Eclipse 中创建项目JavaNIO

创建包buffer

创建类BufferTest

代码如下：

|  |
| --- |
| **package** buffer;  **import** java.nio.IntBuffer;  **public** **class** BufferTest {  **public** **static** **void** main(String[] args) {  // 创建指定长度的缓冲区  IntBuffer buff = IntBuffer.*allocate*(10);  **int**[] array = **new** **int**[] { 3, 5, 1 };  // 使用数组来创建一个缓冲区视图[3, 5, 1]  buff = IntBuffer.*wrap*(array);  // 利用数组的某一个区间来创建视图  // buff = IntBuffer.wrap(array, 0, 2);  // 对缓冲区某个位置上面进行元素修改  buff.put(0, 7);  // 遍历缓冲区中的数据  System.***out***.println("缓冲区数据如下：");  **for** (**int** i = 0; i < buff.limit(); i++) {  // 调用这个get方法会把pos自动递增  System.***out***.print(buff.get() + "\t");  }    System.***out***.println("\n原始数据如下：");  **for** (**int** a : array) {  System.***out***.print(a + "\t");  }    System.***out***.println("\n调用flip方法之前，Buffer类信息：" + buff);  buff.flip();// 对缓冲区进行反转，（limit = pos; pos = 0）  System.***out***.println("\n调用flip方法之后，Buffer类信息：" + buff);  buff.clear(); // 也是把pos清零      IntBuffer buff2 = buff.duplicate(); // duplicate函数是缓冲区  System.***out***.println(buff2); // 复制的缓冲区buff2与原始缓冲区buff具有同样的属性。    //// 0 <= mark <= position <= limit <= capacity    }  } |

在JDK1.8运行结果如下：

|  |
| --- |
| 缓冲区数据如下：  7 5 1  原始数据如下：  7 5 1  调用flip方法之前，Buffer类信息：java.nio.HeapIntBuffer[pos=3 lim=3 cap=3]  调用flip方法之后，Buffer类信息：java.nio.HeapIntBuffer[pos=0 lim=3 cap=3]  java.nio.HeapIntBuffer[pos=0 lim=3 cap=3] |