日历类在：java.util.Calendar

因为Date类对时间分量的操作不是很方便，所以引入Calendar

该类是一个抽象类，其方法可以对时间分量进行逐一设置

Add(int fiel, int num); 直接在时间分量上加减

加：正数

减：负数

使用add增减天数，（会自动根据日历计算）

集合：

是新的数据结构，可以保存不同类型的元素，

而且可以动态拓展

集合由来：

数组定义，数组的长度就固定，不可更改，当添加元素超过了数组的长度，就需要重新定义数组，只能存相同类型的元素

Java内部给我们提供了集合类，该类的实例可以存任意对象，长度是可以动态改变的，随着元素的增加或者减少而改变空间大小

集合和数组的区别：

1. 数组既可以存基本类型的数据，也可以存引用类型数据，基本类型存的是值，引用类型存的是地址值，
2. 集合只能存储引用类型数据，存储的是地址值，也能存储基本类型的数据，但是在存储过程中，会自动装箱成引用类型存入集合
3. 数组长度是固定的，不能自动增减，集合长度是不定的，可以随着元素的增减而动态改变
   1. 如果元素个数是固定的，则推荐使用数组
   2. 如果元素的个数是不定的，则推荐使用集合

集合的继承体系：

1. 父接口：Collection—(Collections集合的工具类)
   1. 子接口：继承Collection接口
      1. List：有序，可以重复的集合 -- 元素下标
         1. 存取是有顺序的
         2. 实现类：ArrayList：数组式（数组实现）
            1. LinkedList：链表式（链表实现）
            2. Vector：数组式
      2. Set：无序，不可以重复的集合（部分有序）--- 元素值
         1. 存和取是一样的
         2. 实现类：
            1. HashSet：哈希算法
            2. TreeSet：二叉树算法

Collection.add(null) 此时Collection不是空

集合的遍历：两种

1. 迭代器遍历
   1. 迭代器遍历集合：Iterator接口
      1. Java希望我们使用统一的方式来遍历集合，定义了遍历集合的接口，同时定义了如何遍历集合—使用迭代器
      2. 迭代器遍历的步骤：
         1. 获取迭代器
         2. 问：hasNext()：询问是否有下一个元素，给予指针的操作，每次调用该方法，指针会下移，返回的是当前指针指向的元素
         3. 取
         4. 删除：在用迭代器遍历的时候，不能使用集合的删除方法，否则会报错（Java.util.ConcurrentModificationException）
2. 增强型for循环