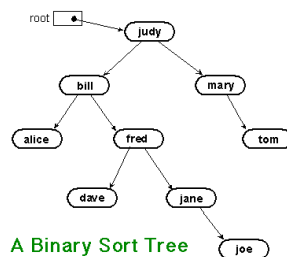


EXERCÍCIOS THREADS E CONCORRÊNCIA

1. Fazer uma busca bi-thread em uma árvore de classificação binária (reutilizar o algoritmo de travessia feito em recursividade). No exemplo abaixo, uma thread percorre o ramo "bill" e a outra o ramo "mary". Ao final da execução das duas threads, o resultado é agrupado e mostrado ao usuário.



2. Em relação aos problemas de concorrência, marque V ou F:
 - a. Exclusão mútua é uma propriedade desejável para recursos críticos e significa que dois ou mais processos podem acessar o recurso ao mesmo tempo.
 - b. Deadlock ocorre quando os processos de um SD entram numa espera eterna por um ou mais recursos que jamais obterão.
 - c. Um banco de dados que não garanta transações atômicas (acesso exclusivo a um objeto para atualizá-lo) pode produzir resultados errados se duas transações tentarem modificá-lo ao mesmo tempo.
 - d. Threads são processos
3. Qual a saída produzida pela execução do código abaixo?

```
1 public class MT implements Runnable
2 {
3     String ms = "S ";
4     public void run()
5     {
6         this.ms = "N ";
7     }
8     public static void main(String[] args)
9     {
10         MT t = new MT();
11         new Thread(t).start();
12         for (int i=0; i < 10; i++)
13             System.out.print(t.ms);
14     }
15 }
```

- a. Erro de compilação
- b. SSSSSSSSSS...
- c. NNNNNNNNNN...
- d. SNSNSNSNSN...
- e. A saída não pode ser determinada

4. Vários objetos da classe A (vide abaixo) são utilizadas num programa multi-thread. O que ocorre quando outras threads utilizam o código abaixo?

```
1  class A
2  {
3      static private int myCount = 0;
4      int yourNumber;
5      private static synchronized int nextCount()
6      {
7          return ++myCount;
8      }
9
10     public void getYourNumber()
11     {
12         yourNumber = nextCount();
13     }
14 }
```

- a. Erro de compilação.
 - b. Erro de execução.
 - c. Cada thread obterá um número único.
 - d. Não se pode garantir que cada thread obterá um número único.
5. O que ocorre no código abaixo quando uma thread t1 invoca p.swap(q) e a thread t2 invoca q.swap(p) concorrentemente, sendo p e q duas instâncias de Swap?

```
1  class Swap {
2      int value;
3      public synchronized int getValue() {
4          return value;
5      }
6      public synchronized void setValue(int i) {
7          value = i;
8      }
9      public synchronized void swap(Swap x) {
10         int tmp = getValue();
11         setValue(x.getValue());
12         x.setValue(tmp);
13     }
14 }
```

6. Baixe o código **JThreadCavalo** e responda às perguntas em comentários no início do Main.java.
- a. É garantido que o programa sempre imprimirá Cavalos comem cenouras?
 - b. Caso não, de duas soluções possíveis para garanti-lo?
7. Modifique o código **JRelogio** incluindo uma nova thread responsável por emitir um alarme sonoro ou visual quando o relógio atingir determinada hora. Esta thread deve ser sincronizada por wait/notify com aquela que roda o relógio.