

## Primeira Questão

No paradigma funcional, um programa é um conjunto de definições de funções que aplicamos a valores. Baseia-se no conceito matemático de função, em que para cada elemento do seu conjunto domínio (entrada) há apenas um elemento no seu conjunto contradomínio (saída). Além disso, as funções são normalmente expressas por meio de outras funções - de modo que obter o valor da função para um determinado conjunto de parâmetros envolve não só aplicar as regras daquela função, mas também fazer uso de outras funções. Modela-se o programa usando álgebra e funções.

Vantagens:

- Paralelização e concorrência de código são mais simples.
- Composição de funções é uma forma simples de otimização.
- Fácil de extrair modelos, fazendo com que códigos funcionais tenham mais garantias durante o processo de compilação.

Desvantagens:

- Difícil de prever performance e requisitos.
- Efeitos colaterais são úteis.
- Menos performática.

## Segunda Questão

Possui quatro características:

Funções puras: Se uma função for chamada com os mesmos argumentos, ela sempre irá retornar o mesmo valor. Ex.:

soma a b = a + b

Funções de ordem maior: Funções podem retornar funções e receber funções como argumentos. Ex.:

```
somatorio l = soma l 0
where
soma [] = 0
soma (x : xs) = x + (soma xs)
tamanho [] = 0
tamanho (x : xs) = 1 + (tamanho xs)
media l = (somatorio l) / (length l)
```

Estados imutáveis: Uma vez que o valor de uma variável for definido ele não pode ser modificado. O programa não deve ter efeitos colaterais. Ex.:

```
x = soma 2 5
```

```
x = 5
```

Esse código apresentará erro de compilação, alegando que há múltiplas declarações para X. "Multiple declarations of `x`".

Composição de funções: A partir de duas funções é possível gerar uma nova função que execute as duas de forma simultânea. Ex.:

```
verifica_par n
| n `mod` 2 == 0 = True
| otherwise = False
mult2 = ( * ) 2

verifica_lista l = filter(verifica_par) l
mult2_lista l = map (mult2) l
faz_tudo = verifica_lista . mult2_lista
```

### **Terceira Questão**

O Facebook utiliza programação funcional em várias áreas, a mais famosa é a ferramenta Sigma. Sigma é uma ferramenta de segurança do Facebook feita em Haskell que combate spam, malware e outros tipos de mensagens nocivas, os deletando automaticamente caso sejam detectados.

O Twitter utiliza amplamente a linguagem Scala, e o Github do Twitter possui um arquivo de texto relativamente extenso descrevendo detalhadamente o seu uso da linguagem. Ou seja, Scala é uma das principais linguagens de programação e uma das mais utilizadas no Twitter.

A Netflix também utiliza programação funcional. O projeto Atlas, que consiste em um sistema de monitoramento em tempo real, é utilizado pela Netflix e seu repositório no Github é constantemente atualizado.

## **Quarta Questão**

Programação orientada a objetos é um paradigma de programação em que os programadores definem não só o tipo de dados de uma estrutura de dados, mas também os tipos de operações (funções) que podem ser aplicadas à estrutura de dados. Desta forma, a estrutura de dados torna-se um objeto que inclui ambos os dados e as funções. Além disso, os programadores podem criar relações entre um objeto e outro. Por exemplo, os objetos podem herdar características de outros objetos.

Uma das principais vantagens de técnicas de programação orientada a objeto sobre técnicas de programação processuais é que eles permitem aos programadores criar módulos que não precisam ser alterados quando um novo tipo de objeto é adicionado. Um programador pode simplesmente criar um novo objeto que herda muitas características de objetos já existentes. Isso faz com que os programas orientados a objetos sejam mais fáceis de modificar. Outros benefícios de programação orientada a objetos:

- Melhoria da interação entre analistas e especialistas;
- Aumento da consistência interna dos resultados da análise;
- Uso de uma representação básica consistente para a análise e projeto;
- Alterabilidade e extensibilidade;
- Legibilidade;
- Reutilização de códigos.

### Quinta Questão

```
1. public class Car
2. {
3.     int year;           →      Atributos – visíveis em toda a classe
4.     String make;       →      Atributos – visíveis em toda a classe
5.     double speed;      →      Atributos – visíveis em toda a classe
6.
7.     public Car(int y, String m, double beginningSpeed)
8.     {
9.         year = y;       →      Parâmetros – escopo local
10.    }
11.
12.    public int getYear()
13.    {
14.        int tmp = year;  →      Variável local (primitivo)
15.        Roda r = new Roda(tmp); → Variável local (objeto)
15.        return year;
16.    }
17. }
```

### **Sexta Questão**

Executando a linha 21 do código o valor de b, que é um endereço de memória, aumenta. Fazendo isso, o valor de b não é alocado dinamicamente, como no começo. Isso torna as linhas 25 e 29, onde encontra-se a expressão 'free(b)', inúteis. Por fim, isso gera um erro de endereço de memória inválido, o Runtime Error.

