

Lista - Concorrência, Arquivos e C

From IF669

Contents

- 1 Exercitando Arquivos
 - 1.1 Primeiro Exercício - Armazenando Contas em Arquivos
 - 1.2 Segundo Exercício - JNotepad (Opcional)
- 2 Exercitando Concorrência
 - 2.1 Primeiro Exercício - Identificando Problemas (Opcional)
 - 2.2 Segundo Exercício - Implementando com Concorrência
- 3 Exercitando C
 - 3.1 Primeiro Exercício - Dividindo com loops
 - 3.2 Segundo Exercício - Lendo e Armazenando Strings
 - 3.3 Terceiro Exercício - Manipulando structs

Exercitando Arquivos

Primeiro Exercício - Armazenando Contas em Arquivos

Primeira Parte - Armazenando Contas

Crie um método main que cria 5 contas, e as armazena em um arquivo objeto (utilize `JavaFile` (<http://www.cin.ufpe.br/~if669/files/miniJava/1.3/doc/br/ufpe/cin/miniJava/io/JavaFile.html>) para este fim). Para armazenamento destas contas, você pode utilizar a mesma classe `Conta` da lista anterior.

Segunda Parte - Resgatando Contas

Crie outro método main, que desta vez faz a leitura de um arquivo objeto e lê todas as contas do mesmo. Imprima as informações das contas na tela utilizando `MiniJavaSystem`.

Terceira Parte - Criando Relatórios

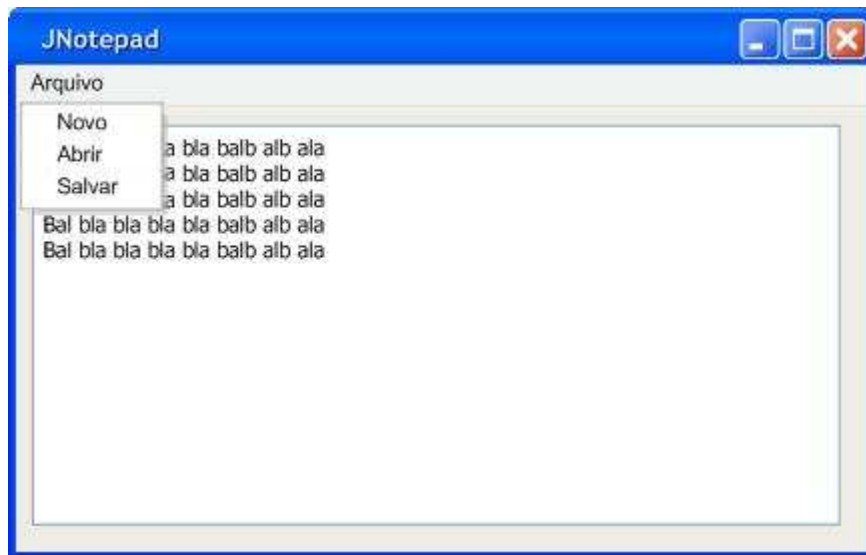
Modifique o método main da questão anterior para que, além da leitura dos arquivos e impressão na tela, ele crie um arquivo texto contendo um curto relatório sobre estas contas:

- Dados sobre a conta com maior saldo;
- Dados sobre a conta com menor saldo;
- A soma do saldo de todas as contas;

Utilize a classe TextFile (<http://www.cin.ufpe.br/~if669/files/miniJava/1.3/doc/br/ufpe/cin/miniJava/io/TextFile.html>) para a criação deste relatório.

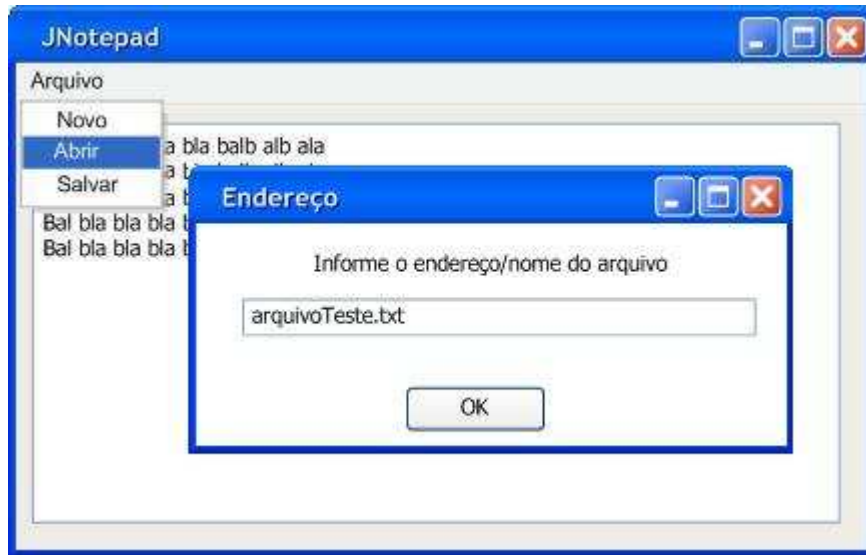
Segundo Exercício - JNotepad (Opcional)

Crie uma versão simplificada do Bloco de Notas, utilizando TextFile e recursos de MiniJava para interface gráfica.



