.      if (b[-1])
24.      {

25.          free(a); free(b);
26.      }
27.      else

28.      {
29.          free(a); free(b);
30.      }
31.

32.     return 0;
33.  }
```

R: No código acima a verificação do if(b[-1]) (linha 23) pode ser substituída por free(a);free(b); uma vez que, independentemente se a verificação é correta ou não ,será liberado espaço da memória referente ao apontador a e apontador b.

Considerando a linha 21 *b++=0 temos que o valor 0 será atribuido ao valor inicial da primeira memória de b, e em seguida será adicionado um endereço de memória,fazendo com que se perca o endereço original de b ocasionando um vazamento de memória.

Universidade Federal de Roraima
Departamento de Ciência da Computação
Linguagens de Programação

[Questão 07] Implemente os seguintes algoritmos e programas nas linguagens de programação Java e Haskell.

(A)

```
programa Cap04_Ex3d_Pg100
var
  N1, N2, N3, N4, MD, NOVA_MD, EX : real
início
  leia N1, N2, N3, N4
  MD ← (N1 + N2 + N3 + N4) / 4
  se (MD >= 7) então
    escreva "Aluno Aprovado", MD
  senão
    leia EX
    NOVA_MD ← (MD + EX) / 2
    se (NOVA_MD >= 5) então
      escreva "Aluno Aprovado em Exame", NOVA_MD
    senão
      escreva "Aluno Reprovado", NOVA_MD
    fim_se
  fim_se
fim
```

Comando para compilar : javac Media.java
Comando para executar : java Media

(B)

```
programa Cap04_Ex3j_Pg100
var
  N, RESTO : inteiro
início
  leia N
  RESTO ← N - 2 * (N div 2)
  se (RESTO = 0) então
    escreva "O valor ", N, " é PAR."
  senão
    escreva "O valor ", N, " é IMPAR."
  fim_se
fim
```

Comando para compilar : javac Par.java
Comando para executar : java Par

(C)

para i ← 1 **até** tamanhode(A)-1

    j ← i

    enquanto j > 0 **e** A[j-1] > A[j] **troca** A[j] **e** A[j-1]

        j ← j - 1

    fim enquanto

fim para
Comando para compilar : javac C.java
Comando para executar : java C

(D)

```
void minmax(int *vec, int n, int *min, int *max) { int i;
  int *min = vec[0]; int *max = vec[0];
                                        for(i = 1; i < n; i++) { if(vec[i] < *min) { *min = vec[i];
        }
        if(vec[i] > *max) { *max = vec[i];
        }
  }
}
```