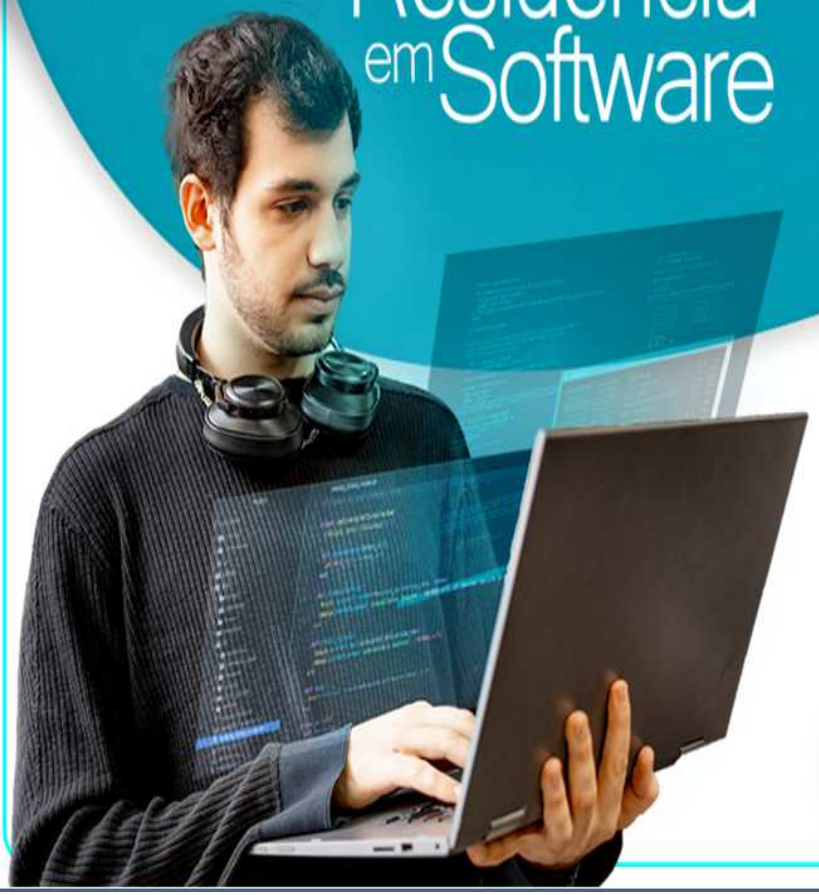
Residência em Tecnologia da Informação e Comunicação

Manipulando Arquivos Texto

Residência em Software



Professor:

Alvaro Degas Coelho

Java

INSTITUIÇÃO EXECUTORA

cepedi UESC

COORDENADORA

MCTI FUTURO Softex

APOIO

MINISTÉRIO DA CIÊNCIA, TECNOLOGIA E INOVAÇÃO

GOVERNO FEDERAL BRASIL UNIÃO E RECONSTRUÇÃO

Ler e Escrever Arquivos Texto

- Conceito geral: o mesmo que C
- Uma variável representa o arquivo
  - Ponteiro para um arquivo
- Este arquivo pode ser aberto
  - Usando o buffer da controladora de disco/usb/ssd etc
- Este arquivo pode ter leituras e escritas
- Este arquivo PRECISA ser fechado

java.io.FILE

- Representa um arquivo no sistema operacional
  - Representa → É um ponteiro
- Abrir um arquivo
  - File arquivo = new File("dados.txt");
  - File arquivo = new File("c:\aplicacaoX\dataFiles\dados.txt");

Após aberto o arquivo

- Será que ele existe no Sistema?
  - boolean existe = arquivo.exists();
- Se não existir, podemos criar o arquivo ou o diretório (pasta)
  - arquivo.createNewFile();
  - arquivo.mkdir();
- Um diretório possui arquivos (que também podem ser diretórios). Podemos saber quais
  - File [] arquivos = arquivo.listFiles();
- Podemos removê-lo do sistema
  - arquivo.delete();

FileWriter e BufferedWriter

- Classes cujos objetos escrevem em arquivos de texto
- FilteWriter acessa diretamente
- BufferedWriter escreve usa um buffer de transferência para sincronizar com a controladora de disco (através de um FW)
  - Desempenho melhorado
  - Mais independência do SO

FileWriter e BufferedWriter

- Construtor de FileWriter recebe o arquivo como parâmetro
  - FileWriter fw = new FileWriter(arquivo);
- Opcionalmente pode-se passar um parâmetro indicando que será aberto para acrescentar dados (append)
  - FileWriter fw = new FileWriter(arquivo, true);

