Streams (2)
Serialização

Prof. Marcelo Cohen

1. Trabalhando com arquivos texto

- Streams trabalham com bytes
- Readers trabalham com caracteres
- Readers são utilizados para se trabalhar com arquivos texto

- *FileReader* : leitura

```
FileReader arqreader;
try
{
      arqreader = new FileReader(fname);
} catch(IOException e)
{
      System.err.println(e.getMessage());
}
```

- *FileWriter* : escrita

- FileReader e FileWriter trabalham com um caractere de cada vez, o que pode ser custoso
- Existem classes para fazer uma leitura/escrita de vários caracteres de uma vez só: *BufferedReader/BufferedWriter*
- Utilização:

```
BufferedReader in = new BufferedReader(new
  FileReader(”teste.txt"));

BufferedWriter out = new BufferedWriter(new
  FileWriter(”teste.txt"));
```

- Métodos de leitura: read(), readLine()
- Métodos de escrita: write(), writeLine()

2. Serialização

• Readers são utilizados para se trabalhar com arquivos texto

• É fácil gravar dados como strings, mas...

• Problema: como armazenar/recuperar uma estrutura de dados mais complexa ?

• Solução: **serializar** os objetos, ou seja, gravá-los de maneira que na hora de restaurá-los seja possível recriar as instâncias e reconectá-las da maneira correta.

• Como ? Fazendo os objetos implementarem a interface *Serializable*

• *Serializable* não tem métodos: serve apenas para indicar que os atributos deste objeto podem ser “serializados” e “deserializados”

Passo 1: declarar o objeto como serializável

```
import java.io.Serializable;

class Dado implements Serializable
{
  private int dado;
  public Dado(int n)
  {
     dado = n;
  }

  public void imp()
  {
    System.out.println("Dado = "+dado);
  }
}
```

Passo 2: gravar o objeto usando ObjectOutputStream

```
...
Dado dado = new Dado(20);
try
{
  FileOutputStream os = new FileOutputStream(”teste.dat");
  ObjectOutputStream oarq = new ObjectOutputStream(os);
  oarq.writeObject(dado);
  oarq.close();
}
catch(IOException e)
{
  System.out.println(e.getMessage());
  e.printStackTrace();
}
…
```

Passo 3: carregar o objeto usando ObjectInputStream

```
...
Dado dado1 = new Dado(20);
try
{
  FileInputStream is = new FileInputStream(”teste.dat");
  ObjectInputStream iarq = new ObjectInputStream(is);
  dado1 = (Dado) iarq.readObject();
  iarq.close();
catch(IOException e)
{
  System.out.println(e.getMessage());
  e.printStackTrace();
}
dado1.imp();
…
```