Universidade Federal de Goiás
Instituto de Informática

POO – Laboratório (26/11/2024): Laboratório Threads

Estudo de Caso: Contagem de Votos em um processo eleitoral brasileiro

(Em grupo, o mesmo do Trabalho Final da Disciplina (TFD))

Submeta as respostas do documento em um único arquivo (.doc, .odt. ou .pdf) submetido em grupo, o grupo do TFD. Somente um estudante do grupo deverá submeter esta Tarefa. Coloque como comentário ao enviar no SIGAA no documento matrícula, nome dos participantes do grupo, turma. Lembro que no arquivo enviado deve ter as identificações dos alunos (matrícula e Nome), do prof. e da disciplina. Embora somente um estudante do grupo deverá enviar a resposta é muito importante cada estudante do grupo ter uma cópia, pois a segunda avaliação será no lab.

Estudante(s) do Grupo matrícula(s) e Nome(s):

1. **Lucas Almeida Oliveira Isaac - 202401802**
2. **Lucca Magnino - 202401805**

Obs.: Baixe os arquivos referentes a Threads:

- **Aula_Threads_parte1.pdf**
- **Aula_Threads_parte2.pdf**
- **Lab_Threads.zip**

Tendo como base as aula teóricas respondam as seguintes questões.

1) **O que é uma *thread*? Dê um exemplo em Java.**

Thread é uma abstração que permite um programa executar diversos trechos simultaneamente. Um exemplo em Java, retirado da aula_parte1, é o seguinte:

```
 package Threads;
/*
 *  A aplicação que cria instância de Thread deve fornecer
 *  o código a ser executado na thread
 *  implementando a classe Runnable
 *  Passando um objeto Runnable
 *  para o construtor  da classe Thread:
 */

public class Thread_Interface implements Runnable {
public void run() {
   System.out.println("Olá de uma thread implementada pela interface
Runnable!");
}

public static void main(String args[]) {
   (new Thread(new Thread_Interface())).start();
 }
}
```

2) **Quais são as semelhanças e diferenças entre um thread do sistema operacional e um thread criada por uma linguagem de programação Orientada a Objetos?**

SEMELHANÇAS:

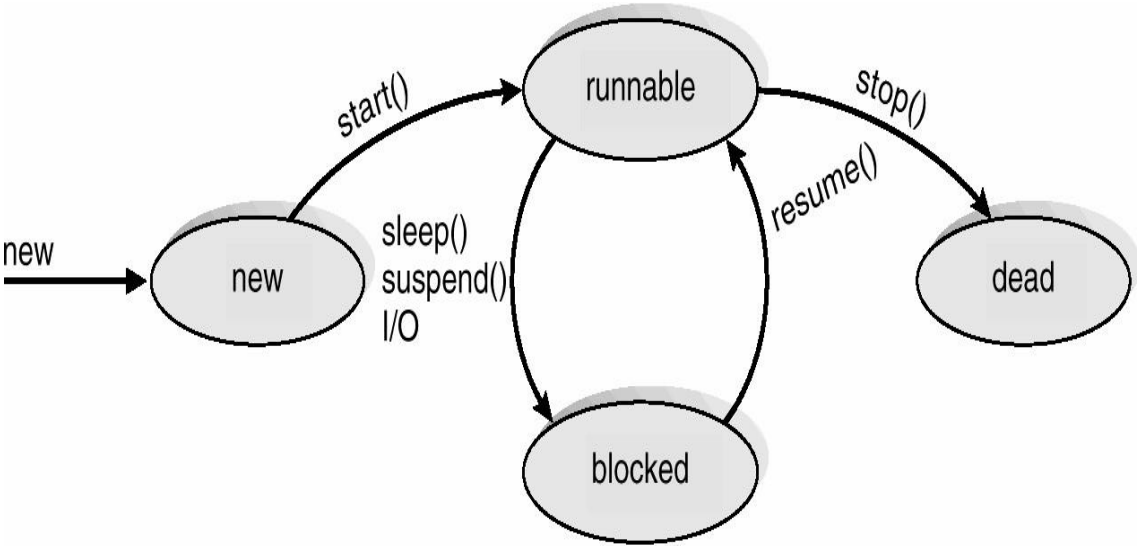
-As duas threads irão executar tarefas simultaneamente ou de forma concorrente.

