打开Eclipse，在之前创建的Java项目JavaNIO中创建包channel，创建类NioChannelServer

代码如下：

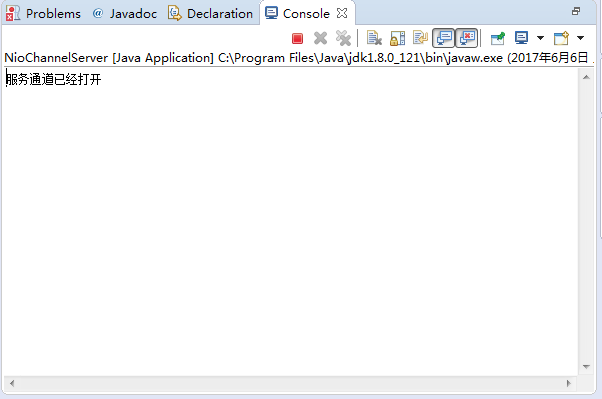
|  |
| --- |
| **package** channel;  **import** java.io.IOException;  **import** java.net.InetSocketAddress;  **import** java.nio.ByteBuffer;  **import** java.nio.IntBuffer;  **import** java.nio.channels.ServerSocketChannel;  **import** java.nio.channels.SocketChannel;  **public** **class** NioChannelServer {  **private** ByteBuffer buff = ByteBuffer.*allocate*(1024);  // 创建一个int缓冲区的视图 此缓冲内容的更改在新缓冲区中是可见的。反之亦然。  **private** IntBuffer intBuff = buff.asIntBuffer();  **private** SocketChannel clientChannel = **null**;  **private** ServerSocketChannel serverChannel = **null**;  /\*\*  \* 打开服务端的通道  \* **@throws** IOException  \*/  **public** **void** openChannel() **throws** IOException {  // 建立一个新的连接的通道  serverChannel = ServerSocketChannel.*open*();  // 为新的通道设置访问的端口  serverChannel.socket().bind(**new** InetSocketAddress(8888));  System.***out***.println("服务通道已经打开");  }  /\*\*  \* 等待客户端的请求连接  \* **@throws** IOException  \*/  **public** **void** waitReqConn() **throws** IOException {  **while** (**true**) {  clientChannel = serverChannel.accept();  **if** (**null** != clientChannel) {  System.***out***.println("新的连接加入！");  }  processReq();// 处理请求  clientChannel.close();  }  }  /\*\*  \* 处理请求过来的数据  \* **@throws** IOException  \*/  **public** **void** processReq() **throws** IOException {  System.***out***.println("开始读取和处理客户端数据...");  buff.clear();// 把当前位置设置为0， 上限值修改为容量的值。  clientChannel.read(buff);  **int** result = intBuff.get(0) + intBuff.get(1);  buff.flip();  buff.clear();  // 修改视图，原来的缓冲区也会变化  intBuff.put(0, result);  clientChannel.write(buff);  System.***out***.println("读取和处理客户端数据完成。");  }    **public** **void** start() {  **try** {  // 打开服务通道  openChannel();  // 监听客户端请求  waitReqConn();  clientChannel.close();  System.***out***.println("服务端处理完毕。");  } **catch** (IOException e) {  e.printStackTrace();  }  }  **public** **static** **void** main(String[] args) {  **new** NioChannelServer().start();  }  } |

在创建NioChannelClient类，代码如下：

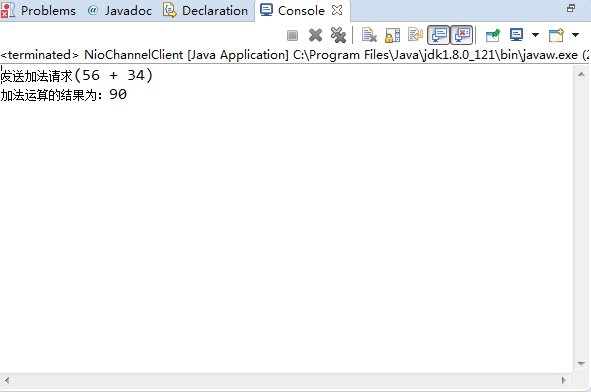
|  |
| --- |
| **package** channel;  **import** java.io.IOException;  **import** java.net.InetSocketAddress;  **import** java.nio.ByteBuffer;  **import** java.nio.IntBuffer;  **import** java.nio.channels.SocketChannel;  **public** **class** NioChannelClient {  **private** SocketChannel channel = **null**;  **private** ByteBuffer buff = ByteBuffer.*allocate*(8);  **private** IntBuffer intBuff = buff.asIntBuffer();  /\*\*  \* 与服务器指定的地址和端口建立连接通道  \*  \* **@return**  \* **@throws** IOException  \*/  **public** SocketChannel connect() **throws** IOException {  **return** SocketChannel.*open*(**new** InetSocketAddress("127.0.0.1", 8888));  }  /\*\*  \* 发送加法请求到服务器  \* **@param** a  \* **@param** b  \* **@throws** IOException  \*/  **public** **void** sendRequest(**int** a, **int** b) **throws** IOException {  buff.clear();  intBuff.put(0, a);  intBuff.put(1, b);  channel.write(buff);  System.***out***.println("发送加法请求(" + a + " + " + b + ")");  }  /\*\*  \* 接收服务器运算结果  \* **@return**  \* **@throws** IOException  \*/  **public** **int** receiveResult() **throws** IOException {  buff.clear();  channel.read(buff);  **return** intBuff.get(0);  }  /\*\*  \* 获得加法运算的结果  \* **@param** a  \* **@param** b  \* **@return**  \*/  **public** **int** getSum(**int** a, **int** b) {  **int** result = 0;  **try** {  channel = connect();  sendRequest(a, b);  result = receiveResult();  } **catch** (IOException e) {  e.printStackTrace();  }  **return** result;  }  **public** **static** **void** main(String[] args) {  **int** result = **new** NioChannelClient().getSum(56, 34);  System.***out***.println("加法运算的结果为：" + result);  }  } |

运行结果

服务端运行结果：



此处等待客户端的运行，此时运行客户端程序，结果如下：



此时服务端控制台已经输出：

