

# Unidade 0 - Nivelamento - Arquivos em Java



**PUC Minas**

Instituto de Ciências Exatas e Informática  
Departamento de Ciência da Computação

# Definição de Arquivo

- Unidade lógica utilizada para armazenar dados em disco ou em qualquer outro dispositivo externo de armazenamento
- Pode-se abrir, fechar, ler, escrever ou apagar um arquivo

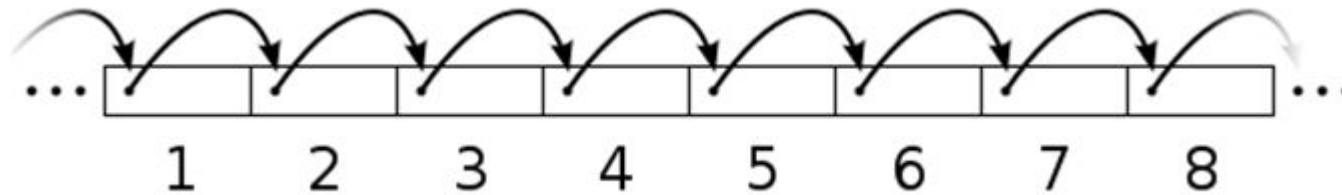


# Introdução

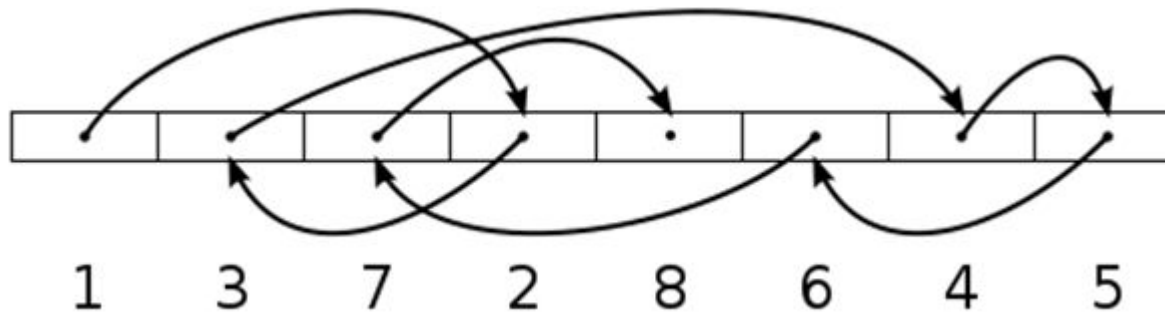
- O Java tem algumas classes para leitura e escrita em arquivo
- A classe `Arq` trabalha com arquivos em modo texto e a `RandomAccessFile`, em modo binário
- Este material usa a classe `Arq` para acesso sequencial e a `RandomAccessFile` para acesso aleatório

# Acessos Sequencial e Aleatório

- Acesso Sequencial



- Acesso Aleatório



## Classe Arq

- Utiliza as classes Formatter e Scanner para a escrita e leitura de arquivos de texto, respectivamente
- Os arquivos escritos com a Arq podem ser lidos com editores de texto e vice-versa

# Exemplo para Escrita com a Arq

```
//Arquivo ExemploArq01Escrita.java
class ExemploArq01Escrita {
    public static void main (String[] args){
        Arq.openWrite("exemplo.txt");

        Arq.println(1);
        Arq.println(5.3);
        Arq.println('X');
        Arq.println(true);
        Arq.println("Algoritmos");

        Arq.close();
    }
}
```

# Exemplo para Escrita com a Arq

```
//Arquivo ExemploArq01Escrita.java
class ExemploArq01Escrita {
    public static void main (String[] args){
        Arq.openWrite("exemplo.txt");

        Arq.println(1);
        Arq.println(5.3);
        Arq.println('X');
        Arq.println(true);
        Arq.println("Algoritmos");

        Arq.close();
    }
}
```

