在使用 indexesOfObjectsPassingTest:方法后，迭代索引集合的每个索引，并添加相应的元素到数组中）。

表 15.11 列出了 NSIndexSet 的一些方法。其中，idx 是一个 NSUInteger 整型。你可以查看相关文档加深对这个类的了解。NSOrderedSet 和 NSMutableOrderedSet 是用来处理有序、不重复集合（类似于数组）的。

表 15.11　NSIndexSet 的一些方法

方　法	描　述
+(instancetype) indexSet	创建一个空的索引集合
-(BOOL) containIndex: *idx*	如果索引集合包含索引 idx，则返回 YES，否则返回 NO
-(NSUinteger) count	返回索引集合中索引的数量
-(NSUinteger) firstIndex	返回集合中的第一个索引，如果集合为空，则返回 NSNotFound
-(NSUinteger) indexLessThanIndex: *idx*	返回集合中小于 idx 的最接近的索引，如果没有小于 idx 的索引，则返回 NSNotFound，类似 indexLessThanOrEqualToIndex:、indexGreaterThanIndex 和 indexGreaterThanOrEqualIndex：
-(NSUinteger) lastIndex	返回集合的最后一个索引，如果集合为空，则返回 NSNotFound
-(NSIndexSet *) indexesPassingTest: (BOOL) (^) (NSUinteger *idx*, BOOL *stop) *block*	区块应用在集合中的每个元素。idx 添加到了 NSIndexSet 中返回 YES，否则返回 NO；设置指针变量 stop 为 YES，表示中断处理

15.6 练习

1. 在文档中查找 NSDate 类，然后向该类添加一个名为 ElapsedDays 的新分类。在这个新分类中，使用以下方法声明添加一个新方法：

```
-(unsigned long) elapsedDays: (NSDate *) theDate;
```

让新方法返回接收者到方法的参数之间经过的天数，并编写一个测试程序测试新方法（提示：可参见 years:months:days:hours:minutes:seconds:sinceDate:方法）。

2. 修改本章为类 AddressBook 编写的 lookup:方法，使它能够找出匹配的 name。消息表达式[myBook lookup: @"steve"]匹配名称中任何位置包含字符串“steve”的记录。将 sortedArrayUsingSelector:方法替换为本章最后出现的 indexesOfObjectsPassingTest:方法。

3. 使用练习 2 的结果修改 lookup:方法，使它能够搜索地址簿，并找到所有匹配的地址卡片，返回值为包含所有匹配的地址卡片的数组，若匹配不成功，则返回 nil。使用 sortedArrayUsingSelector:方法替换为本章最后出现的 indexesOfObjectsPassingTest:方法（注意，本章最后的例子中返回的是 NSIndexSet，但是这里希望返回 AddressCards 数组）。

4. 在 AddressCard 类添加新字段，建议将 name 字段分隔成姓氏和名字字段，添加地址（可能包含单独的州、城市、邮编和国家字段）和电话号码字段。编写合适的 setter 和 getter 方法，确定 print 方法和 list 方法能够显示这些字段。

5. 完成练习 4 后，修改练习 3 的 lookup:方法，使它能够对地址卡片中所有的字段进行搜索。能否想出一种方式设计 AddressCard 类和 AddressBook 类，使得后者不必了解存储在前者中的所有字段。

6. 在 AddressBook 类中添加 removeName:方法，以便删除地址簿中的某条记录。给定以下声明：

```
-(BOOL) removeName: (NSString *) theName;
```

使用练习 2 中的 lookup:方法。如果名字未查找到或者查找到多条记录，则方法返回 NO。如果记录成功移除，则返回 YES。

7. 使用在第一部分定义的 Fraction 类，根据任意一些值创建一个分数数组，为 Fraction 添加 description 方法。使用 3 种不同的方法显示分数的值：1）常规的 for 循环，2）快速迭代，3）使用%@。

8. 使用第一部分中定义的 Fraction 类，根据任意一些值创建一个可变的分数数组。使用 NSMutableArray 类的 sortUsingSelector:方法给数组排序，向 Fraction 类添加一个 Comparison 类型，实现 comparison 方法。

9. 定义 3 个新类，分别命名为 Song、Playlist 和 MusicCollection。Song 对象包含歌曲的信息，如歌曲名、艺术家、专辑、歌曲长度等；Playlist 对象包含播放列表名称和一个歌曲的集合；MusicCollection 对象包含播放列表集合，包括一个名为 library 的主播放列表，这个列表包含歌曲

集合中的所有歌曲。定义上述的 3 个类，编写方法实现以下任务：

- 创建一个 Song 对象，设置歌曲信息。
- 创建一个 Playlist 对象，向播放列表中添加和删除歌曲。如果一首新歌不在主列表中，将其添加进去。当从主播放列表删除一首歌时，也要从音乐集合的其他播放列表删除歌曲。
- 创建一个 MusicCollection 对象，并对集合添加和删除播放列表。
- 搜索和显示所有歌曲、播放列表或整个音乐集合的信息。

注意

这可能是最具指导性的练习，但不容易，图 15.6 演示了名为 mymusic 的MusicCollection 示例。它具有 3 个播放列表，其中有一个主播放列表 library 包含 5 首歌，playlist1 有两首歌，playlist2 有一首。这里有一个提示：充分利用 NSMutableArray 类存储的仅是每个新的播放列表中歌曲的引用（而不是复制）（使用 addObject:方法即可）。

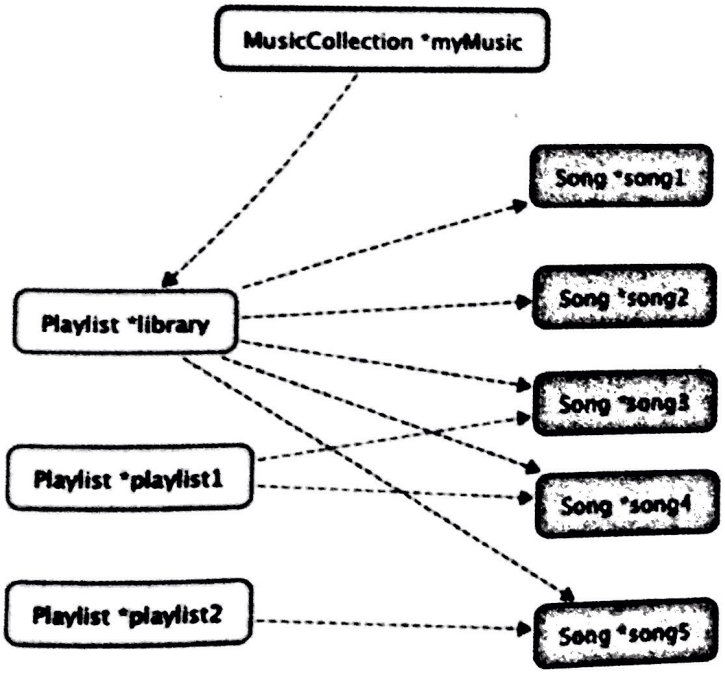

图 15.6　示例中的音乐集合数据结构

10. 编写一个程序，使用 NSNumber 对象的 NSArray（每个 NSNumber 代表一个整数）生成一个频率图表，列出每个整数和它在数组中出现的次数。

11. 当使用方法 addCard:向地址簿添加地址卡片时，谁拥有这些地址卡片？这些卡片上的信息发生改变是否会影响到存储在地址簿中的卡片？思考一下，如何实现一个更加安全的 addCard:方法。