-Ambas as threads possuem sua própria execução, compartilhando o stack e os registradores com o processo principal, além do compartilhamento de recursos como memória e variáveis.

DIFERENÇAS:

-As threads do Sistema Operacional são gerenciadas pelo kernel do sistema operacional, já em Linguagem de programação orientada a objeto é gerenciada pelo runtime da linguagem (como o JVM).
-A aplicação das threads do Sistema Operacional são voltadas para tarefas críticas de desempenho no sistema, já em linguagem de POO a aplicação são voltadas para atividades específicas como servidores e banco de dados.
-Podemos ver as diferenças na criação do código:
No sistema operacional: pthread_create();
Em POO: new Thread(() -> {...}).start();

3) **A figura abaixo representa os Estados possíveis de uma thread. Explique o que faz cada uma das funções no contexto da figura. Qual destes métodos é usado no código-fonte do lab?**



New: cria um nova instância de uma classe que implementa a interface runnable.

Start(): Inicia a execução de uma thread.

Stop(): Usada para terimnar a execução de uma thread.

Sleep() ou Suspend(): Pausa a execução da thread

Resume(): Retoma a execução após o sleep() ou suspend().

I/O: Operações de entrada e saída realizadas dentro de uma thread.

Os métodos utilizados no código-fonte do laboratório foram: new, start() e operações I/O, segue a print dos usos:

```
public class ContaCargos extends Thread {
```

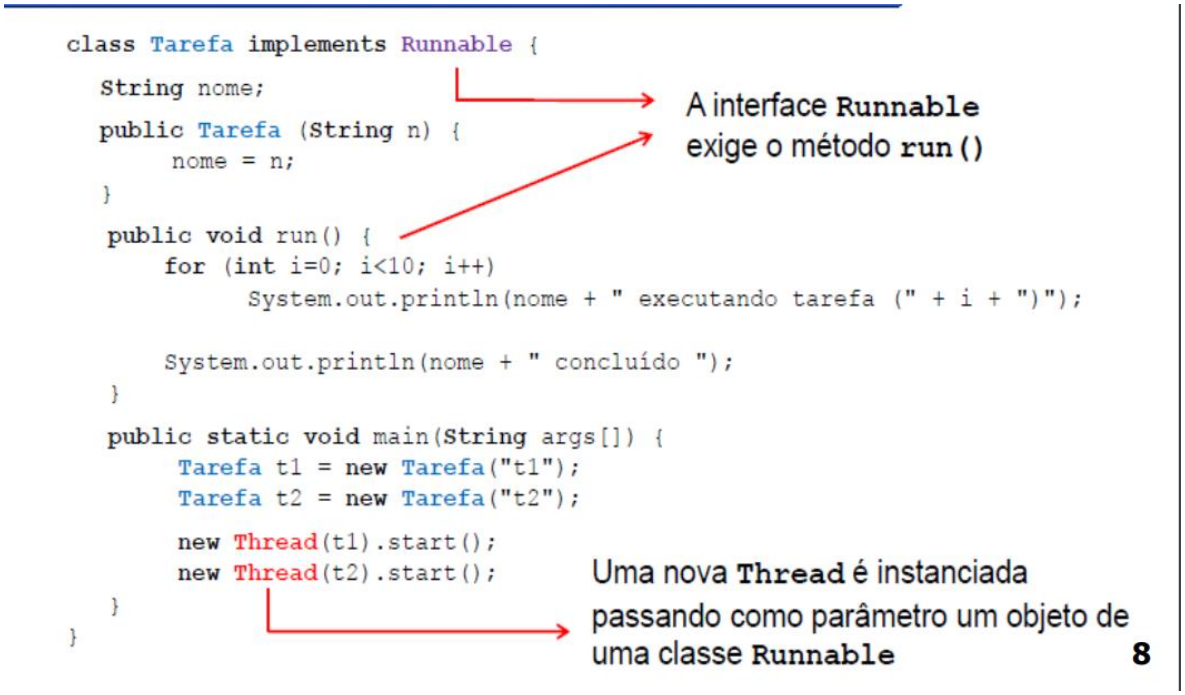
```
c1.start();
c2.start();
c3.start();
c4.start();
c5.start();
```

```
} catch (IOException e) {
    System.out.println(x:"fim");
}
```

```
ContaCargos c1 = new ContaCargos(path_arq, cargo:"Presidente");
ContaCargos c2 = new ContaCargos(path_arq, cargo:"Governador");
ContaCargos c3 = new ContaCargos(path_arq, cargo:"Deputado Federal");
ContaCargos c4 = new ContaCargos(path_arq, cargo:"Deputado Estadual");
ContaCargos c5 = new ContaCargos(path_arq, cargo:"Senador");
```

