FIAP GRADUAÇÃO

FIAP

ENTERPRISE APPLICATION DEVELOPMENT

Prof. Me. Thiago T. I. Yamamoto

#08 – SERIALIZAÇÃO DE OBJETOS E SOCKETS

 *thiagoyama@gmail.com*  *thiagoyama*

TRAJETÓRIA

- JPA Introdução
- JPA API
- Design Patterns
- Relacionamentos
- JPQL
- Mapeamento Avançado
- Serialização de objetos e Sockets

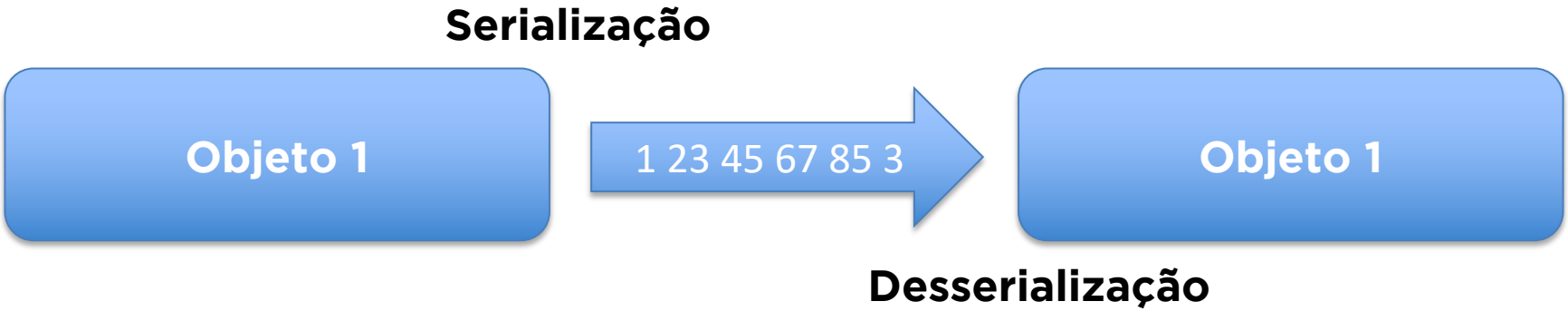
#08 - AGENDA

- Serialização
- Streams
- Sockets

SERIALIZAÇÃO

SERIALIZAÇÃO

- É o processo de **conversão de objetos** em uma **sequência ordenada** de **bits** (em série);
- A **serialização possibilita que os bytes** de um objeto sejam gravados em um **arquivo** ou **transmitidos pela rede**;
- O **processo inverso** da serialização chama-se **desserialização** e consiste na conversão de uma **sequência ordenada de bits em objetos**;
- Muito utilizado na **distribuição de objetos**;
- Somente o **valor** dos atributos (estado do objeto) são **serializados**.



SERIALIZAÇÃO EM JAVA

FIAP

- Para que um **objeto** seja **serializado** é necessário que sua classe implemente a interface **java.io.Serializable;**
- Cada classe que implementa Serializable possui um **identificador único** denominado **serialVersionUID** que é uma espécie de controle de versão da classe;
- Um objeto com um **serialVersionUID** serializado não poderá ser deserializado para uma classe com o **serialVersionUID diferente**;
- Atributos declarados como **transient** não são serializados

```
public class Pessoa implements Serializable {
    private static final long serialVersionUID = 7901296742271950 5L;
    //…
}
```

STREAMS

STREAMS

- Representam fluxos de dados entre uma origem e um destino:
  - **Origem**: memória do computador → **Destino**: um arquivo;
  - **Origem**: memória do computador → **Destino**: memória de outro computador na rede;

- **Fluxos de saída** são denominados **OutputStreams** onde bits podem ser escritos (write);

- **Fluxos de entrada** são denominados **InputStreams** onde bits podem ser lidos (read);

- **Objetos serializados** podem ser escritos ou lidos de um stream;

STREAM EM JAVA

FIAP

Stream de Arquivos

- **FileOutputStream**: permite escrever bytes em um arquivo;
- **FileInputStream**: permite ler bytes de um arquivo;

Stream de Objetos

É possível serializar objetos a partir de um stream já existente por meio das classes **ObjectOutputStream** e **ObjectInputStream** com os construtores:

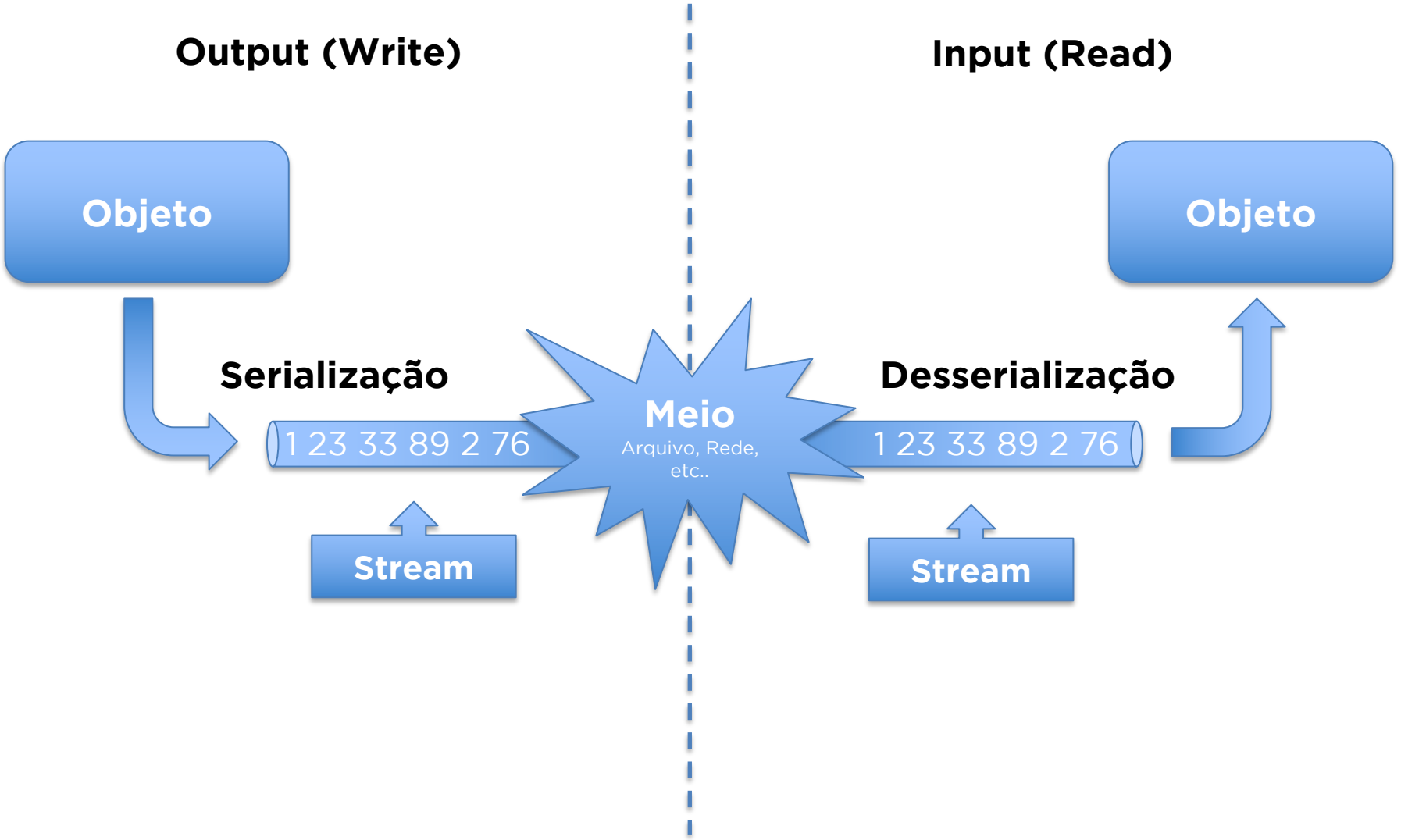
- **ObjectOutputStream (OutputStream P1)**: cria um output stream de objetos sobre um output stream já existente;
- **ObjectInputStream (InputStream P1)**: cria um input stream de objetos sobre um input stream já existente;

STREAMS EM JAVA

Dois **métodos** podem ser utilizados para escrita e leitura dos streams de objetos:

- **writeObject (Object)** → escreve estado (serializa) de um objeto serializável no **ObjectOutputStream;**
- **readObject()** → lê estado (deserializa) de um objeto serializável a partir de um **ObjectInputtStream**;

RESUMO STREAM E SERIALIZAÇÃO



SOCKETS

SOCKETS

FIAP

- Um ***socket*** representa o ponto final de um link de **comunicação** de duas vias entre **dois programas** executados em uma rede;
- **ServerSocket (int P1)**:
  - Atende requisições em uma determinada porta P1 por meio do método **accept();**
  - O método **accept()** suspende a execução do Thread até que um socket cliente estabeleça uma conexão;
- **Socket (String P1, int P2)**:
  - Para se conectar a um **ServerSocket**, especificar o endereço P1 e porta P2 do servidor;
  - Pode-se obter um *stream* de saída (*output*, *write*) e de entrada (*input*, *read*) de um socket por meio dos métodos **getInputStream()** e **getOutputStream()**;

SOCKETS – EXEMPLO SERVER

```
ServerSocket ss = new ServerSocket(8000);
Socket s = ss.accept();

InputStream is = s.getInputStream();

ObjectInputStream oi = new ObjectInputStream(is);

Object o = oi.readObject();

oi.close();
s.close();
```

SOCKETS – EXEMPLO CLIENTE

```
Socket s = new Socket("127.0.0.1",8000);
OutputStream os =  s.getOutputStream();

ObjectOutputStream oo = new ObjectOutputStream(os);

Object obj = new Object();

oo.writeObject(obj);

oo.close();
s.close();
```

VOCÊ APRENDEU..

- Como **serializar** e **desserializar** um objeto para um arquivo e através da rede;
- Para que serve o **serialVersionUID**;
- Trabalhar com **InputStream** e **OutputStream**;
- Realizar uma conexão **socket**;

FIAP

Copyright © 2013 – 2019
Prof. Me. Thiago T. I. Yamamoto

Todos direitos reservados. Reprodução ou divulgação total ou parcial deste documento é expressamente proíbido sem o consentimento formal, por escrito, do Professor (autor).

"O primeiro passo rumo ao sucesso é dado quando você se recusa ao ser um refém do ambiente em que se encontra"