**Trabalho da Disciplina de Programação Concorrente e Paralela**

Prof. Paulo Lacerda

Nome: Henrique Alexandrino Pires Paes RA: 1320855 Semestre: 7º

Nome: Elder Sergio Xavier RA: Semestre: 7º

Ciência da Computação

1. Qual a diferença entre Computação distribuída e Computação Paralela? Dê exemplos.

**Resposta**: Computação distribuída é quando vários computadores estão executando um sistema único por exemplo o uso de um software que divide um programa em várias tarefas para que assim todos possa carregar um único sistema e diminuir “lentidão”.

Computação paralela é quando um programa executa diversas tarefa tendo seu princípio termina todas elas concorrentemente.

1. Qual a diferença entre um sistema fortemente acoplado e fracamente acoplado? Dê exemplos.

**Resposta:** Sistema fortemente acoplado tem maior capacidade de processamento e geralmente é conhecido como “multiprocessador”

Sistema fracamente acoplado geralmente tem um sistema com pouco processamento.

1. O que são sistemas coesos? Quais os tipos de coesão?

**Resposta** é a melhor forma de criar um sistema deixando coerente e legível, um exemplo é a criação de orientação a objeto que tende a ter maior coesão para melhor entendimento.

1. Cite dois exemplos de Sistemas de Computação distribuída e paralela?

**Resposta:** Exemplo de sistema distribuído é o ARIVO que distribui suas competências administrativas para melhor delas.

Exemplo de sistema paralela são maquinas antigas que utilizava prioridade de gravação exemplo quando estava imprimindo algo. Se pedisse para gravar ele pausava o processo de impressão e fazia a gravação com prioridade.

1. O que podemos definir como um **THREAD**?

**Resposta:** TREADS são subi tarefas em um sistema, um exemplo é um software aberto porem existem subi tarefas executando no sistema principal.

1. Por que o sistema operacional Android não consegue operacionalizar uma chamada a uma Tela (Activity) e enviar um e-mail sem o uso de uma Thread?

**Resposta:** Porque os aparelhos Android ser single core impossibilitando a utilização de múltiplos threads com a criação de processadores multicores ira ter uma melhora no processo.

1. Exiba 50 números sorteados de 1 a 100 para o usuário.

**Resposta:**

**package exercicio7;**

**import java.util.Random;**

**public class Exercicio7 {**

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

**Random t1 = new Random();**

**for (int n=1;n<=50;n++)**

**{**

**int sorteio = t1.nextInt(99);**

**System.out.println(sorteio+1+"\n");**

**}**

**}**

**}**

1. Crie um jogo para o usuário descobrir um número sorteado de 1 a 100. A cada tentativa dele, forneça uma dica mostrando se o número é maior ou menor. Quando ele descobrir exiba uma mensagem de parabéns e mostre em quantas tentativas ele conseguiu.

**Resposta:**

**Obs: todos os códigos estao em https://github.com/elderxavier/prog\_concorrente\_atividadeI**

**package br.edu.sumare.pergunta8;**

**import java.util.Random;**

**public class Pergunta8 {**

**private final int sorteado;**

**private static int contador;**

**public Pergunta8() {**

**Random t1 = new Random();**

**this.sorteado = t1.nextInt(99) + 1;**

**}**

**public void Executa() {**

**//System.out.printf(String.valueOf(this.sorteado));**

**String info;**

**java.util.Scanner scanner = new java.util.Scanner(System.in);**

**boolean flag = false;**

**while (!flag) {**

**System.out.println("Digite um numero inteiro: ");**

**if (scanner.hasNextLine()) {**

**info = scanner.nextLine();**

**flag = this.Compara(info);**

**}**

**}**

**}**

**private boolean Compara(String num) {**

**contador++;**

**int convert = 0;**

**boolean ret = false;**

**convert = this.isNumero(num) ? Integer.parseInt(num) : 0;**

**if (convert > this.sorteado) {**

**System.out.println("O numero é maior:");**

**} else if (convert < this.sorteado) {**

**System.out.println("O numero é menor:");**

**} else {**

**System.out.println("Parabens, você acertou em "+ String.valueOf(contador) + " vez(es)!");**

**ret = true;**

**}**

**return ret;**

**}**

**private boolean isNumero(String numero) {**

**boolean ret = true;**

**int convert = 0;**

**try {**

**convert = Integer.parseInt(numero);**

**} catch (NumberFormatException ne) {**

**ret = false;**

**}**

**return ret;**

**}**

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

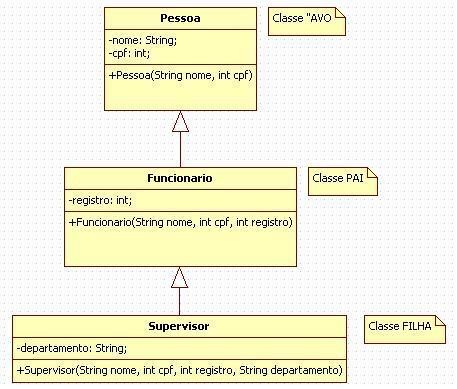
**Pergunta8 pergunta = new Pergunta8();**

