java.nio包介绍

# Package java.nio

Defines **buffers**, which **are containers for data**, and provides an overview of the other **NIO** packages.

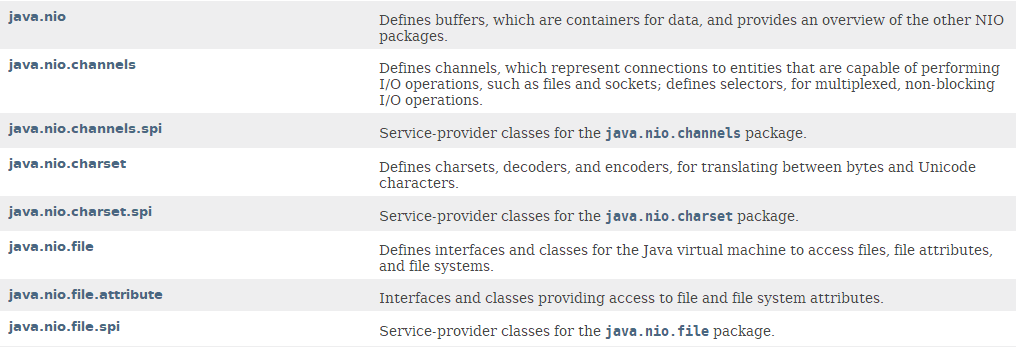
定义一些数据存放的**缓冲区**，为其他NIO包提供了预览。

**重要子包**： **java.nio.charset；java.nio.channels**

The **java.nio** package defines **the buffer classes**, which are used throughout the NIO APIs. The **charset** API is defined in the **java.nio.charset** package, and the channel and selector APIs are defined in the **java.nio.channels** package. Each of these subpackages has its own service-provider (SPI) subpackage, the contents of which can be used to extend the platform's default implementations or to construct alternative implementations.

Since JDK1.4。

所有的子包：



# NIO：

The central abstractions of the NIO APIs are: Buffers、Channels、Selectors

## Buffers

Buffers, which are containers for data;

## Charsets

Charsets and their associated decoders and encoders, which translate between bytes and Unicode characters;

## Channels

**Channels of various types**, which represent **connections** to entities capable of performing I/O operations;

## Selectors

**Selectors and selection keys**, which together with selectable channels define a multiplexed, non-blocking I/O facility.

# 缓冲区Buffers

## Buffer类概括(10个)

**Buffer** A container for data of a specific primitive type.

**ByteBuffer** A byte buffer.

ByteOrder A typesafe enumeration for byte orders.

MappedByteBuffer A direct byte buffer whose content is a memory-mapped region of a file.

**CharBuffer** A char buffer.

**DoubleBuffer** A double buffer.

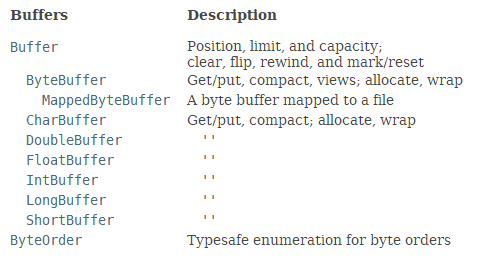
**FloatBuffer** A float buffer.

**IntBuffer** An int buffer.

**LongBuffer** A long buffer.

**ShortBuffer** A short buffer.

一共10个，其中Byte比较特殊，有ByteBuffer、ByteOrder、MappedByteBuffer三种，另外还有个Buffer。一共8-2+3+1 = 10.



## 描述

除了boolean之外，其他7中原生数据类型都有对应的缓冲区类XxxBuffer。

A buffer is a container for a fixed amount of data of **a specific primitive type**(原生类型). In addition to its content **a buffer has a position**, which is the index of the next element to be read or written, **and a limit**, which is the index of the first element that should not be read or written. The base Buffer class defines these properties as well as methods for clearing, flipping, and rewinding, for marking the current position, and for resetting the position to the previous mark.

There is a buffer class for **each non-boolean primitive type**. Each class defines a family of get and put methods for moving data out of and in to a buffer, methods for compacting, duplicating, and slicing a buffer, and static methods for allocating a new buffer as well as for wrapping an existing array into a buffer.

**Byte buffers与众不同**

**Byte buffers are distinguished** in that they can be used as the sources and targets of I/O operations. They also support several features not found in the other buffer classes:

### **A byte buffer** can be allocated as a direct buffer, in which case the Java virtual machine will make a best effort to perform native I/O operations directly upon it.

### **A byte buffer** can be created by mapping a region of a file directly into memory, in which case a few additional file-related operations defined in the **MappedByteBuffer** class are available.

### **A byte buffer** provides access to its content as either a heterogeneous or homogeneous sequence of binary data of any non-boolean primitive type, in either big-endian or little-endian byte order.

不支持Null，否则抛出NullPointerException异常

Unless otherwise noted, passing **a null argument** to a constructor or method in any class or interface in this package will cause a NullPointerException to be thrown.

# Buffer类及子类的具体介绍

## 父类Buffer：抽象类。

public **abstract** class **Buffer** extends Object

**7中基本数据类型对应的XxxBuffer**都继承Buffer类。

Direct Known Subclasses:

ByteBuffer, CharBuffer, DoubleBuffer, FloatBuffer, IntBuffer, LongBuffer, ShortBuffer

## 功能描述

A container for data of a specific primitive type.

A buffer is a linear, finite sequence of elements of a specific primitive type. Aside from its content, the essential properties of a buffer are its capacity, limit, and position:

**A buffer's capacity** is the number of elements it contains. The capacity of a buffer is never negative and never changes.

A buffer's limit is the index of the first element that should not be read or written. A buffer's limit is never negative and is never greater than its capacity.

A buffer's position is the index of the next element to be read or written. A buffer's position is never negative and is never greater than its limit.

There is one subclass of this class for each non-boolean primitive type.

## 特征

### Transferring data

### Marking and resetting

### Invariants

### Clearing, flipping, and rewinding

### Read-only buffers只读缓冲区

### Thread safety线程安全

### Invocation chaining调用链

## 定义的基本方法

### abstract Object array()

Returns the array that backs this buffer (optional operation).

### abstract int arrayOffset()

Returns the offset within this buffer's backing array of the first element of the buffer (optional operation).

### int capacity()

Returns this buffer's capacity.

Buffer clear()

Clears this buffer.

Buffer flip()

Flips this buffer.

abstract boolean hasArray()

Tells whether or not this buffer is backed by an accessible array.

boolean hasRemaining()

Tells whether there are any elements between the current position and the limit.

abstract boolean isDirect()

Tells whether or not this buffer is direct.

abstract boolean isReadOnly()

Tells whether or not this buffer is read-only.

int limit()

Returns this buffer's limit.

Buffer limit(int newLimit)

Sets this buffer's limit.

Buffer mark()

Sets this buffer's mark at its position.

int position()

Returns this buffer's position.

Buffer position(int newPosition)

Sets this buffer's position.

int remaining()

Returns the number of elements between the current position and the limit.

Buffer reset()

Resets this buffer's position to the previously-marked position.

Buffer rewind()

Rewinds this buffer.