Este programa cria o arquivo exemplo.txt da seguinte forma:

```
1
5.3
X
true
Algoritmos
```

Para confirmar, abra o arquivo exemplo.txt com um editor de texto

# Exemplo para Leitura com a Arq

```
//Arquivo ExemploArq02Leitura.java
```

```
class ExemploArq02Leitura {  
    public static void main (String[] args){  
        Arq.openRead("exemplo.txt");  
  
        int inteiro = Arq.readInt();  
        double real = Arq.readDouble();  
        char caractere = Arq.readChar();  
        boolean boleano = Arq.readBoolean();  
        String str = Arq.readString();  
  
        Arq.close();  
  
        System.out.println("inteiro: " + inteiro);  
        System.out.println("real: " + real);  
        System.out.println("caractere: " + caractere);  
        System.out.println("boleano: " + boleano);  
        System.out.println("str: " + str);  
    }  
}
```

exemplo.txt

```
1  
5.3  
X  
true  
Algoritmos
```



# Exemplo para Leitura com a Arq

//Arquivo ExemploArq02Leitura.java

```
class ExemploArq02Leitura {  
    public static void main (String[] args){  
        Arq.openRead("exemplo.txt");  
  
        int inteiro = Arq.readInt();  
        double real = Arq.readDouble();  
        char caractere = Arq.readChar();  
        boolean boleano = Arq.readBoolean();  
        String str = Arq.readString();  
  
        Arq.close();  
  
        System.out.println("inteiro: " + inteiro);  
        System.out.println("real: " + real);  
        System.out.println("caractere: " + caractere);  
        System.out.println("boleano: " + boleano);  
        System.out.println("str: " + str);  
    }  
}
```

exemplo.txt

```
1  
5.3  
X  
true  
Algoritmos
```

Para confirmar, veja os valores mostrados na tela

- Execute e brinque com os arquivos

ExemploArq01Escrita.java e ExemploArq02Leitura.java

# Métodos da Arq

- boolean openWrite(String nomeArq)
- boolean openWrite(String nomeArq, String charset)
- boolean openRead(String nomeArq)
- boolean openRead(String nomeArq, String charset)
- void close()

Todos públicos e estáticos

- void print(int x)
- void print(double x)
- void print(String x)
- void print(boolean x)
- void print(char x)

# Exercício Resolvido

- Faça um programa que abre um arquivo e cria uma cópia

## Exercício Resolvido

- Faça um programa que abre um arquivo e cria uma cópia

```
class ExemploArq03Exercicio {  
    public static void main (String[] args){  
        Arq.openRead("exemplo.txt");  
  
        String str = "";  
        while (Arq.hasNext() == true){  
            str += Arq.readChar();  
        }  
  
        Arq.close();  
  
        Arq.openWrite("copia.txt");  
        Arq.print(str);  
        Arq.close();  
    }  
}
```

## Exercício Resolvido

- Faça um programa que abre um arquivo e cria uma cópia

O comando `Arq.openRead("exemplo.txt")` poderia ser substituído:

- 1) `Arq.openRead("C:\\entrada01.txt")`
- 2) `Arq.openRead("/home/maxm/entrada01.txt")`
- 3) `Arq.openRead(MyIO.readString("Digite o nome do arquivo: "))`

## Exercício Resolvido

- Faça um programa que abre um arquivo e cria uma cópia

Os comandos `String str = "";` poderiam ser substituídos por:

```
while (Arq.hasNext() == true){  
    str += Arq.readChar();  
}
```

1) `String str = "";`  
`while (Arq.hasNext() == true){`  
    `str += Arq.readString();`  
`}`

2) `String str = "";`  
`while (Arq.hasNext() == true){`  
    `str += Arq.readLine();`  
`}`

3) `String str = Arq.readAll();`

## Classe *Random Access File*

- Permite o acesso aleatório de leitura e escrita a arquivos
- Manipula arquivos em modo binário
- Considera que cada arquivo é um *array* de bytes indexado por um cabeçote (ponteiro cursor)



