

LISTA 2

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

Programação funcional é um paradigma onde os programas são executados por meio de avaliação de funções, sem haver mudança de estado. Assim, não existem sentenças de atribuição e nem variáveis no sentido imperativo, e como iterações são regidas pela mutabilidade das variáveis elas não são possíveis nesse paradigma.

Vantagens: Gera códigos completos e autossuficiente, seu código é mais fácil de ser refeito, receber mudanças e manutenções com facilidade, facilidade nos testes e na busca por bugs os testes podem ser direcionados para expressões específicas, possibilitando programas mais exatos e imunes a erros.

Desvantagens: Compiladores/interpretadores mais complexos, difícil prever os custos de execução (tempo/espço), alguns algoritmos são mais eficientes quando implementados de forma imperativa.

SITE: <http://www.dcc.fc.up.pt/~pbv/aulas/pf/slides/aula1.pdf>

[Questão 02] 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.

1) **Google:**

Haskell é usada em um pequeno número de projetos internos no Google, para suporte interno de infraestrutura de TI e o projeto de código aberto Ganeti. Ganeti é uma ferramenta para gerenciar clusters de servidores virtuais criados em Xen e KVM.

Convert.rocks:

Scala é utilizado na API HTTP, todo recebimento, transformação, análises e disparos de eventos é basicamente Scala + outros.

Microsoft:

Haskell é usada para seu sistema de serialização de produção, Bond. Bond é amplamente utilizado na Microsoft em serviços de alta escala. A Microsoft Research, separadamente, foi um dos principais patrocinadores do desenvolvimento da Haskell desde o final da década de 1990.

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

[Questão 03] Defina Orientação a Objetos (OO) e apresente benefícios.

Orientação Objeto é um paradigma de análise que combina estrutura de dados e comportamentos funcionais de sistemas de software baseado na composição e interação entre diversas unidades de software chamadas de objetos. Seus benefícios são a 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 extensividade, legibilidade e reutilização de códigos.

Benefícios:

- A codificação fica mais próxima do cenário real do problema a ser resolvido;
- A manutenção futura fica mais simples e rápida: Pode-se somente alterar as classes que são necessárias, sem precisar ficar procurando funções perdidas em meio aos arquivos para alterar algo;
- Maior reutilização de código;
- Padronização do sistema: criando um sistema em programação procedural, ele fica com o padrão do programador ou da empresa;
- Segurança: é muito mais difícil burlar um sistema orientado a objetos do que um sistema procedural, uma vez que não basta simplesmente inserir alguma função ou algo do gênero, pois é preciso instanciar objetos para que a “coisa funcione”. Ainda existe a possibilidade de definir propriedades e métodos públicos, protegidos ou privados, aumentando o nível de segurança;
- Herança de código: uma classe *pessoas* com as propriedades *nome*, *email* e *telefone* e precise gerenciar pessoas físicas (com CPF) e jurídicas (com CNPJ). Basta criar uma classe *física* herdando tudo da classe *pessoas* e adicionando a propriedade *CPF*, e fazer a mesma coisa para jurídica;

SITE: 1) <http://dev.rbtech.info/vantagens-de-programar-com-orientacao-a-objetos-oop/>
2) <https://wpjr2.wordpress.com/2008/04/23/orientacao-por-objetos-vantagens-e-desvantagens/>

[Questão 04] No programa em java abaixo, apresente e determine: o nível de acesso; o escopo; e o tipo (primitivo ou objeto) de cada variável/atributo.

```
1. public class Car
2. {
3.     int year; // Default; Atributo; Primitivo;
4.     String make; // Default; Atributo; Objeto;
5.     double speed; // Default; Atributo; Primitivo;
6.     private int num = 175; // Privado (Só dentro da classe); Atributo; Primitivo;
7.
8.     public Car(int y // Default; Variável; Primitivo, String m // Default; Variável; Objeto,
double beginningSpeed // Default; Variável; Primitivo)
9.     {
10.         year = y;
11.     }
12.
13.     public int getYear()
14.     {
15.         int tmp = year; // Default; Variável; Primitivo
16.         Roda r = new Roda(tmp); // Default; Variável; Objeto
17.         return year;
18.     }
19. }
```

**** Observações: Atributo = Variável no escopo da classe;**
Variável = Variável no escopo do método;

[Questão 05] 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.

O erro está ao tentar colocar espaços na memória onde não foram destinados.