4) **Pode-se implementar threads em Java por herança ou pela interface *Runnable*. Quando usar uma solução ou a outra. Dê um exemplo de cada uma delas.**

Usando a interface Runnable permite que a lógica seja escrita em um método, sendo assim, esse código pode ser compartilhado entre várias threads, fazendo com que o seu uso seja muito mais amplo. Segue um exemplo:



Implementar uma thread em Java por herança apresenta um uso mais específico, visto que a herança é única, ou seja, uma classe irá herdar somente uma classe, impedindo que herde a thread e mais alguma outra classe. Segue um exemplo:

```
package Threads;

public class ThreadHeranca extends Thread {
public void run() {
        System.out.println("Alô herdei da classe thread!");
}

public static void main(String args[]) {
        (new ThreadHeranca()).start();
}
}
```

5) Explique a sincronização de threads em Java. Dê um exemplo.

A sincronização é um método que permite usar memória compartilhada entre as threads, utilizando de "locks" (permissões para que uma thread possa usar um recurso por vez), sendo obtidos pelo comando synchronized. Segue um exemplo:

```
public void teste() {
   synchronized(this) {
      façaAlgo();
   }
}
```

Sobre os códigos-fontes do laboratório (arquivo Lab_Threads.zip) respondam as seguintes questões.

6) Quais são os trechos de códigos referentes aos métodos construtores de cada uma das classes? O que eles constroem? Porque alguns atributos do construtor usam a palavra-chave *this* e outros não?

A classe Candidato tem o seguinte construtor:

```
Candidato (String nome, int numero, String partido) {
    this.nome = nome;
    this.numero = numero;
    this.partido = partido;
    qtde = 1;
}
```

Esse construtor usa do this para diferenciar os atributos “nome, numero e partido” do parâmetro nome, numero e partido. Já a qtde não usa this, visto que não há um parâmetro qtde, ou seja, não há a necessidade de diferencia-los.

A classe ContaCargos tem o seguinte método construtor:

```
ContaCargos(String arq, String cargo) {
    arquivo = arq;
    this.cargo = cargo;
}
```

Esse construtor inicializa o atributo arquivo sem o this, visto que o nome é único e não é parâmetro. Já o cargo é inicializado com this, visto que há um parâmetro “cargo”, havendo a necessidade de diferencia-los.

7) No código-fonte a estrutura de Dados ArrayList armazena quais dados e deque tipos. Qual a relação entre o ArrayList e o arquivo votos.txt?

A estrutura de dados ArrayList armazena objetos da classe Candidato, sendo eles: nome, numero, partido e qtde. O arquivo votos.txt é usado como banco de dados, contendo as informações dos votos para diferentes candidatos, caso a informação seja contida no ArrayList o programa soma na votação, ou seja, a relação é de comparação e, em seguida, há uma operação.

8) Para que é usado o controle de exceções em Java no caso específico do código-fonte do lab? Descreva um possível cenário onde o trecho de códigode exceções seria executado ao rodar o código-fonte?

O controle de exceções em Java, nesse caso, é usado para avisar ao usuário ou ao programador algum erro relacionado a leitura do arquivo, encerrando o programa, sendo assim, não irá permitir que rode com erros graves na leitura. Um cenário que a exceção seria executada é quando o arquivo não é encontrado, sendo executada a IOExcepetion, segue o trecho responsável por isso:

```
try {
    FileReader r = new FileReader(arquivo);
    BufferedReader bf = new BufferedReader(r);

    String linha;

    while ((linha = bf.readLine()) != null) {

        if (!linha.equals(anObject:"")) {
            info = linha.split(regex:";");

            if (info[0].equals(cargo)) {

                int i = 0;

                while (i < candidatos.size()) {
                    if (candidatos.get(i).achouCandidato(info[1])) {
                        candidatos.get(i).soma();
                        break;
                    }
                    i++;
                }

                // Se chegou ao final e n□o encontrou o                    candidato na lista
                if (i == candidatos.size()) {
                    if (!info[1].equals(anObject:"Branco") && !info[1].equals(anObject:"Anulado")) {
                        candidatos.add(new Candidato(info[1], Integer.parseInt(info[2]), info[3]));
                    } else
                        candidatos.add(new Candidato(info[1], Integer.parseInt(info[2]), partido:" "));
                }
            }
        }
    } // fim while de leitura de linha
```

9) Explique como as *threads* são usadas para ajudar a calcular o vencedor decada um dos cargos existentes no arquivo votos.txt?