- Sua posição indica o local da próxima leitura ou escrita
- Após uma operação de leitura / escrita, ele “anda” no *array* o número de bytes lidos / escritos
- O método *long getFilePointer()* retorna sua posição corrente e o *void seek(long)* altera esse valor

# Os Dois Construtores da RAF

- Recebem como parâmetro uma referência para um arquivo e um modo de operação indicando leitura ou escrita
  - `RandomAccessFile(File file, String mode)`
  - `RandomAccessFile(String name, String mode)`

# Os Dois Construtores da RAF

- Recebem como parâmetro uma referência para um arquivo e um modo de operação indicando leitura ou escrita
  - `RandomAccessFile(File file, String mode)`
  - `RandomAccessFile(String name, String mode)`

"r", apenas leitura. A leitura de um arquivo inexistente causa uma `FileNotFoundException` e a chamada de um método de escrita causa uma `IOException`

"rw", leitura/escrita. Se o arquivo não existir, o mesmo é criado

"rws" e "rwd", leitura/escrita com sincronizações

Outras exceções: `IllegalArgumentException` e `SecurityException`

# Exemplo para Escrita com a RAF

```
//Arquivo ExemploRAF01Escrita.java
```

```
import java.io.*;
```

```
class ExemploRAF01Escrita {  
    public static void main (String[] args) throws Exception{
```

```
        RandomAccessFile raf = new RandomAccessFile("exemplo.txt", "rw");
```

```
        raf.writeInt(1);
```

```
        raf.writeDouble(5.3);
```

```
        raf.writeChar('X');
```

```
        raf.writeBoolean(true);
```

```
        raf.writeBytes("Algoritmos");
```

```
        raf.close();
```

```
    }
```

```
}
```

# Exemplo para Escrita com a RAF

//Arquivo ExemploRAF01Escrita.java

import java.io.\*;

class ExemploRAF01Escrita {  
 public static void main (String[] args) throws Exception{

RandomAccessFile raf = new RandomAccessFile("exemplo.txt", "rw");

raf.writeInt(1);  
 raf.writeDouble(5.3);  
 raf.writeChar('X');  
 raf.writeBoolean(true);  
 raf.writeBytes("Algoritmos");

raf.close();

}

}

Arquivo exemplo.txt aberto em  
um editor de texto:

@333333 XAlgoritmos

# Exemplo para Escrita com a RAF

```
//Arquivo ExemploRAF01Escrita.java
```

```
import java.io.*;
```

```
class ExemploRAF01Escrita {  
    public static void main (String[] args) throws Exception{
```

```
        RandomAccessFile raf = new RandomAccessFile("exemplo.txt", "rw");
```

```
        raf.writeInt(1);  
        raf.writeDouble(5.3);  
        raf.writeChar('X');  
        raf.writeBoolean(true);  
        raf.writeBytes("Algoritmos");
```

```
        raf.close();
```

```
    }
```

```
}
```

Os métodos da RAF podem gerar exceções e não são estáticos

# Exemplo para Leitura com a RAF

```
//Arquivo ExemploArq02Leitura.java
```

```
import java.io.*;
```

```
class ExemploArq02Leitura {
```

```
    public static void main (String[] args) throws Exception{
```

```
        RandomAccessFile raf = new RandomAccessFile("exemplo.txt", "rw");
```

```
        int inteiro = raf.readInt();
```

```
        double real = raf.readDouble();
```

```
        char caractere = raf.readChar();
```

```
        boolean boleano = raf.readBoolean();
```

```
        String str = raf.readLine();
```

```
        raf.close();
```

```
        System.out.println(inteiro + " " + real + " " + caractere + " " + boleano + " " + str);
```

```
    }
```

```
}
```

