

## Material de Apoio Aula 4

### Arquivo com acesso randômico

Em algumas aplicações é necessário abrir um arquivo de dados para permitir operações de leitura e escrita em posições determinadas pelas aplicações. Isto é possível através da classe `RandomAccessFile` da API do Java. Um exemplo de uso desta classe está apresentado na Figura 1. Neste caso, o objeto que manipula o arquivo tem associado um apontador para a posição corrente que deve ser tratada no arquivo. Cada acesso, em leitura e/ou escrita faz com que este apontador passe a referenciar a posição que corresponde ao último byte manipulado pela referida operação.

```
import java.io.*;

class ArquivoAcessoRandomico {
    public static void main( String args[] ) throws java.io.IOException {
        RandomAccessFile raf = new RandomAccessFile("Raf.txt", "rw");

        raf.writeBytes ("Escrevendo a primeira linha.\n");
        raf.writeBytes ("Escrevendo a segunda linha.\n");
        raf.seek(0);
        String s = raf.readLine ();
        System.out.println ("Primeira linha no texto: "+s);
        raf.seek( raf.length() ); // vai para o final do arquivo
        raf.writeBytes ("Escrevendo a ultima linha.\n");
    }
}
```

**Figura 1.** Acesso randômico a posições de arquivo pela classe `RandomAccessFile`.

Na Figura 1, é criado um objeto para manipular um arquivo com acesso randômico. Alguns métodos da classe `RandomAccessFile` encontram-se representados, como o construtor, que requer o nome do arquivo e as operações a serem efetuadas nele (no caso do exemplo, “rw”, leitura e escrita) e os métodos:

- `writeBytes`: permite a escrita de dados no arquivo. Também existem os métodos `writeInt`, `writeFloat`, ...
- `readLine`: permite a leitura de uma linha do arquivos. Também existem os métodos `readInt`, `readFloat`, ...
- `length`: retorna o tamanho (em bytes) do arquivo.
- `seek(int v)`: desloca o apontador do arquivo para a posição, em bytes, representada por `v`. Note que  $v < \text{length}$  e  $v \geq 0$ .

A documentação completa desta classe é apresentada em <http://java.sun.com/j2se/1.5.0/docs/api/>

### Exercícios

1. Execute o programa da Figura 1 e veja o resultado abrindo o arquivo “raf.txt” em um editor de texto.
2. Modifique o programa da Figura 1, incluindo as seguintes linhas no final do método `main(String args[])`:

```
raf.seek( 0 ); // vai para o final do arquivo
raf.writeBytes ("Ups, nova primeira linha.\n");
```

Verifique o resultado da execução no arquivo raf.txt e compare com o resultado obtido com a execução prévia.

3. Estude o código da Figura 2. Implemente-o em um programa e analise o resultado.
4. Implemente um programa completo, onde seja oferecido ao usuário funcionalidades de:
  - Incluir novos registros no arquivo. O usuário deve fornecer quantidade inicial de itens em estoque e valor unitário.
  - Vender itens. O usuário deve informar o código e o número de itens que foram vendidos.
  - Modificar valor unitário: O usuário deve informar o código e o novo valor de um produto.

```

import java.io.*;

class estoque {
    public static void main (String args []) throws java.io.IOException {
        final int tamanhoRegistro = 24; // 2 * 4 + 2 * 8
        RandomAccessFile raf = new RandomAccessFile("estoque.txt", "rw");

        //Escrevendo Registros
        for(int i = 0 ; i < 10 ; i++ ) {
            raf.writeInt(i);
            raf.writeInt(5);
            raf.writeDouble(3.50);
            raf.writeDouble(5*3.50);
        }

        //Lendo Registros
        System.out.println("Imprimindo todo arquivo de estoque");
        raf.seek(0);
        int codigo, quantidade;
        double valorunitario, valorpassivo;
        for( int i = 0 ; i < 10 ; i++ ) {
            codigo = raf.readInt();
            quantidade = raf.readInt();
            valorunitario = raf.readDouble();
            valorpassivo = raf.readDouble();
            System.out.println("Item: " + codigo + " " +
                "Estoque: " + quantidade + " " +
                "R$ (un): " + valorunitario + " " +
                "R$ (pas): " + valorpassivo);
        }

        // Iprimir dados do registro de codigo 4
        System.out.println("Imprimindo o registro 4");
        raf.seek( 5 * tamanhoRegistro );
        codigo = raf.readInt();
        quantidade = raf.readInt();
        valorunitario = raf.readDouble();
        valorpassivo = raf.readDouble();
        System.out.println("Item: " + codigo + " " +
            "Estoque: " + quantidade + " " +
            "R$ (un): " + valorunitario + " " +
            "R$ (pas): " + valorpassivo);

        // Modificando o registro 4
        System.out.println("Vendendo um produto no registro 4");
        raf.seek( 5 * tamanhoRegistro + 4 ); //caminhando ate a quantidade
        quantidade = raf.readInt(); //lendo a quantidade
        valorunitario = raf.readDouble(); //lendo o preco unitario
        quantidade = quantidade - 1;
        //agora o ponteiro do arquivo esta posicionado no valor passivo
        raf.writeDouble( quantidade*valorunitario ); //atualiza passivo
        raf.seek( 5 * tamanhoRegistro + 4 ); //caminhando ate a quantidade
        raf.writeInt( quantidade ); //atualiza quantidade

        System.out.println("Registro 4 apos a venda");
        raf.seek( 5 * tamanhoRegistro ); //posiciona no registro 4
        codigo = raf.readInt();
        quantidade = raf.readInt();
        valorunitario = raf.readDouble();
        valorpassivo = raf.readDouble();
        System.out.println("Item: " + codigo + " " +
            "Estoque: " + quantidade + " " +
            "R$ (un): " + valorunitario + " " +
            "R$ (pas): " + valorpassivo);
    }
}

```

**Figura 2.** Exemplo de manipulação de arquivo.