|  |  |  |
| --- | --- | --- |
| Camada | Arquivos | Responsabilidade |
| Controle | Controle.java | Recebe as requisições e direciona as ações para o destino correto. |
| Modelo | Pessoa.java  PessoaDAO.java | Contém as fórmulas de cálculos, as regras de negócios, assim como as restrições. |
| Visão | Index.html  cadastrar.jsp | Apresentar resultados, conteúdos, mensagens etc. |

1. Com base no projeto abordado em aula, referente ao conceito de padrão arquitetural MVC, escreva uma instrução para cada item:
2. instrução que evidencia o acoplamento entre as camadas visão e controle.

controller.requisitar(v);

1. instrução que evidencia o acoplamento entre as camadas controle e modelo.

dao.inserir(**new** Cliente(nome, email));

3) A definição de uma classe interna pode ter modificador privado? Crie um exemplo prático e funcional.

public class Cliente {

    private String nome;

    private String email;

    public Controle() {

        dao = **new** ClienteDAO();

    }

4) Responda às questões referentes à classe anônima.

1. O que é uma classe anônima?

A classe define e instancia ao mesmo tempo.

Útil quando desejamos realizar algum processamento em um único ponto do código.

1. Escreva um exemplo prático e funcional.

public class Anonima {

   Exibir nome=new Exibir() {

         @Override

             public void exibirNome() {

                 System.out.println("Iago");

             }

         };

         nome.exibirNome();

    }

}

5) O que são expressões lambdas? Escreva um exemplo prático e funcional.

public class Teste {

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

*Runnable* r1 = () *->* {

            Singleton sin = Singleton.getObject();

            System.out.println(sin);

    };

    Thread t1 = **new** Thread(r1);

    t1.start();

}

}

6) Uma interface funcional deve ter, obrigatoriamente, a anotação @FunctionalInterface? O que caracteriza uma interface funcional? Cite duas interfaces funcionais nativas do Java. Explique o que é um interface funcional mencionando o uso da referida anotação. COMPARATOR E CONSUMER

* 1. É correto afirmar que um padrão arquitetural tem como base um design pattern, do catálogo GoF? Explique, relacionando arquitetura de software e design pattern.

* 1. Escreva as classes anônimas correspondentes às expressões lambdas.

|  |
| --- |
| Runnable r1=()->{    Singleton s = Singleton.getObject();    System.out.println(s);  }; |

|  |
| --- |
| Consumer consumer=(Object obj) -> {              Pessoa p=(Pessoa) obj;              System.out.println(p);  }; |

|  |
| --- |
| Comparator<Pessoa> compara = (Pessoa p1, Pessoa p2) -> {              if (p1.getNome().compareTo(p2.getNome()) > 0) {                  return 1;              } else if (p1.getNome().compareTo(p2.getNome()) < 0) {                  return -1;              } else {                  return 0;              }          }; |

[Respostas](https://docs.google.com/document/d/1HZbB0kBijBZyTzik2MfJHUfFD5VDZZR8ZoBO1VBW_os/edit)

* 1. Elabore um projeto MVC com TAD Aluno, nome, email, curso (String), id (int). O id deve ser auto incrementado, iniciando em um.

Na classe DAO, elabore as operações: cadastrar, atualizar email pelo id, remover pelo id, pesquisar pelo id.

Na camada view elabore uma GUI conforme modelo.

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

* 1. Uma empresa tem um serviço newsletter para divulgar em primeira mão, notícias sobre os seus produtos. Aplique o padrão adequado à situação, para que assinantes do serviço recebam estas notícias.

* 1. Considere um padrão composite e implemente o padrão Facade para minimizar a codificação na classe Cliente. Na classe Facade proponha pelo menos três produtos, por exemplo, cesta de café da manhã, cesta de aniversário, cesta de natal. Escreva uma classe cliente (principal) e mostre que houve melhora com a utilização do facade.

1. Considerando o pattern state e o exemplo publicado neste documento. Refaça a solução apresentada utilizando Enum.