# Exemplo usando o Cabeçote da RAF

```
//Arquivo ExemploRAF03Cabecote.java
```

```
import java.io.*;
```

```
class ExemploRAF03Cabecote {  
    public static void main (String[] args) throws Exception{
```

```
        RandomAccessFile raf = new RandomAccessFile("exemplo.txt", "rw");
```

```
        raf.writeInt(1);
```

```
        raf.writeDouble(5.3);
```

```
        raf.writeChar('X');
```

```
        raf.writeBoolean(true);
```

```
        raf.writeBytes("Algoritmos");
```

```
        ...
```



# Exemplo usando o Cabeçote da RAF

...

```
raf.seek(0);           //Retornando o cabecote para a posicao 0  
System.out.println(raf.readInt()); //imprimindo o inteiro
```

```
raf.seek(12);          //Setando o cabecote na posicao 12 (do caractere,  
                      //12 = 4 do int + 8 do double)  
System.out.println(raf.readChar());
```

```
raf.seek(12);          //Setando o cabecote novamente na posicao 12  
raf.writeChar('@');    //Substituindo o caractere
```

```
raf.seek(12);  
System.out.println(raf.readChar());
```

```
raf.close();
```

```
}  
}
```

# Exercício

- Execute e brinque com os arquivos

ExemploRAF01Escrita.java, ExemploArq02Leitura.java e  
ExemploRAF03Cabecote.java

# Métodos da RandomAccessFile

- void close()
- FileChannel getChannel()
- FileDescriptor getFD()
- long length()
- void seek(long pos)
- void setLength(long newLength)
- int skipBytes(int n)

# Métodos da RandomAccessFile

- `int read()`
- `int read(byte[] b)`
- `int read(byte[] b, int off, int len)`
- `boolean readBoolean()`
- `byte readByte()`
- `char readChar()`
- `double readDouble()`
- `float readFloat()`
- `void readFully(byte[] b, int off, int len)`
- `void readFully(byte[] b)`

# Métodos da RandomAccessFile

- void write(byte[] b)
- void write(byte[] b, int off, int len)
- void write(int b)
- void writeBoolean(boolean v)
- void writeByte(int v)
- void writeBytes(String s)
- void writeChar(int v)
- void writeChars(String s)

## Exercício: Faça um Programa que

- Leia o nome de um arquivo e uma frase e salve essa frase nesse arquivo
- Leia o nome de um arquivo e mostre seu conteúdo na tela
- Leia o nome de um arquivo e mostre seu conteúdo convertido para letras maiúsculas
- Leia o nome de dois arquivos e copie o conteúdo do primeiro para o último

## Exercício: Faça um Programa que

- Leia o nome de dois arquivos, abra o primeiro, converta seu conteúdo para letra maiúscula e salve o no segundo
- Leia o nome de dois arquivos e copie o conteúdo do primeiro para o segundo invertendo a ordem dos caracteres. O último caractere no arquivo de entrada será o primeiro do de saída, o penúltimo caractere será o segundo, ...

## Exercício: Faça um Programa que

- Leia o nome de um arquivo e mostre na tela o conteúdo desse arquivo criptografado usando o ciframento de César ( $k = 3$ )
- Leia o nome de um arquivo contendo uma mensagem criptografada com o Ciframento de César ( $k = 3$ ), descriptografe a mensagem e mostre-a na tela



## Exercício

- Seja os arquivos Pilha.java e PrincipalPilha.java (fonte/tadEstatica), altere o segundo de tal forma que ele contenha um menu com as opções (1) Inserir, (2) Remover (3) Listar (4) Sair, permitindo inserções, remoções e listar os elementos de uma pilha. Quando executamos o PrincipalPilha, ele lê os elementos da pilha de um arquivo pilha.dat. Quando o usuário seleciona a opção (4) Sair, o programa salva os elementos da pilha no arquivo pilha.dat. Devemos manter a ordem dos elementos a cada nova execução do PrincipalPilha.