java.util.Iterator 主要用于迭代访问(即遍历)Collection中的元素.故Iterator对象被称为迭代器.

迭代:Collection集合元素通用获取方式。取前先判断集合中有无元素,若有则取出,继续再判断,直到把元素全部取出.

常用方法: public boolean hasNext():判断是否可以迭代
public E next():返回迭代的下一元素
public void remove():从指向的collection中移除迭代器返回的最后一个元素

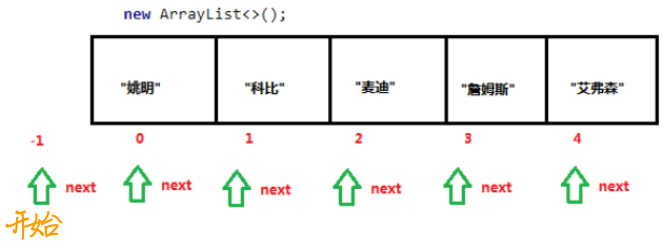
Iterator迭代器是一个接口,需实现类.
Collection接口中有一个方法:Iterator<E> iterator().返回的是迭代器的实现对象

```java
//创建一个集合对象
Collection<String> coll = new ArrayList<>();
//往集合中添加元素
coll.add("姚明");
coll.add("科比");
coll.add("麦迪");
coll.add("詹姆斯");
coll.add("艾弗森");

/*
    1.使用集合中的方法iterator()获取迭代器的实现类对象,使用Iterator接口接收(多态)
    注意:
        Iterator<E>接口也是有泛型的,迭代器的泛型跟着集合走,集合是什么泛型,迭代器就是什么泛型
 */
//多态  接口              实现类对象
Iterator<String> it = coll.iterator(); 获取迭代器的实现类对象,并且会把指针(索引)指向集合的-1索引


/*
    发现使用迭代器取出集合中元素的代码,是一个重复的过程
    所以我们可以使用循环优化
    不知道集合中有多少元素,使用while循环
    循环结束的条件,hasNext方法返回false
 */
while(it.hasNext()){  判断集合中还有没有下一个元素
    String e = it.next();  做了两件事情:
    System.out.println(e);
}
```

1.取出下一个元素 "姚明" "科比" "麦迪" "詹姆斯" "艾弗森"
2.会把指针向后移动一位




增强For循环:foreach 仅作为遍历操作出现
底层也是迭代器,所有的单列集合的遍历都可用
快要有被遍历的目标(只能是Collection或数组)