FileWriter e BufferedWriter

- Tendo o FW pronto, podemos instanciar nosso BW
  - BufferedWriter bw = new BufferedWriter(fw);
- E usar o BW para escrever dados
  - bw.write("Texto a ser escrito no txt");
- Ou para acrescentar novas linhas
  - bw.newLine();

Fechar os aqruivos

- Tanto o BufferedWriter quanto o FileWriter devem ser fechados
- Isto garante que o SO comandará a controladora para salvar os dados no dispositivos
  - bw.close();
  - fw.close();

Exercício

- Faça um programa que pergunte ao usuário o nome de um arquivo
  - Crie o arquivo
- Depois peça ao usuário para digitar uma linha de texto
  - Crie a linha digitada dentro do arquivo
- Repita a criação de linhas até que o usuário digite uma linha vazia (string sem dado nenhum)
- Encerre o programa
- Vá ao diretório de seu programa e abra o arquivo que foi criado para certificar que está tudo certo.

java.io.FileReader e java.io.BufferedReader

- Para leitura de arquivos em texto
- FileReader recebe o arquivo e permite que ele seja acessível (através da controladora)
  - FileReader fr = new FileReader(arquivo);
- BufferedReader é o objeto que recebe os dados do buffer e retorna para o programa
  - BufferedReader br = new BufferedReader(fr);

Lendo os dados

- Método ready()
  - Será que ainda há mais dados para serem lidos?
- Método readline()
  - Traz uma nova linha do arquivo de texto
- Pequeno exemplo típico

```
while( br.ready() ){
        String linha = br.readLine();
        //faz algo com a linha
}
```

Fechar os recursos

- Fechar o BufferedReader
  - br.close();
- Fechar o FileReader
  - fr.close();

Exercício

- Faça um programa que solicite ao usuário o nome de um arquivo.
- Abra este arquivo e mostre na tela todos os seus conteúdos, linha a linha

Arquivos .csv

- Há estratégias para formatos de arquivos para transferência e/ou armazenamento de dados de aplicações
- Ideia geral: separar os tokens por algum delimitador bem conhecido
  - Token? → um pedaço coerente de dado
- Exemplo: supondo que o delimitador seja o caracter “;” - uma linha de arquivo de dados
  - Jose Frigobar;Rua left, casa 13;234234

Splitters

- Método split() da classe String
- Gera um array de Strings separando-os por um dado caracter
- Exemplo

```
String string = “Antonio da Silva;Professor”
String[] splitted= string.split(“-”);
String splitted1= splitted[0];
String splitted2 = splitted[1];
```

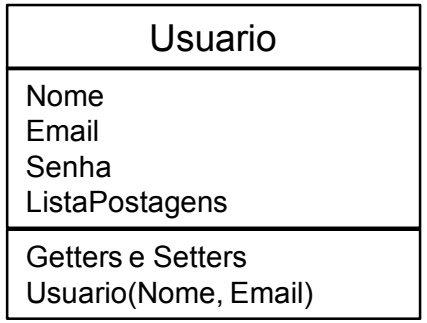
Para agora e Para casa

- Para agora
  - Crie um CRUD de Usuário (conforme no modelo)
    - Crie uma classe Usuario conforme o modelo ao lado
    - Crie uma classe Usuarios que contem uma lista de Usuario
- Crie métodos para inserir, buscar, listar todos e excluir usuários
- Crie um Menu com todas essas opções
- Altere o método de inserir para que ele abra um arquivo e salve os dados só usuário separando-os por ; (arquivo .csv)

Usuario
Nome Email Senha ListaPostagens
Getters e Setters Usuario(Nome, Email)

Para agora e Para casa

- Para casa
- Altere seu programa de forma que ele recupere todos os dados de usuários assim que o programa é iniciado
- Altere o método de excluir para que ele recrie o arquivo apenas com os dados dos usuários que não foram excluídos



Para agora e Para casa

- Para agora
  - Observe as classes que já definimos nos casos de uso
    - Construa o diagrama de classes que as inclua até o momento
- Para casa
  - Prossiga com os demais casos de uso do sistema do Tribunal
- Para se divertir: estude mais sobre casos de uso, descrição de casos de uso, diagramas de clasos de uso e, Forte recomendação...
  - DIAGRAMAS DE SEQUÊNCIA

Para agora e Para casa

- Para agora
  - Observe as classes que já definimos nos casos de uso
    - Construa o diagrama de classes que as inclua até o momento
- Para casa
  - Prossiga com os demais casos de uso do sistema do Tribunal
- Para se divertir: estude mais sobre casos de uso, descrição de casos de uso, diagramas de clasos de uso e, Forte recomendação...
  - DIAGRAMAS DE SEQUÊNCIA