PROCESSO SELETIVO PARA ESTÁGIO REMUNERADO NO TRIBUNAL DE CONTAS DO ESTADO DO MARANHÃO

ENSINO SUPERIOR - INFORMÁTICA

LEIA ATENTAMENTE AS INSTRUÇÕES ABAIXO.

Você recebeu um CADERNO DE QUESTÕES contendo 5 questões.

Você recebeu, também, uma FOLHA-RESPOSTA, onde deve escrever o seu nome, o número do seu documento de identificação e assinar no espaço próprio.

Na FOLHA RESPOSTA, as respostas às questões deverão ser devidamente identificadas e escritas na ordem correspondente ao número da questão.

Escreva somente dentro das linhas impressas. As respostas escritas em entrelinhas ou sobrescritas serão desconsideradas

As respostas de cada questão deverão ser redigidas de forma contínua, não podendo ser intercaladas com respostas de outra questão.

Tenha muito cuidado com o FOLHA-RESPOSTA, para não o dobrar, amassar, ou manchar.

O fiscal de sala NÃO está habilitado a esclarecer as questões propostas, dado o caráter técnico dos enunciados.

Utilize letra legível e evite rasuras.

NÃO serão fornecidas folhas extras para resposta.

SERÁ ELIMINADO do processo seletivo o candidato que:
a) se utilizar, durante a realização das provas, de máquinas e/ou relógios de calcular, bem como de rádios gravadores, headphones, telefones celulares ou fontes de consulta de qualquer espécie;
b) se ausentar da sala em que se realizam as provas levando consigo o CADERNO DE QUESTÕES e/ou o FOLHA-RESPOSTA;
c) assinar o FOLHA-RESPOSTA fora no espaço próprio ou inserir qualquer caractere, símbolo ou meio de identificação no espaço destinado às respostas.

O candidato só poderá se ausentar do recinto das provas após 1 (uma) hora contada a partir do efetivo início das mesmas. Por motivos de segurança, o candidato NÃO PODERÁ LEVAR O CADERNO DE QUESTÕES, a qualquer momento.

Verifique se este material está em ordem e se seus dados conferem com os que aparecem no FOLHA-RESPOSTA. Caso contrário, notifique o fato IMEDIATAMENTE ao fiscal.

O resultado da correção será divulgada na data prevista no Edital do Processo Seletivo.

1. Java Persistence API (ou simplesmente **JPA**) é uma API padrão da linguagem Java que descreve uma interface comum para frameworks de persistência de dados. A JPA define um meio de mapeamento objeto-relacional para objetos Java simples e comuns (POJOs), denominados *beans de entidade*. Diversos frameworks de mapeamento objeto/relacional como o Hibernate implementam a JPA. Também gerencia o desenvolvimento de entidades do Modelo Relacional usando a plataforma nativa Java SE e Java EE.

Originou-se num projeto comum entre os desenvolvedores para se criar o padrão. Fortemente baseado nas ideias trazidas pelo Hibernate, tanto que o líder da primeira versão dessa especificação é o criador do framework.

Utilizando a JPA transforme as tabelas de banco de dados abaixo juntamente com os seus relacionamentos em um modelo baseado em objetos Java(Classes de Entidade e relacionamentos)

SETOR		
Tipo	Campo	Tamanho
Int (PK)	SETOR_ID	
Text	NOME	255
PARTE		
Tipo	Campo	Tamanho
Int (PK)	PARTE_ID	
Text	NOME	255
Text	CPF	11
PROCESSO		
Tipo	Campo	Tamanho
Int (PK)	PROCESSO_ID	
Int	NUMERO	
Int	ANO	
Int	EXERCICIO	
Text	DESCRICAO	4000
Int (FK)	PARTE_ID	
Int(FK)	SETOR_ATUAL_ID	
Date	DATA_ENTRADA	

ANDAMENTO		
Tipo	Campo	Tamanho
Int (PK)	ANDAMENTO_ID	
Int (FK)	PROCESSO_ID	
Date	DATA	
Int (FK)	SETOR_ID	

Obs1: Um processo pode ter 1 ou vários andamentos

Obs2: Se preferir, pode realizar o mapeamento diretamente nos campos e suprimir os métodos getters e setters

2. Um *Data Access Object* (DAO) é um objeto que provê uma interface abstraindo um banco de dados ou algum outro mecanismo de persistência externo à aplicação.

