增强型，for循环

使用增强型for循环遍历集合 – Java5.0出现的新特性，该循环不通用于传统for循环工作，其知识用于集合或者数组的遍历

语法：for（元素类型 元素名：集合或者数组名）{

}

增强型for循环不是新语种，知识在编译器在编译生成.class文件时，将新循环转成迭代器，所以在遍历集合时，不能通过集合的方式来改变集合

增强型for循环的本质为迭代器。

注意点：不允许在遍历的过程中修改集合元素，因为本质是迭代器

泛型：

1. 主要作用是限制集合中所存放的元素的类型，方便后期对集合数据的操作，所有集合都支持泛型，迭代器也支持泛型
2. Jdk1.5之后引入的概念

<Integer> 🡪泛型

List 集合：

1. 是可以通过元素下标来操作元素（下标是从0开始）
2. ArrayList：数组结构—检索数组比较快
3. LinkedList：链表结构—插入和删除速度快
4. 在处理数据量比较大的时候，体现的较为明显
5. subList(int index, int endIndex)如果子集发生改变，对应的原集合也会发生改变
6. 集合与数组之间的转换
   1. toArray()：自动转成Object类型的数组，比较麻烦
   2. toArray(new 引用类型[数组长度])：这种转换方式会直接转成我们想要的类型的数组
      1. 数组长度：定义的数组长度，大于集合元素的个数，多余的部分，补充null，
      2. 定义的数组长度，小于集合元素的个数，系统会给我们定义一个默认为集合个数的数组长度，不再使用我们自己定义的数组长度
7. 数组转集合：一般是转成List集合（set中不允许重复）
   1. asList，转成对应泛型的List集合
   2. 从数组转成集合后，不能向集合中添加元素，否则会报异常，需要重新建立一个集合，将该转成后得集合复制过去，就可以想新集合中添加元素
   3. 使用addAll()使用创建集合的有参数的构造方法

Collections：是关于一个Collection接口集合的一个类（工具类），里面封装着一些对集合操作的特定的方法

Sort(List<T> list)：按照默认的排序顺序，给给定的集合进行排序