**pergunta.Executa();**

**}**

**}**

1. Crie o script que representa o diagrama abaixo:



**Resposta:**

**package br.edu.sumare.pergunta9;**

**public class Pessoa {**

**private String nome;**

**private int cpf;**

**public Pessoa( String nome, int cpf) {**

**this.nome = nome;**

**this.cpf = cpf;**

**}**

**}**

**package br.edu.sumare.pergunta9;**

**public class Funcionario extends Pessoa {**

**private int registro;**

**public Funcionario(String nome, int cpf, int registro) {**

**super(nome, cpf);**

**this.registro = registro;**

**}**

**}**

**package br.edu.sumare.pergunta9;**

**public class Supervisor extends Funcionario{**

**private String departamento;**

**public Supervisor(String nome, int cpf, int registro, String departamento) {**

**super(nome, cpf, registro);**

**this.departamento = departamento;**

**}**

**}**

1. Dada uma sequência de números inteiros não-nulos, seguida por 0, imprimir seus quadrados.

**Resposta:**

**package br.edu.sumare.pergunta10;**

**import java.util.ArrayList;**

**import java.util.List;**

**public class Pergunta10 {**

**public void Executa() {**

**List<Integer> minhalista = new ArrayList();**

**String info;**

**java.util.Scanner scanner = new java.util.Scanner(System.in);**

**boolean flag = false;**

**int parse = 1;**

**while (!flag) {**

**System.out.println("Digite um numero inteiro: ");**

**if (scanner.hasNextLine()) {**

**info = scanner.nextLine();**

**parse = isNumero(info) ? Integer.parseInt(info): 1;**

**minhalista.add(parse);**

**flag = parse == 0 ? true: false;**

**}**

**}**

**int cont = 1;**

**for(int item : minhalista ){**

**if(cont < minhalista.size()){**

**System.out.println("O quadrado de: "+ String.valueOf(item) + " é : "+ String.valueOf(item \* item) );**

**}**

**cont++;**

**}**

**}**

**private boolean isNumero(String numero) {**

**boolean ret = true;**

**int convert = 0;**

**try {**

**convert = Integer.parseInt(numero);**

**} catch (NumberFormatException ne) {**

**ret = false;**

**}**

**return ret;**

**}**

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

**Pergunta10 pergunta = new Pergunta10();**

**pergunta.Executa();**

**}**

**}**

1. Uma loja de discos anota diariamente durante o mês de março a quantidade de discos vendidos. Determinar em que dia desse mês ocorreu a maior venda e qual foi a quantidade de discos vendida nesse dia.

**Resposta:**

**package br.edu.sumare.pergunta11;**

**public class Pergunta11 {**

**public void Executa(){**

**int[] dias = new int[31];**

**String info;**

**java.util.Scanner scanner = new java.util.Scanner(System.in);**

**boolean flag = false;**

**int parse = 1;**

**int diamais = 0;**

**int aux = 0;**

**for(int i = 0; i < dias.length; i++){**

**System.out.println("Infor a quntidade vendida no dia " +String.valueOf(i+1));**

**if (scanner.hasNextLine()) {**

**info = scanner.nextLine();**

**parse = isNumero(info) ? Integer.parseInt(info): 0;**

**aux = parse;**

**if( parse > aux){**

**aux = parse;**

**diamais = i +1;**

**}**

**dias[i] = parse;**

**}**

**}**

**System.out.println("Maior quntidade vendida no dia " +String.valueOf(diamais));**

**}**

**private boolean isNumero(String numero) {**

**boolean ret = true;**

**int convert = 0;**

**try {**

**convert = Integer.parseInt(numero);**

**} catch (NumberFormatException ne) {**

**ret = false;**

**}**

**return ret;**

**}**

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

**Pergunta11 pergunta = new Pergunta11();**

**pergunta.Executa();**

**}**

**}**

12)Dizemos que um número natural é *triangular* se ele é produto de três números naturais consecutivos. Exemplo: 120 é triangular, pois 4.5.6 = 120. Dado um inteiro não-negativo *n*, verificar se *n* é triangular.

