

Algoritmos II

Ricardo Ribeiro Assink
 ricardo.assink@unisul.br
 ricardo@equipedigital.com

<http://www.ricardoassink.com.br/>

Algoritmos II - UNIDADE 4, parte 2- Ricardo Ribeiro Assink - ricardo.assink@unisul.br

1

Algoritmos II

ARQUIVO BINÁRIO

Arquivos

O acesso a um **arquivo binário** dentro do algoritmo é feito da mesma maneira que a leitura e escrita de variáveis. No algoritmo, o arquivo deve ser declarado e aberto, antes que tal acesso possa ser feito a leitura ou escrita. No final do algoritmo, ou quando houver necessidade, o arquivo deve ser fechado.

A declaração de um arquivo binário é feita através da especificação a seguir:

```
Algoritmo DeclaraçãoArquivoBinario
Variáveis
    <Nome_arquivo> : Arquivo Binário de <Tipo_de_dado>
Início
Fim.
```

Onde:

<Nome_arquivo> é a referência ao arquivo binário dentro do algoritmo.

Arquivo Binário de são palavras reservadas

<Tipo_de_dado> define o tipo de dado a ser mantido no arquivo binário. O tipo de dado pode ser um tipo já existente (Inteiro, Real, Literal) ou um tipo definido pelo usuário (Registro).

Algoritmos II - UNIDADE 4, parte 2 - Ricardo Ribeiro Assink - ricardo.assink@unisul.br

2

Algoritmos II

Arquivos

ARQUIVO BINÁRIO

A **abertura de um arquivo binário** é feita através da especificação a seguir:

```

Algoritmo AberturaArquivoBinario
Variáveis
    <Nome_arquivo> : Arquivo Binario de <Tipo_de_dado>
Início

    abrirArquivo( <Nome_arquivo>, <Nome_arquivo_SO>, <Utilização>
    )
Fim.
  
```

Onde:

- <Nome_arquivo> é o arquivo que será aberto pelo comando abrir.
- abrirArquivo** é uma palavra reservada. Alguns autores pode utilizar a palavra Abra, OpenFile, etc...
- <Nome_arquivo_SO> é o nome do arquivo no sistema operacional.
- <Utilização> especifica se o arquivo será usado somente para leitura ou somente para escrita ou ambos (somente em arquivos binário). Se for para leitura e escrita não é necessário especificar a forma de utilização.

Algoritmos II - UNIDADE 4. parte 2 - Ricardo Ribeiro Assink - ricardo.assink@unisul.br 3

Algoritmos II

Arquivos

ARQUIVO BINÁRIO

O **fechamento de um arquivo binário** é feito através da especificação a seguir:

```

Algoritmo FechamentoArquivoBinario
Variáveis
    <Nome_arquivo> : Arquivo Texto
Início
    fecharArquivo (<Nome_arquivo>)
Fim.
  
```

Onde:

- <Nome_arquivo> é o arquivo que será fechando pelo comando fechar.
- fecharArquivo** é uma palavra reservada. Alguns autores pode utilizar a palavra Feche, CloseFile, etc..

Algoritmos II - UNIDADE 4. parte 2 - Ricardo Ribeiro Assink - ricardo.assink@unisul.br 4

Algoritmos II

Arquivos

Exemplo de Abertura e Fechamento de ARQUIVO BINÁRIO

```

Algoritmo AbrindoArquivoBinario
Constante
    NOME_ARQUIVO = "AGENDA.DAT"
Tipo
    Pessoa : Registro
        nome: Literal;
        altura: Real;
        idade: Inteiro;
        Fim_Registro
Variáveis
    agenda : Arquivo Binario de Pessoa
Início
    abrirArquivo(agenda,NOME_ARQUIVO)
    fecharArquivo(agenda)
Fim.

```

Algoritmos II

Arquivos

ARQUIVO BINÁRIO

A **escrita em um arquivo binário** é feita através da especificação a seguir:

```

Algoritmo EscrevendoArquivoBinario
Variáveis
    <Nome_arquivo> : Arquivo Binario de
    <Tipo_de_dado>
    <Nome_variável> : <Tipo_de_dado>
Início
    Escreva (<Nome_arquivo>,<Nome_variavel>)
Fim.

