Algoritmo II

Arquivos

Prof. Me. Rober Marcone Rosi
Unidade de Engenharia, Computação e Sistemas

O que é um arquivo?

- Um **arquivo** é um local reservado para se guardar informações escritas para uso futuro. Um bom exemplo são os arquivos de aço, com pastas e documentos.
- O arquivo de computador é uma maneira de armazenar informações em meios físicos, magnéticos ou ópticos como, por exemplo, discos rígidos, discos flexíveis, *pen drives*, CDs e etc.

Arquivo-texto

- Os arquivos-texto são organizados em registros e campos.

Nome: _____

Endereço: _____

Cep: _____ Tel.: _____

	Campos			
Registros	**Nome**	**Endereço**	**Cep**	**Tel**
	João Ninguém	Rua do Bosque, 10	08000102	67867766
	Maria Bonita	Rua da Bruta, 247	09009904	31237788
	José Filho Jr.	Av. Sul, 3196	07989001	78966998

Formas de acesso

- O Java vê cada arquivo como um fluxo sequencial de bytes.
- O sistema operacional fornece um mecanismo para determinar o final do arquivo:
  - Como um marcador de fim do arquivo ou uma contagem do total de bytes no arquivo que é registrado nos dados mantidos na estrutura do sistema administrativo.
  - Um programa Java que processa um fluxo de bytes recebe uma indicação do sistema operacional sobre quando o programa alcança o final do fluxo.

Formas de acesso

- Fluxos de arquivos:
  - Fluxos baseados em bytes – representam dados no formato binário.
    - Arquivos binários – criados a partir de fluxos baseados em bytes, lidos por um programa que converte os dados em formato legível por humanos.
  - Fluxos baseados em caracteres – armazenam os dados como uma seqüência de caracteres.
    - Arquivos de texto – criados a partir de fluxos baseados em caracteres, eles podem ser lidos por editores de textos.
- O Java abre o arquivo criando um objeto e associando um fluxo a ele.

Formas de acesso

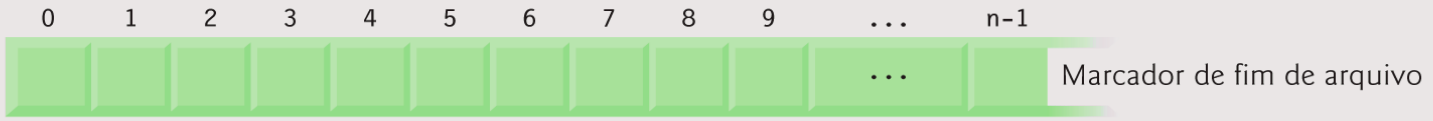
- Fluxos-padrão – cada fluxo pode ser redirecionado:
  - System.in – objeto do fluxo de entrada-padrão, ele pode ser redirecionado com o método setIn.
  - System.out – objeto do fluxo de saída-padrão, ele pode ser redirecionado com o método setOut.
  - System.err – objeto do fluxo de erro-padrão, ele pode ser redirecionado com o método setErr.

Formas de acesso

- Classes java.io:
  - FileInputStream e FileOutputStream – E/S baseada em bytes.
  - FileReader e FileWriter – E/S baseada em caracteres.
  - ObjectInputStream e ObjectOutputStream – os objetos dessas classes podem ser utilizados para E/S de objetos ou variáveis de tipos de dados primitivos.
  - File – útil para obter informações sobre arquivos e diretórios.

Formas de acesso

- Classes Scanner e Formatter
  - Scanner – pode ser utilizada para ler facilmente os dados em um arquivo.
  - Formatter – pode ser utilizada para gravar facilmente dados em um arquivo.



Visualização do Java de um arquivo de *nbytes*.

Formas de acesso

- Arquivos seqüenciais:
  - armazenam informações em caracteres no formato ASCII e os dados são gravados na ordem em que são digitados. As informações são lidas na mesma ordem em que foram inseridas, isto é, em seqüência;
- Arquivos de acesso aleatório ou randômico:
  - cada registro é gravado em uma posição específica. As informações podem ser lidas independentemente da ordem em que foram inseridas.

Operações de manipulação de arquivos

- Para a manipulação dos arquivos, existem quatro operações básicas que podem ser realizadas:

Inserção de dados: inclusão de novos registros;	**C**
Consulta aos dados: operação de leitura ou busca dos dados já armazenados;	**R**
Alteração dos dados: trata-se da possibilidade de alteração de um ou mais campos do conjunto;	**U**
Exclusão de dados: corresponde à operação de eliminação de registros.	**D**

***Create, Retrieve, Update e Delete**

A Classe File

- Classe File – útil para recuperar informações sobre arquivos e diretórios no disco: nome, diretório, tamanho, permissões de escrita e leitura e outras informações.
- A classe File é ainda uma representação para arquivos e diretórios de sistema, trazendo informações adicionais sobre o sistema operacional, tais como qual caractere é separador de diretório, informações sobre discos disponíveis, etc.
- Os objetos da classe File não abrem arquivos nem fornecem capacidades de processamento de arquivos.