**Resposta:**

**package br.edu.sumare.pergunta12;**

**public class Pergunta12 {**

**public void Executa() {**

**int[] dias = new int[31];**

**String info;**

**java.util.Scanner scanner = new java.util.Scanner(System.in);**

**boolean flag = false;**

**int parse = 1;**

**int aux = 0;**

**int aux2 = 0;**

**System.out.println("Informe um número inteiro: ");**

**if (scanner.hasNextLine()) {**

**info = scanner.nextLine();**

**parse = isNumero(info) ? Integer.parseInt(info) : 0;**

**}**

**int limit = parse < 10 ? (parse - 2): ( (parse / 3) -2);**

**for(int i = 0; i < limit; i++){**

**aux = i \* (i+1)\* (i+2);**

**if(aux == parse){**

**flag = true;**

**aux2 = i;**

**break;**

**}**

**}**

**if(flag){**

**System.out.println("O numero " + String.valueOf(parse) + "é um triangular, pois ");**

**System.out.println(String.valueOf( aux2 \* (aux2 +1) \* (aux2 +2) )+ " é igual "+ String.valueOf(parse) );**

**}else{**

**System.out.println("O numero " + String.valueOf(parse) + " não é um triangular");**

**}**

**}**

**private boolean isNumero(String numero) {**

**boolean ret = true;**

**int convert = 0;**

**try {**

**convert = Integer.parseInt(numero);**

**} catch (NumberFormatException ne) {**

**ret = false;**

**}**

**return ret;**

**}**

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

**Pergunta12 pergunta = new Pergunta12();**

**pergunta.Executa();**

**}**

**}**

13)Crie uma classe para representar uma conta corrente, com métodos para depositar uma quantia, sacar uma quantia e obter o saldo. Para cada saque será debitada também uma taxa de operação equivalente à 0,5% do valor sacado. Crie, em seguida, uma subclasse desta classe anterior para representar uma conta corrente de um cliente especial. Clientes especiais pagam taxas de operação de apenas 0,1% do valor sacado. Faça testes com as duas classes e verifique seus resultados.

**Resposta:**

**package br.edu.sumare.pergunta13;**

**public class Contacorrente {**

**private static double saldo;**

**public static double getSaldo() {**

**return saldo;**

**}**

**public static void setSaldo(double aSaldo) {**

**saldo = aSaldo;**

**}**

**public void depositar(double deposito){**

**setSaldo(getSaldo() + deposito);**

**}**

**public void sacar(double saque){**

**setSaldo(getSaldo() - saque - (saque \* 0.05)) ;**

**}**

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

**Contacorrente conta = new Contacorrente();**

**conta.depositar(100.00);**

**conta.sacar(10.00);**

**System.out.println("Saldo: " + String.valueOf(conta.getSaldo()));**

**}**

**}**

**package br.edu.sumare.pergunta13;**

**public class Clienteespecial extends Contacorrente{**

**@Override**

**public void sacar(double saque){**

**setSaldo(getSaldo() - saque - (saque \* 0.01)) ;**

**}**

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

**Clienteespecial conta = new Clienteespecial();**

**conta.depositar(100.00);**

**conta.sacar(10.00);**

**System.out.println("Saldo: " + String.valueOf(conta.getSaldo()));**

**}**

**}**

14)Crie uma hierarquia de classes de domínio para uma loja que venda livros, CDs e DVDs. Sobrescreva o método toString() para que imprima:

* Para livros: nome, preço e autor; • Para CDs: nome, preço e número de faixas;
* Para DVDs: nome, preço e duração.

Evite ao máximo repetição de código utilizando a palavra super no construtor e no método sobrescrito. Em seguida, crie uma classe Loja com o método main() que adicione 5 produtos diferentes (a sua escolha) a um vetor e, por fim, imprima o conteúdo do vetor.

**Resposta:**

**package br.edu.sumare.pergunta14;**

**public abstract class Midia {**

**private String nome;**

**private String preco;**

**public Midia(String nome, String preco) {**

**this.nome = nome;**

**this.preco = preco;**

**}**

**public String toString(){**

**return "Nome: "+ this.getNome() + ", Preco: "+ this.getPreco();**

**}**

**public String getNome() {**

**return nome;**

**}**

**public String getPreco() {**

**return preco;**

**}**

**}**

**package br.edu.sumare.pergunta14;**

**public class Cds extends Midia{**

**private String autor;**

**public Cds(String nome, String preco, String autor) {**

**super(nome, preco);**

**this.autor = autor;**

**}**

**@Override**

**public String toString(){**

**return "Nome: "+ this.getNome() + ", Preco: "+ this.getPreco() + ", Autor: "+ this.getAutor();**

**}**

**public String getAutor() {**

**return autor;**

**}**

**}**

**package br.edu.sumare.pergunta14;**

**public class Dvds extends Midia{**

**private String duracao;**

**public Dvds(String nome, String preco, String duracao) {**

**super(nome, preco);**

**this.duracao = duracao;**

**}**

**@Override**

**public String toString(){**

**return "Nome: "+ this.getNome() + ", Preco: "+ this.getPreco() + ", Autor: "+ this.getDuracao();**

**}**

**public String getDuracao() {**

**return duracao;**

**}**

**}**

**package br.edu.sumare.pergunta14;**

**public class Loja {**

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

**Object[] obj = new Object[5];**

**Dvds dvd = new Dvds("Matrix", "R$ 35,00", " 120 min");**

**obj[0]=dvd;**

**dvd = new Dvds("Matrix II", "R$ 50,00", " 120 min");**

**obj[1]=dvd;**

**dvd = new Dvds("Matrix III", "R$ 55,00", " 120 min");**

**obj[2]=dvd;**

**Cds cd = new Cds("Cazuza", "25", "Cazuza");**

**obj[3]= cd;**

**cd = new Cds("Um abraçaço", "20", "Caetano Veloso");**

**obj[4]= cd;**

**for(Object o : obj){**

**System.out.println( o.toString() );**

**}**

**}**

**}**