

ARQUIVOS E FLUXOS

ACESSO SEQUENCIAL E ALEATÓRIO



5/10/2007

Arquivos e Fluxos
Prof. Ademair Schmitz, M.Sc.

1



BIBLIOGRAFIA



- HORSTMANN, Cay. **Conceitos de Computação com o Essencial de Java**. 3ª. Edição. São Paulo: Bookman, 2003.
 - Capítulo 15: Fluxos
- DEITEL, H. M., DEITEL, P.J. **Java – Como Programar**. 6ª ed. São Paulo: Pearson, 2005.
 - Capítulo 14: Arquivos e Fluxos

5/10/2007

Arquivos e Fluxos
Prof. Ademair Schmitz, M.Sc.

2



ARQUIVOS



- Um **arquivo** é um local reservado para se guardar informações escritas.
- As informações em um arquivo podem ser organizadas em **registros**.
- Os registros, por sua vez, são organizados em **campos** e nos campos são inseridas as **informações**.

5/10/2007

Arquivos e Fluxos
Prof. Ademair Schmitz, M.Sc.

3



ARQUIVOS



- Inserção de dados:
 - Trata-se da inclusão de novos registros.
- Consulta de dados:
 - Trata-se da leitura de dados já armazenados.
- Alteração de dados:
 - Trata-se da alteração de um ou mais campos do conjunto.
- Exclusão de dados:
 - Trata-se na eliminação de registros.

5/10/2007

Arquivos e Fluxos
Prof. Ademair Schmitz, M.Sc.

4



ARQUIVOS



- Arquivos podem ser diferenciados quando a forma de acesso aos dados:
 - Arquivos de acesso seqüencial.
 - Arquivos de acesso aleatório (randômico).

5/10/2007

Arquivos e Fluxos
Prof. Ademair Schmitz, M.Sc.

5



ACESSO SEQUENCIAL




- Informações são lidas na mesma ordem em que são gravadas.
- Um arquivo é processado um byte por vez.

5/10/2007

Arquivos e Fluxos
Prof. Ademair Schmitz, M.Sc.


6



ACESSO SEQUENCIAL

- Os fluxos ***System.in*** e ***System.out***, que são anexados ao teclado e à janela do terminal, permitem apenas acesso sequencial.


5/10/2007 Arquivos e Fluxos Prof. Ademar Schmitz, M.Sc. 7



ACESSO SEQUENCIAL

- Apresenta alguns inconvenientes:
 - É necessário que o arquivo inteiro seja percorrido até que a informação seja localizada.
 - As informações não podem ser alteradas diretamente no arquivo.
 - Não são recomendados para trabalhos com grande volume de informações.
 - Não é possível abri-lo para leitura e escrita.


5/10/2007 Arquivos e Fluxos Prof. Ademar Schmitz, M.Sc. 8



ACESSO ALEATÓRIO

- Cada registro é armazenado em um posição específica do arquivo.
- As informações podem ser lidas independentemente da ordem em que foram inseridas.
- Permite acesso a posições arbitrárias no arquivo, sem primeiro ler os *bytes* que precedem a posição em questão.


5/10/2007 Arquivos e Fluxos Prof. Ademar Schmitz, M.Sc. 9



ACESSO ALEATÓRIO

- Somente arquivos em disco suportam acesso aleatório.
- Cada arquivo de disco tem uma posição *file pointer* (ponteiro).


5/10/2007 Arquivos e Fluxos Prof. Ademar Schmitz, M.Sc. 10



ACESSO ALEATÓRIO

- Normalmente, o ponteiro de arquivo está no fim do arquivo, e qualquer saída é anexada ao final.
- Se movermos o ponteiro de arquivo para o meio de arquivo e gravarmos no arquivo, a saída sobrescreve o que está lá.


5/10/2007 Arquivos e Fluxos Prof. Ademar Schmitz, M.Sc. 11



ACESSO ALEATÓRIO

- O comando ***read()*** começa a ler a entrada na posição do ponteiro do arquivo.
- Podemos mover o ponteiro para o ponto exatamente após o último byte atualmente no arquivo, não mais adiante.


5/10/2007 Arquivos e Fluxos Prof. Ademar Schmitz, M.Sc. 12



ACESSO ALEATÓRIO

- Um **ponteiro de arquivo** é uma posição em um arquivo aleatório.
- Como os arquivos podem ser muito grandes, o ponteiro de arquivo é do tipo **long**.


5/10/2007 Arquivos e Fluxos Prof. Ademar Schmitz, M.Sc. 13



ACESSO ALEATÓRIO

- Para podermos atualizar um arquivo, temos de dar a cada campo um tamanho fixo que seja grande o suficiente.
- Como resultado, cada registro do arquivo terá o mesmo tamanho.

5/10/2007 Arquivos e Fluxos Prof. Ademar Schmitz, M.Sc. 14



ACESSO ALEATÓRIO

- Em Java, para ter acesso aleatório a um arquivo, utilizamos a classe **RandomAccessFile**.

5/10/2007 Arquivos e Fluxos Prof. Ademar Schmitz, M.Sc. 15