## Sétima Questão

Nas questões abaixo, para compilação e teste, foi utilizado o site [ideone.com](http://ideone.com).

Códigos-fonte:

(A)Haskell

```
import System.IO
```

```
main = do
```

```
    inputjar <- getLine
```

```
    let n1 = read inputjar :: Double
```

```
    inputjar <- getLine
```

```
    let n2 = read inputjar :: Double
```

```
    inputjar <- getLine
```

```
    let n3 = read inputjar :: Double
```

```
    inputjar <- getLine
```

```
    let n4 = read inputjar :: Double
```

```
    let soma = n1 + n2 + n3 + n4
```

```
    let media = soma/4
```

```
    if media >= 7.0
```

```
        then putStrLn "APROVADO"
```

```
        else do
```

```
            inputjar <- getLine
```

```
            let ex = read inputjar :: Double
```

```
            let novamedia = (media+ex)/2
```

```

if novamedia >=5
then putStrLn "APROVADO"
else putStrLn "REPROVADO"

```

(A)Java

```

import java.util.Scanner;

public class Questao7A {

    public static void main(String[] args) {

        double media=0,exam,mediafinal;

        double n[] = new double[4];

        int i;

        Scanner ler = new Scanner(System.in);

        for(i=0;i<4;i++){

            System.out.printf("Digite a N[%d] = ",i+1);

            n[i]=ler.nextDouble();

            media+=n[i];

        }

        media/=4;

        if(media>=7){

            System.out.println("APROVADO");

        }else{

            exam=ler.nextDouble();

            mediafinal=(exam+media)/2;

            if(mediafinal>=7){

                System.out.println("APROVADO");

            }else{

                System.out.println("REPROVADO");

            }

        }

    }

}

```



(B)Haskell

```
import System.IO
```

```
main = do
```

```
    putStrLn "Please input a number."
```

```
    inputjar <- getLine
```

```
    let n = read inputjar :: Int
```

```
    if n `mod` 2 == 0
```

```
        then putStrLn "Par"
```

```
        else putStrLn "Impar"
```

(B)Java

```
import java.util.Scanner;
```

```
public class Questao7B {
```

```
    public static void main(String[] args) {
```

```
        Scanner ler = new Scanner(System.in);
```

```
        int n;
```

```
        System.out.println("Digite um número qualquer");
```

```
        n=ler.nextInt();
```

```
        if(n%2==0){
```

```
            System.out.println("Par");
```

```
        }else{
```

```
            System.out.println("Impar");
```

```
        }
```

```
    }
```

```
}
```

(C)Haskell

```
swp[x]=[x]
```

```
swp(x:y:zs)
```

```
    |x>y=y:swp(x:zs)
```

```
    |otherwise=x:swp(y:zs)
```

main = do

```
let k=[1,7,6,8,40,-45,3]
```

```
let m=srt(k)
```

```
print("Array normal")
```

```
print(k)
```

```
print("Array ordenada")
```

```
print(m)
```

(C)Java

```
import java.util.Scanner;
public class Questao7C {
    public static void main(String[] args) {
        Scanner ler = new Scanner(System.in);
        int tam,i,j;
        System.out.print("Digite o tamanho do vetor: ");
        tam=ler.nextInt();
        int A[] = new int[tam];
        for(i=0;i<tam;i++){
            System.out.printf("Valor de A[%d] = ",i); A[i]=ler.nextInt();
        }
        System.out.println("Array normal:");
        for(i=0;i<tam;i++){
            System.out.printf("Vetor A[%d] = %d\n",i,A[i]);
        }
        for(i=1;i<tam;i++){
            j=i;
            while(j>0 && A[j-1]>A[j]){
                A[j-1]^=A[j];
                A[j]^=A[j-1];
                A[j-1]^=A[j];
                j=j-1;
            }
        }
        System.out.println("\nArray ordenado:");
        for(i=0;i<tam;i++){
            System.out.printf("Vetor A[%d] = %d\n",i,A[i]);
        }
    }
}
```

(D)Haskell

maxi [x] = x

maxi (x:xs)

  | x > maxi xs = x

  | otherwise = maxi xs

mini [x] = x

mini (x:xs)

  | x < mini xs = x

  | otherwise = mini xs

(D)Java

```
import java.util.Scanner;
public class Questao7D {
    public static void main(String[] args) {
        Scanner ler = new Scanner(System.in);
        int x,i,j,maior,menor;
        System.out.print("Digite o tamanho do Vetor = "); x=ler.nextInt();
        int vet[] = new int[x];
        for(i=0;i<x;i++){
            System.out.printf("Digite o valor do Vetor[%d] = ",i);
            vet[i]=ler.nextInt();
        }
        maior = vet[0]; menor = vet[0];
        for(i=1;i<x;i++){
            if(vet[i] < menor){
                menor = vet[i];
            }

            if(vet[i] > maior){
                maior = vet[i];
            }
        }
        System.out.printf("\nMaior valor do vetor = %d\nMenor valor do\nvetor = %d\n",maior,menor);
    }
}
```

**Jaqueline Miranda Neves**