A Classe File

- Em Java, a classe File permite representar arquivos nesse nível de abstração. Um dos construtores desta classe recebe como argumento uma string que pode identificar, por exemplo, o nome de um arquivo em disco.
- Os métodos desta classe permitem obter informações sobre o arquivo. Por exemplo: exists(), canRead(), canWrite(), length() e lastModified(); e realizar operações sobre o arquivo como um todo, como em delete() e deleteOnExit().

A Classe File

- A classe File fornece quatro construtores:

1. Recebe String que especifica nome e caminho (localização do arquivo no disco).
2. Recebe duas Strings: a primeira especificando o caminho e a segunda especificando o nome do arquivo.
3. Recebe o objeto File que especifica o caminho e String que especifica o nome do arquivo.
4. Recebe o objeto URI que especifica o nome e a localização do arquivo.

A Classe File

Método	Descrição
`boolean canRead()`	Retorna `true` se um arquivo for legível pelo aplicativo atual.
`boolean canWrite()`	Retorna `true` se um arquivo for gravável pelo aplicativo atual.
`boolean exists()`	Retorna `true` se o nome especificado como o argumento para o construtor File for um arquivo ou diretório no caminho especificado.
`boolean isFile()`	Retorna `true` se o nome especificado como o argumento para o construtor `File` for um arquivo.
`boolean isDirectory()`	Retorna `true` se o nome especificado como o argumento para o construtor `File` for um diretório.
`boolean isAbsolute()`	Retorna `true` se os argumentos especificados para o construtor `File` indicarem um caminho absoluto para um arquivo ou diretório.

A Classe File

Método	Descrição
`String getAbsolutePath()`	Retorna uma string com o caminho absoluto do arquivo ou diretório.
`String getName()`	Retorna uma string com o nome do arquivo ou diretório.
`String getPath()`	Retorna uma string com o caminho do arquivo ou diretório.
`String getParent()`	Retorna uma string com o diretório-pai do arquivo ou diretório (isto é, o diretório em que o arquivo ou diretório pode ser localizado).
`long length()`	Retorna o comprimento do arquivo, em bytes. Se o objeto `File` representar um diretório, 0 é retornado.
`long lastModified()`	Retorna uma representação dependente de plataforma da data/hora em que o arquivo ou diretório for modificado pela última vez. O valor retornado é útil somente para comparação com outros valores retornados por esse método.
`String[] list()`	Retorna um array de strings que representam o conteúdo de um diretório. Retorna `null` se o objeto `File` não representar um diretório.

Métodos File comuns

- exists **– retorna true se o arquivo existir onde especificado.**
- isFile **– retorna true se File for um arquivo, não um diretório.**
- isDirectory **– retorna true se File for um diretório.**
- getPath **– retorna o caminho de arquivo como uma string.**
- list **– recupera o conteúdo de um diretório.**

Dica

- O método File utiliza isFile para determinar se um objeto File representa um arquivo (não um diretório) antes de tentar abrir o arquivo.
- Utilizar \como um separador de diretório em vez de \\em uma literal de string é um erro de lógica.Uma \simples indica que a \seguida pelo próximo caractere representa uma sequencia de escape.Utilize \\para inserir uma \ em uma literal de string.

Arquivos de texto de acesso sequencial

- Os registros são armazenados na ordem por campo de chave de registro.
- Podem ser criados como arquivos de texto ou arquivos binários.
- O Java não impõe nenhuma estrutura a um arquivo; registros não existem como parte da linguagem Java.
- O programador deve estruturar os arquivos.

Criando um arquivo de texto de acesso sequencial

- A classe Formatter pode ser utilizada para abrir um arquivo de texto para gravar:
  - Passa o nome de arquivo para o construtor.
  - Se o arquivo não existir, ele será criado.
  - Se o arquivo já existir, o conteúdo será truncado (descartado).
  - Utiliza o método format para gravar texto formatado no arquivo.
  - Utiliza o método close para fechar o objeto Formatter (se esse método não for chamado, o SO normalmente fecha o arquivo quando o programa é fechado).

Criando um arquivo de texto de acesso sequencial

FAESA
faça acontecer

- Possíveis exceções:
  - SecurityException – ocorre ao abrir o arquivo utilizando o objeto Formatter, se o usuário não tiver permissão para gravar dados no arquivo.
  - FileNotFoundException – ocorre ao abrir o arquivo utilizando o objeto Formatter, se o arquivo não puder ser localizado e um novo arquivo não puder ser criado.
  - NoSuchElementException – ocorre quando uma entrada inválida é lida por um objeto Scanner.
  - FormatterClosedException – ocorre quando é feita uma tentativa de gravar em um arquivo utilizando um objeto Formatter já fechado.

Lendo dados a partir de um arquivo de texto de acesso sequencial

- Os dados são armazenados em arquivos de modo que eles possam ser recuperados para processamento quando necessário.
- O objeto **Scanner** pode ser utilizado para ler dados sequencialmente em um arquivo de texto:
  - Passa o objeto **File**, que representa o arquivo a ser lido, para o construtor **Scanner**.
  - FileNotFoundException ocorre se o arquivo não puder ser localizado.
  - Os dados são lidos no arquivo utilizando os mesmos métodos como entrada de teclado: nextInt, nextDouble, next, etc.
  - IllegalStateException ocorre se for feita uma tentativa de ler um objeto Scanner fechado.