

在 Android 音频处理的上下文中，`ByteArray` 和 `ShortArray` 的互相转换通常用于处理音频数据。一个 16 位的音频采样点可以表示为两个字节（即 `ByteArray` 中的两个元素），而 `ShortArray` 中的每个元素则直接表示一个 16 位的采样点。

## 1. `ByteArray` 转 `ShortArray`

假设 `ByteArray` 中的数据是 16 位 PCM 格式的音频数据，那么你可以将每两个字节转换为一个 `short`（注意字节的顺序）。这是一个小端序（Little Endian）的例子：

```
fun byteArrayToShortArray(byteArray: ByteArray): ShortArray {
    val shortArray = ShortArray(byteArray.size / 2)
    for (i in shortArray.indices) {
        shortArray[i] = ((byteArray[i * 2 + 1].toInt() shl 8) or
(byteArray[i * 2].toInt() and 0xFF)).toShort()
    }
    return shortArray
}
```

## 2. `ShortArray` 转 `ByteArray`

相似地，`ShortArray` 可以被转换为 `ByteArray`，每个 `short` 拆分为两个字节：

```
fun shortArrayToByteArray(shortArray: ShortArray): ByteArray {
    val byteArray = ByteArray(shortArray.size * 2)
    for (i in shortArray.indices) {
        byteArray[i * 2] = (shortArray[i].toInt() and 0xFF).toByte() // 低字
节
        byteArray[i * 2 + 1] = ((shortArray[i].toInt() shr 8) and
0xFF).toByte() // 高字节
    }
    return byteArray
}
```

## 3. 代码示例

下面是一个完整的代码示例，演示了如何进行这两个转换：

```

fun main() {
    // 示例数据：一个包含4个字节（即2个short值）
    val byteArray = byteArrayOf(0x01, 0x02, 0x03, 0x04)

    // 将 ByteArray 转为 ShortArray
    val shortArray = byteArrayToShortArray(byteArray)
    println("ShortArray: ${shortArray.joinToString()}") // 输出: ShortArray:
513, 1027

    // 再将 ShortArray 转回 ByteArray
    val convertedByteArray = shortArrayToByteArray(shortArray)
    println("ByteArray: ${convertedByteArray.joinToString()}") // 输出:
ByteArray: 1, 2, 3, 4
}

fun byteArrayToShortArray(byteArray: ByteArray): ShortArray {
    val shortArray = ShortArray(byteArray.size / 2)
    for (i in shortArray.indices) {
        shortArray[i] = ((byteArray[i * 2 + 1].toInt() shl 8) or
(byteArray[i * 2].toInt() and 0xFF)).toShort()
    }
    return shortArray
}

fun shortArrayToByteArray(shortArray: ShortArray): ByteArray {
    val byteArray = ByteArray(shortArray.size * 2)
    for (i in shortArray.indices) {
        byteArray[i * 2] = (shortArray[i].toInt() and 0xFF).toByte() // 低字
节
        byteArray[i * 2 + 1] = ((shortArray[i].toInt() shr 8) and
0xFF).toByte() // 高字节
    }
    return byteArray
}

```

## 4. 注意事项

- **字节序**：这个例子假设音频数据是以小端序存储的（即低字节在前，高字节在后）。如果你的数据是大端序（Big Endian），你需要调整字节顺序。
- **数据范围**：short 的范围是 -32768 到 32767，而 byte 的范围是 -128 到 127，所以直接将两个 byte 转换为 short 时要特别注意数据的有效范围和符号。

通过这些转换方法，你可以在 ByteArray 和 ShortArray 之间进行互相转换，以满足不同场景下的需求。