```

Onde:

Escreva é palavra reservada.

<Nome_arquivo> é o arquivo que será lido pelo comando escreva.

<Nome_variável> é o nome da variável que contem o valor a ser escrito no arquivo. A variável será do mesmo tipo de dado do arquivo a ser escrito.

Algoritmos II

Arquivos

Exemplo ESCRITA – ARQUIVO BINÁRIO

```

import java.io.*;
import javax.swing.*;

public class EscrevendoArquivoBinario {
    static final String NOME_ARQUIVO = "AGENDA.DAT";
    static class Pessoa {
        String nome;
        double altura;
        int idade;
    }
    public static RandomAccessFile abrirArquivo() {
        RandomAccessFile arquivo = null;
        try {
            arquivo = new RandomAccessFile(new File(NOME_ARQUIVO), "rw");
        }
        catch (IOException e) {
            JOptionPane.showMessageDialog(null, "Não foi possível abrir o
arquivo!");
        }
        return arquivo;
    }
}

```

CONTINUA ->

Algoritmos II - UNIDADE 4. parte 2 - Ricardo Ribeiro Assink - ricardo.assink@unisul.br

7

Algoritmos II

Arquivos

Exemplo ESCRITA – ARQUIVO BINÁRIO

```

    public static void fecharArquivo(RandomAccessFile _arquivo) {
        try {
            _arquivo.close();
        }
        catch (IOException e) {
            JOptionPane.showMessageDialog(null, "Não foi possível fechar o
arquivo!");
        }
    }
    public static void escreveArquivo(RandomAccessFile _arquivo, Pessoa _reg) {
        try {
            escrevePalavra(_arquivo, _reg.nome, 15);
            _arquivo.writeDouble(_reg.altura);
            _arquivo.writeInt(_reg.idade);
        }
        catch (IOException e){}
    }
}

```

CONTINUA ->

Algoritmos II - UNIDADE 4. parte 2 - Ricardo Ribeiro Assink - ricardo.assink@unisul.br

8

Algoritmos II

Arquivos

Exemplo ESCRITA – ARQUIVO BINÁRIO

```

private static void escrevePalavra(RandomAccessFile _arquivo, String _palavra,
int _quantidade) {
    try {
        StringBuffer buf = null;

        if (_palavra != null) {
            buf = new StringBuffer(_palavra);
        }
        else {
            buf = new StringBuffer(_quantidade);
        }

        buf.setLength(_quantidade);

        _arquivo.writeChars(buf.toString());

    } catch (IOException e){}
}

```

CONTINUA ->

Algoritmos II - UNIDADE 4. parte 2 - Ricardo Ribeiro Assink - ricardo.assink@unisul.br 9

Algoritmos II

Arquivos

Exemplo ESCRITA – ARQUIVO BINÁRIO

```

public static void main(String args[]) {

    RandomAccessFile agenda = abrirArquivo();
    Pessoa reg = new Pessoa();
    try {
        reg.nome = JOptionPane.showInputDialog("Digite um nome ou ENTER para sair");

        while (!reg.nome.equals("")) {
            reg.altura = Double.parseDouble(JOptionPane.showInputDialog("Digite a
Altura"));
            reg.idade = Integer.parseInt(JOptionPane.showInputDialog("Digite a Idade"));

            agenda.seek( agenda.length()); // posiciona gravação no final do arquivo

            escreveArquivo(agenda, reg);
            reg.nome = JOptionPane.showInputDialog("Digite um nome ou ENTER para sair");
        } // fim do while

    } catch (IOException e){
        System.out.println("Problemas no arquivo");
    } // fim do try / catch

    fecharArquivo(agenda);

} // fim do main
} // fim da classe

```

Algoritmos II - UNIDADE 4. parte 2 - Ricardo Ribeiro Assink - ricardo.assink@unisul.br 10

Algoritmos II

Arquivos

ARQUIVO BINÁRIO

A **leitura de um arquivo binário** é feito através da especificação a seguir:

```

Algoritmo EscrevendoArquivoBinario
Variáveis
    <Nome_arquivo> : Arquivo Binario de
    <Tipo_de_dado>
    <Nome_variavel> : <Tipo_de_dado>
Início
    Leia (<Nome_arquivo>, <Nome_variavel>)
Fim.
  