Esta versão deve prover as funcionalidades:

- **Novo** : limpa o campo de texto para um novo texto.
- **Abrir** : abre um arquivo (de um endereço a ser especificado pelo usuário, como ilustrado na figura abaixo) e carrega seu conteúdo no campo de texto. Caso o arquivo não exista, uma mensagem de erro deve ser mostrada ao usuário.
- **Salvar**: salva o arquivo em um endereço especificado pelo usuário.



Exercitando Concorrência

Primeiro Exercício - Identificando Problemas (Opcional)

Veja o código abaixo:

```
class RepositorioContasArray {  
  
    private Conta[] contas;  
    private int proxima;  
  
    RepositorioContasArray(int tamanho) {  
        contas = new Conta[tamanho];  
        proxima = 0;  
    }  
  
    void inserir(Conta conta) {  
        contas[proxima] = conta;  
        proxima = proxima + 1;  
    }  
  
    Conta procurar(String n) {
```

```

        int indice = this.procurarIndice(n);

        Conta resultado = null;
        if (indice != proxima)
            resultado = contas[indice];
        else
            resultado = null;
        return resultado;
    }

    private int procurarIndice(String n) {
        int i = 0;
        if (n != null) {
            boolean achou = false;
            while ((!achou) && (i < proxima)) {
                if (n.equals(contas[i].getNumero()))
                    achou = true;
                else
                    i = i + 1;
            }
        } else
            i = proxima;
        return i;
    }
}

```

O que pode acontecer de errado caso o método `inserir` acima seja executado concorrentemente ao método `procurar`? Em que situações este erro ocorre (dica: imagine que a inserção só é feita depois da chamada ao `procurarIndice`)? O que deveria ser modificado nestes métodos para evitar o erro? **Descreva de forma clara.**

Segundo Exercício - Implementando com Concorrência

Altere os métodos `procurar` e `procurarIndice` da questão acima para que a procura no array seja feita de forma concorrente, isto é, procura-se ao mesmo tempo em cada metade do array. Um *thread* procura na primeira metade do array e outro na segunda metade; cada *thread* deve ter como atributo o resultado da procura, para que este valor depois seja inspecionado pelo método `procurar`. Será necessário definir métodos `protected` para ler os atributos do repositório. Esta é uma boa forma de aumentar a performance do método `procurar`? **Justifique sua resposta.**

Exercitando C

Primeiro Exercício - Dividindo com loops

Defina uma função `main` em C que lê dois números inteiros do teclado e imprime o resultado da divisão inteira do primeiro número pelo segundo. Não use o operador

de divisão '/', mas sim o operador de subtração '-', considerando que '9/4', por exemplo, é equivalente ao número de vezes que podemos subtrair 4 de 9 sem que o resultado seja negativo. Considere primeiro que os números lidos não podem ser negativos e depois ajuste o programa para que ele funcione corretamente com entradas negativas. Dê uma mensagem de erro caso o segundo número seja 0; que resultado seria gerado caso o seu programa não verificasse esta situação?

Segundo Exercício - Lendo e Armazenando Strings

Defina uma função main em C que lê strings do teclado até que seja lida uma string igual a uma das quinze últimas digitadas. Quando isto ocorre, o programa imprime o número de strings lidas, o tamanho da maior string lida, e a concatenação de todas as strings lidas exceto a última. Use um array para armazenar as quinze últimas strings lidas. Use também funções auxiliares para facilitar o entendimento do código. Veja a biblioteca de C para manipulação de strings: string.h (<http://www.cplusplus.com/ref/cstring/>).

Terceiro Exercício - Manipulando structs

Altere o programa conta.c (<http://www.cin.ufpe.br/~if669/files/MaterialDeEnsino/ListaConcorrenCIArquivosC/conta.c>) para que todo registro do tipo Conta contenha também uma referência para um registro do tipo Pessoa (veja pessoa.c (<http://www.cin.ufpe.br/~if669/files/MaterialDeEnsino/ListaConcorrenCIArquivosC/pessoa.c>)). Altere todas as funções, inclusive a main, para que elas agora considerem esta nova informação associada às contas. Em particular, a função transferir deve imprimir uma mensagem caso a transferência seja entre contas da mesma pessoa. Teste este novo comportamento na função main.

Retrieved from "http://www.cin.ufpe.br/~if669/index.php/Lista_-_Concorr%C3%Aancia%2C_Arquivos_e_C"

- This page was last modified 9 Mar 2007 da 12:38.
- This page has been accessed 840 times.
- Content is available under Attribution-NonCommercial-NoDerivs 2.5.
- Privacy policy
- About IF669
- Disclaimers