Um repositório implementa parte das regras de negócio no que se refere à composição das entidades. Ele é fortemente **vinculado ao domínio da aplicação** e este é um reflexo direto das regras de negócio, pois **ele abstrai armazenamento e consulta de um ou mais entidades de domínio**.

Frameworks como Hibernate (Java) ou NHibernate (.NET) disponibilizam uma interface bem mais próxima ao padrão *Repository* porque:

1. Trabalham com coleções. Vide métodos persist() e find().
2. As pesquisas são realizadas por uma interface independente do tipo de banco de dados.
3. Realizam o trabalho de construir e armazenar as entidades da aplicação, inclusive com a possibilidade de recuperar valores de relações automaticamente, como no exemplo acima sobre a lista de endereços do cliente.

Além disso, eles também fazem o papel do DAO, pois:

Armazenam dados diretamente num banco de dados.

- Geral as consultas SQL necessárias.
- Mapeiam os tipos de dados da fonte para um tipo adequado nos objetos e vice-versa.

Diante do exposto acima, utilizando JPA/HIBERNATE crie os repositórios das classes de entidade que foram implementadas na questão anterior. O repositório de cada entidade deve possuir os métodos básicos que consigam:

- salvar uma entidade
- atualizar uma entidade
- remover uma entidade
- recuperar uma entidade
- recuperar todas as entidades

3- Surgiu uma nova demanda por relatórios sobre os processos que tramitam no Tribunal de Contas do Estado do Maranhão. Para suprir essa demanda, utilizando a linguagem de consultas JPQL crie no repositório de processos, implementado na questão anterior as seguintes buscas.

a) Crie uma consulta que dado um número de CPF de uma parte, retorne todos os processos desta parte que são do exercício 2017 e com data de entrada ente 01/01/2017 e 31/05/2017

b) Crie uma consulta que dado um setor, retorne a quantidade de processos que estão atualmente neste setor

c) Crie uma consulta que retorne a quantidade de processos para cada parte

d) Crie uma consulta que dado um processo, calcule a diferença em dias entre o seu último e o primeiro andamento.

4- Utilizando ainda como base as tabelas da questão 1, e utilizando a linguagem SQL escreva uma consulta que retorne a quantidade de processos que entraram no Tribunal de Contas do Estado do Maranhão (DATA_ENTRADA) em cada mês do ano durante o exercício 2017.

5- Considere as classes abaixo:

```
public enum Profissao {
   ADVOGADO,CONTADOR;
}

public class Pessoa {

   private String cpf;
   private String nome;
   private Profissao profissao;

   public Pessoa(String cpf,String nome,Profissao profissao){
   this.cpf = cpf;
   this.nome = nome;
   this.profissao = profissao;
   }
```

```
    public String getCpf() {
        return cpf;
    }


    public String getNome() {
        return nome;
    }

    public Profissao getProfissao() {
        return profissao;
    }
}

public class Contador {


    private String cpf;
    private String nome;

    public Contador(String cpf,String nome){
    this.cpf = cpf;
    this.nome = nome;
    }


    public String getCpf() {
        return cpf;
    }
```

```
    public String getNome() {
        return nome;
    }

}
```

Converta a lista de objetos **Pessoa** em uma lista de objetos **Contador**

```
public class ContadorConversor {

    public static void main(String[] args) {


        List<Pessoa> pessoas = Arrays.asList(
                new Pessoa("23453456543","Ricardo",Profissao.ADVOGADO),
                new Pessoa("89658907654","Leandro",Profissao.CONTADOR),
                new Pessoa("06543878877","Ricardo",Profissao.ADVOGADO),
                new Pessoa("65776543234","Bruno",Profissao.ADVOGADO),
                new Pessoa("09812365476","Jacó",Profissao.CONTADOR),
                new Pessoa("09613579641","Jonatas",Profissao.CONTADOR)
        );


    }

}
```

FOLHA-RESPOSTA

Nome do candidato por extenso	Número do documento de identificação	Assinatura do candidato