

O que deve ser entregue:

- 1) Deve ser enviado por email um arquivo zipado contendo os códigos fontes dos programas implementados;

Regras para entrega do trabalho prático:

Este trabalho deve ser realizado em grupo de no máximo 3 alunos. Cada grupo deve escolher um representante que será responsável por enviar o trabalho em um arquivo zipado, cujo nome deve respeitar o seguinte formato: CK069_<matricula do aluno>_T<numero do trabalho pratico>, por exemplo, CK069_123456_T1 é o arquivo contendo o trabalho pratico 1 do estudante com matricula 123456. O arquivo compactado deve ser enviado pelo SIGAA Caso o aluno não respeite as regras apresentadas anteriormente, poderá sofrer redução na nota do trabalho.

Como será avaliado o trabalho:

Os grupos que resolverem corretamente a questão no prazo estipulado pelo professor, receberão um bônus de 1.0 ponto na primeira prova. Os grupos que entregarem o exercício resolvido, no prazo estipulado, ganham os pontos relacionados ao laboratório conforme a pontuação a seguir.

Trabalho Prático 5 – Sincronização Processos (DESAFIO)

Seu grupo foi contratado pela empresa XPTO para resolver um problema em um dos seus sistemas. O código abaixo apresenta um programa que implementa a funcionalidade problemática deste sistema. Neste programa são criados dois consumidores e um produtor. Um consumidor consome somente números pares e outro consumidor consome somente números ímpares. O único produtor existente, gera números pares e ímpares, indistintamente. Claramente, um número consumido deve ser eliminado logo após sua utilização. Analise o código abaixo, discuta com seu grupo o problema que está acontecendo e implemente a solução para resolver este problema.

<pre>public class ProducerConsumerExample { private static boolean Even = true; private static boolean Odd = false; public static void main(String[] args) { Dropbox dropbox = new Dropbox(); (new Thread(new Consumer(Even, dropbox))).start(); (new Thread(new Consumer(Odd, dropbox))).start(); (new Thread(new Producer(dropbox))).start(); } }</pre>	<pre>public class Dropbox { private int number; private boolean evenNumber = false; public int take(final boolean even) { System.out.format("%s CONSUMIDOR obtem %d.%n", even ? "PAR" : "IMPAR", number); return number; } public void put(int number) { this.number = number; evenNumber = number % 2 == 0; System.out.format("PRODUTOR gera %d.%n", number); } }</pre>
<pre>import java.util.Random; public class Producer implements Runnable { private Dropbox dropbox; public Producer(Dropbox dropbox) { this.dropbox = dropbox; } public void run() { Random random = new Random(); while (true) { int number = random.nextInt(10); try { Thread.sleep(random.nextInt(100)); dropbox.put(number); } catch (InterruptedException e) { } } } }</pre>	<pre>import java.util.Random; public class Consumer implements Runnable { private final Dropbox dropbox; private final boolean even; public Consumer(boolean even, Dropbox dropbox) { this.even = even; this.dropbox = dropbox; } public void run() { Random random = new Random(); while (true) { dropbox.take(even); try { Thread.sleep(random.nextInt(100)); } catch (InterruptedException e) { } } } }</pre>

Tarefas a serem executadas pelo grupo:

1. (4,0) Executar o código apresentado e descrever o problema encontrado;
2. (6,0) Modificar o código proposto e solucionar o problema;