```

Onde:

Leia é palavra reservada.

<Nome_arquivo> é o arquivo binário que será lido pelo comando leia.

<Nome_variavel> é o nome da variável que irá conter o valor lido do arquivo binário. A variável será do mesmo tipo de dado do arquivo a ser lido.

Algoritmos II - UNIDADE 4, parte 2 - Ricardo Ribeiro Assink - ricardo.assink@unisul.br 11

Algoritmos II

Arquivos

Exemplo LEITURA – ARQUIVO BINÁRIO

```

import java.io.*;
import javax.swing.*;

public class Unidade4_04 {
    static final String NOME_ARQUIVO = "AGENDA.DAT";
    static class Pessoa {
        String nome;
        double altura;
        int idade;
    }
    public static RandomAccessFile abrirArquivo() {
        RandomAccessFile arquivo = null;
        try {
            arquivo = new RandomAccessFile(new File(NOME_ARQUIVO), "rw");
        }
        catch (IOException e) {
            JOptionPane.showMessageDialog(null, "Não foi possível abrir o
arquivo!");
        }
        return arquivo;
    }
}
  
```

CONTINUA ->

Algoritmos II - UNIDADE 4, parte 2 - Ricardo Ribeiro Assink - ricardo.assink@unisul.br 12

Algoritmos II

Arquivos

Exemplo LEITURA – ARQUIVO BINÁRIO

```

public static void fecharArquivo(RandomAccessFile _arquivo) {
    try {
        _arquivo.close();
    }
    catch (IOException e) {
        JOptionPane.showMessageDialog(null, "Não foi possível fechar o
arquivo!");
    }
}

public static void leituraArquivo(RandomAccessFile _arquivo, Pessoa
_reg) throws IOException {
    _reg.nome = montaPalavra(_arquivo,15);
    _reg.altura = _arquivo.readDouble();
    _reg.idade = _arquivo.readInt();
}

```

CONTINUA ->

Algoritmos II - UNIDADE 4, parte 2 - Ricardo Ribeiro Assink - ricardo.assink@unisul.br 13

Algoritmos II

Arquivos

Exemplo LEITURA – ARQUIVO BINÁRIO

```

private static String montaPalavra( RandomAccessFile _arquivo, int
_quantidade) {
    char palavra[] = new char[_quantidade];
    try {
        for(int i=0; i < palavra.length; i++) {
            palavra[i] = _arquivo.readChar();
        }
    } catch (IOException e){}
    return new String (palavra).replace('\0',' ');
}

```

CONTINUA ->

Algoritmos II - UNIDADE 4, parte 2 - Ricardo Ribeiro Assink - ricardo.assink@unisul.br 14

Algoritmos II

Arquivos

Exemplo LEITURA – ARQUIVO BINÁRIO

```

public static void main(String args[])    {

    RandomAccessFile agenda = abrirArquivo();

    Pessoa reg = new Pessoa();
    try {

        agenda.seek(0); // posiciona a leitura do arquivo no inicio

        while (agenda.getFilePointer() < agenda.length()) {
            leituraArquivo(agenda, reg);
            JOptionPane.showMessageDialog(null, "Nome: "+reg.nome +
                "\nAltura: "+reg.altura+
                "\nIdade: "+reg.idade);
        }
    }
    catch (IOException e){
        System.out.println("Problemas no arquivo");
    }
    fecharArquivo(agenda);

    } // fim do main
} // fim da class

```

15

Algoritmos II - UNIDADE 4. parte 2 - Ricardo Ribeiro Assink - ricardo.assink@unisul.br

Algoritmos II

Arquivos

FIM

Material Original: Osmar de Oliveira Braz Junior

Algoritmos II - UNIDADE 4. parte 2 - Ricardo Ribeiro Assink - ricardo.assink@unisul.br

16