O uso de threads permite que o código realize a contagem de votos de diferentes cargos em paralelo, aproveitando melhor os recursos de hardware e aumentando a eficiência do programa, especialmente ao lidar com grandes volumes de dados. Um exemplo do que foi falado pode ser observado na classe ContaCargos e main, onde a thread irá ser executada para processar de maneira individual cada uma das instâncias:

```
ContaCargos c1 = new ContaCargos(path_arq, cargo:"Presidente");
ContaCargos c2 = new ContaCargos(path_arq, cargo:"Governador");
ContaCargos c3 = new ContaCargos(path_arq, cargo:"Deputado Federal");
ContaCargos c4 = new ContaCargos(path_arq, cargo:"Deputado Estadual");
ContaCargos c5 = new ContaCargos(path_arq, cargo:"Senador");
```

10) Sobre o método *main* responda as seguintes questões:

a) Como o método main faz para ler o arquivo votos.txt e como faz paraimprimir aonde ele está armazenado?

O método main cria um objeto file com o caminho relativo (somado ao diretorio de trabalho), sendo assim, basta imprimi-lo.

b) Como o método main criar as *threads* e como ele as executa?

O método main cria as threads por meio de objetos (c1 até c5), representando um caso específico cada. Já a execução é feita inicializando cada um dos objetos por meio do start().

Criação:

```
ContaCargos c1 = new ContaCargos(path_arq, cargo:"Presidente");
ContaCargos c2 = new ContaCargos(path_arq, cargo:"Governador");
ContaCargos c3 = new ContaCargos(path_arq, cargo:"Deputado Federal");
ContaCargos c4 = new ContaCargos(path_arq, cargo:"Deputado Estadual");
ContaCargos c5 = new ContaCargos(path_arq, cargo:"Senador");
```

c)
d)

Execução:

```
c1.start();
c2.start();
c3.start();
c4.start();
c5.start();
```

c) Que modificações serão necessárias fazer no método *main* para ele calcular apenas os resultados referente ao voto de presidente?

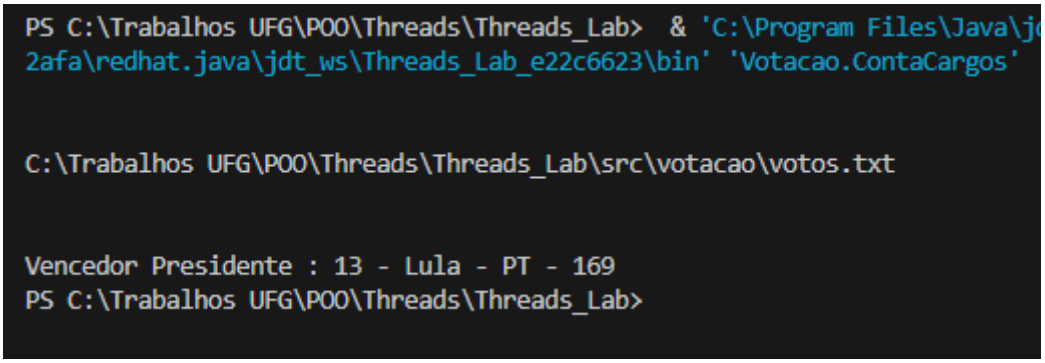
Uma modificação possível seria a de executar somente o c1. Segue o exemplo:

```
c1.start();
/*c2.start();
c3.start();
c4.start();
c5.start();*/
```

Transformando em comentário, o programa não irá executar o c2 em diante, sendo assim, somente os resultados referentes ao voto de presidente serão calculados.

d) Capture a tela de execução do método main.

Execução somente de presidente:



Execução completa: