Programação Paralela e Distribuída

Prof. Cidcley T. de Souza

Java Sockets

- Java Streams
  - Utilizados para enviar e receber dados pelos sockets, escondendo as características desses dados;
  - Derivados das classes OutputStream e InputStream;

Java Sockets

- Java Streams
  - OutputStream
    - Classe abstrata que define operações básicas com streams de saída;
    - Construtor: OutputStream()
    - void write(byte[] b): envia um conjunto de bytes pela stream;
    - void flush(): Força qualquer dado do buffer a ser enviado;
    - void close(): Fecha o stream;

Java Sockets

- Java Streams
  - DataOutputStream
    - Classe que permite se enviar tipos primitivos Java por uma stream de saída;
    - São oferecidos métodos para os tipos básicos Java, como: writeInt, writeShort, writeFloat, writeLong;
    - writeUTF é utilizado para se enviar uma string Java;

Java Sockets

- DataOutputStream(OutputStream out)
  - void writeInt(): Escreve um inteiro no stream
  - void writeShort(): Escreve um número inteiro no stream
  - void writeFloat(): Escreve um número real no stream
  - void writeUTF(): Escreve uma string no stream

Java Sockets

- Java Streams
  - InputStream
    - Classe abstrata que define operações básicas com streams de entrada;
    - Construtor: InputStream()
    - void read(byte[] b): lê um conjunto de bytes de uma stream;
    - void read(byte[] b, int off, int len): lê um conjunto de len bytes de uma stream de entrada;
    - void close(): Fecha o stream;

Java Sockets

- Java Streams
  - DataInputStream
    - Classe que permite se enviar tipos primitivos Java por uma stream de entrada;
    - São oferecidos métodos de leitura para os tipos básicos Java, como: readInt, readShort, readFloat, readLong;
    - readUTF é utilizado para receber uma string Java;

Java Sockets

- DataInputStream(InputStream in)
  - void readInt(): Lê um inteiro no stream
  - void readShort(): Lê um número inteiro no stream
  - void readFloat(): Lê um número real no stream
  - void readUTF(): Lê uma string no stream

Java Sockets

- Classes para Stream Sockets
  - Socket
    - Implementa um socket do tipo stream;
    - Os construtores dessa classe permitem a definição do endereço e porta de destino;
    - Os métodos getInputStream e getOutputStream retornam streams de entrada e saída, respectivamente, para se ler e escrever dados;

Java Sockets (Criação de um Socket TCP)

```
import java.net.*
import java.io.*
...
// Cria um socket conectado ao servidor web do IFCE
Socket socket = new Socket (“www.ifce.edu.br”, 80);

// Obtém stream de envio de dados
OutputStream out = socket.getOutputStream();

// Obtém stream de recebimento de dados
InputStream in = socket.getInputStream();
```

Java Sockets

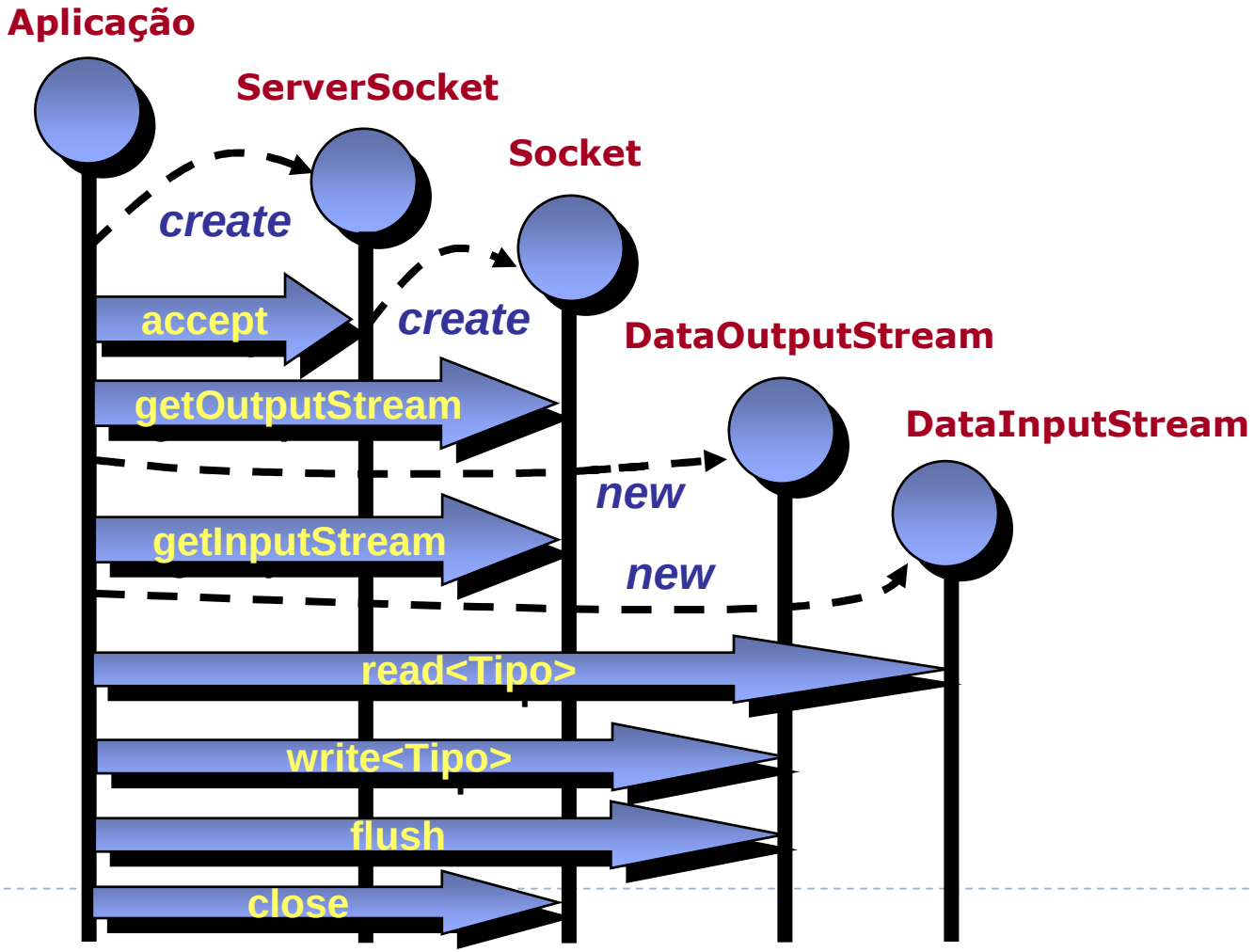
- Classes para Stream Sockets
  - ServerSocket
    - Implementa o lado servidor de um socket do tipo stream;
    - Escuta por conexões de clientes;
    - Pode ser passado como parâmetro para o construtor o endereço da porta que esse socket escutará;
    - O método accept, espera conexão e gera uma thread com um objeto do tipo Socket, para comunicação com o cliente;

Java Sockets (Criação de um ServerSocket TCP)

```
İmport java.net.*
...
// Cria um ServerSocket escutando à porta 1024
ServerSocket server = new ServerSocket(1024);
// Loop para tratamento de conexões
    while (true) {
// Aguarda uma nova solicitação de conexão de um cliente
        Socket cliente = server.accept();
// Trata a nova conexão aceita
            ...
// Fecha a conexão
        cliente.close();
}
```

Java Sockets

- Cenário Streams



Conclusão

- Um Socket é um mecanismo básico de comunicação entre sistemas;
- Totalmente padronizado;
- Totalmente portável;
- Socket = Programação em Rede;
- Ainda é muito baixo nível;