

```
1 public interface Collection<AnyType> extends Iterable<AnyType>
2 {
3     int size( );
4     boolean isEmpty( );
5     void clear( );
6     boolean contains( AnyType x );
7     boolean add( AnyType x );
8     boolean remove( AnyType x );
9     java.util.Iterator<AnyType> iterator( );
10 }
```

图 3-5 java.util 包中 Collection 接口的子集

在 Collection 接口中的许多方法所做的工作由它们的英文名称可以看出, 因此 size 返回集合中的项数; isEmpty 返回 true 当且仅当集合的大小为 0。如果 x 在集合中, 则 contains 返回 true。注意, 这个接口并不规定集合如何决定 x 是否属于该集合——这要由实现该 Collection 接口的具体的类来确定。add 和 remove 从集合中添加和删除 x, 如果操作成功则返回 true, 如果因某个看似有理(非异常)的原因失败则返回 false。例如, 如果要删除的项不在集合中, 则 remove 可能失败, 而如果特定的集合不允许重复, 那么当企图插入一项重复项时, add 操作就可能失败。

Collection 接口扩展了 Iterable 接口。实现 Iterable 接口的那些类可以拥有增强的 for 循环, 该循环施于这些类之上以观察它们所有的项。例如, 图 3-6 中的例程可以用来打印任意集合中的所有的项。这种方式的 print 的实现和当 coll 具有类型 AnyType[] 时能够使用的相应的实现是完全相同的, 它们逐个字符都是一样的。

```
1 public static <AnyType> void print( Collection<AnyType> coll )
2 {
3     for( AnyType item : coll )
4         System.out.println( item );
5 }
```

图 3-6 在 Iterable 类型上使用增强的 for 循环

### 3.3.2 Iterator 接口

实现 Iterable 接口的集合必须提供一个称为 iterator 的方法, 该方法返回一个 Iterator 类型的对象。该 Iterator 是一个在 java.util 包中定义的接口, 见图 3-7。

```
1 public interface Iterator<AnyType>
2 {
3     boolean hasNext( );
4     AnyType next( );
5     void remove( );
6 }
```

图 3-7 java.util 包中的 Iterator 接口

Iterator 接口的思路是, 通过 iterator 方法, 每个集合均可创建并返回给客户一个实现 Iterator 接口的对象, 并将当前位置的概念在对象内部存储下来。

每次对 next 的调用都给出集合的(尚未见到的)下一项。因此, 第 1 次调用 next 给出第 1 项, 第 2 次调用给出第 2 项, 等等。hasNext 用来告诉是否存在下一项。当编译器见到一个正在