

Introdução à Programação

Orientada a Objetos com Java

Arquivos

Leonardo Cole e Paulo Borba
Centro de Informática
Universidade Federal de Pernambuco

Depois de aprender a construir uma aplicação ...

Precisamos aprender a
persistir
os dados desta aplicação



E precisamos persistir dados pois...

Os dados de uma aplicação devem estar disponíveis para uso e modificação mesmo depois da aplicação ser reiniciada

- Precisamos de memória

Tipos de memória

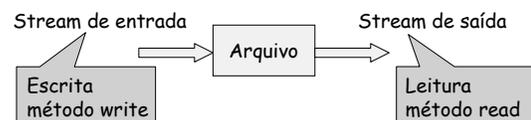
- Memória temporária (arrays, listas)
 - Principal
 - Rápida e cara
- Memória permanente (arquivos, SGBDs)
 - Secundária
 - Lenta e barata

Stream vs Arquivos

Streams representam fluxo de dados
Arquivos são representados por streams de bytes

Não existem tipos

Arquivos



Guardamos em arquivos ...

- Texto
- Objetos
- Tipos primitivos
- Bytes

Usamos as classes do pacote
java.io

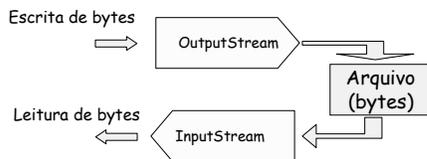
Arquivos binários

- InputStream/OutputStream
 - Leitura e escrita de arquivos binários
 - Estrutura abstrata
 - Ilegível para humanos
 - Unidade básica: bytes



Ligando aos arquivos

- FileInputStream/FileOutputStream
 - leitura e escrita de bytes em arquivos



Escrevendo em arquivos

```
OutputStream out =  
    new FileOutputStream(nome_arquivo);  
out.write((byte)32);  
out.close();
```

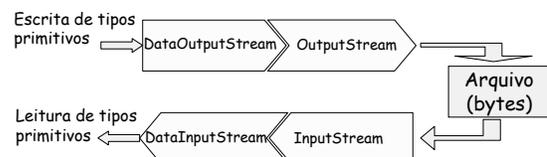
Sempre fechar
os arquivos
usando o método
close

Nome inclui o diretório
Existem outros
construtores

Lendo de arquivos

```
InputStream in =  
    new FileInputStream(nome_arquivo);  
byte b = in.read();  
in.close();
```

Leitura e escrita formatada



Leitura e escrita formatada

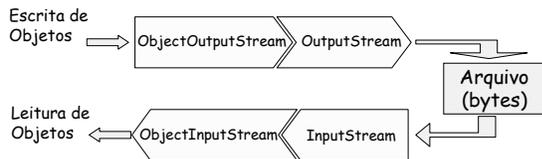
- **DataInputStream/DataOutputStream**
 - Escrita e leitura de tipos primitivos
 - readInt, writeInt
 - readFloat, writeFloat
 - ...
- **Strings**
 - Sequência de caracteres (bytes)
 - Problema: saber onde a string termina
 - readUTF, writeUTF (strings)
 - UTF representa strings de tamanho variável e terminação independente de plataforma

Ligando as classes

```
OutputStream out =  
    new FileOutputStream(nome_arquivo);  
DataOutputStream dataOut =  
    new DataOutputStream(out);
```

```
InputStream in =  
    new FileInputStream(nome_arquivo);  
DataInputStream dataIn =  
    new DataInputStream(in);
```

Objetos



Objetos

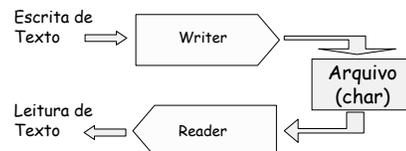
- **ObjectInputStream/ObjectOutputStream**
 - Escrita e leitura de Objetos
 - readObject, writeObject

Os objetos devem implementar a interface Serializable

Arquivos de texto

- Caracteres ao invés de bytes
- Possibilita leitura por humanos
- Tipos primitivos representados como strings
 - int 32 representado como "32"

Arquivos de texto



Classes usadas

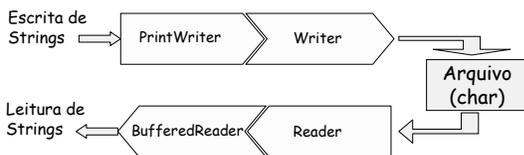
- **Reader/Writer**
 - Leitura e escrita de texto
 - Unidade básica: char
 - `int read()`
 - `write (int)`
- **FileReader/FileWriter**
 - leitura e escrita de texto em arquivos

Lendo e escrevendo texto

```
Writer out =  
    new FileWriter(nome_arquivo);  
out.write('a');  
out.close();
```

```
Reader in =  
    new FileReader(nome_arquivo);  
char c = (char)in.read();  
in.close();
```

Strings formatadas

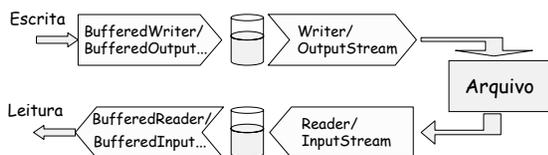


Strings formatadas

- **BufferedReader/PrintWriter**
 - Escrita e leitura de Strings
 - `String readLine()`
 - `write(String)`
 - `print(int), print(long), print(String), ...`
 - `println(int), println(long), println(String), ...`

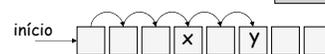
Melhorando performance

- **Usando buffers**
 - flush

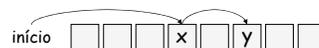


Acesso à Memória

- **Duas opções para ler x e depois y**
 - Sequencial



- Aleatório



Acesso randômico

- `RandomAccessFile`
 - Modo de leitura e escrita (`rw`)
 - Mode de somente leitura (`r`)
- Leitura e escrita formatada
 - `readInt`, `writeInt`, `readFloat`, `writeFloat` ...
 - `readUTF`, `writeUTF`
- Métodos `seek` e `skipBytes`

Escrevendo com acesso randômico

```
RandomAccessFile raf =  
    new RandomAccessFile("rw");  
raf.seek(10);  
raf.writeInt(32);  
raf.close();
```

Lendo com acesso randômico

```
RandomAccessFile raf =  
    new RandomAccessFile("r");  
raf.seek(10);  
int numero = raf.readInt();  
raf.close();
```

Resumindo...

- Memória primária e secundária
- Streams e arquivos
- Escrita e leitura em arquivos binários
- Decorações para abstrair bytes
- Escrita e leitura em arquivos de texto
- Decorações para abstrair caracteres
- Decorações para melhorar performance
- Tipos de acesso
- Usando acesso randômico

