Grouping Objects Using Lists and Nested For Loops

Container objects - hold and manage collections of objects

Lists - store sequence of values in order

Sets - store unordered collection of values

Dictionaries/Maps - store lookup tables to find associated data

Interface - only lists declarations (basically names) of the public methods in the class ; you can create an object and implement the methods using the class, since the interface names them already

- regular fields cannot be declared in an interface, although constants can be.
- to create an actual object you need a concrete class, one that implements the interface

```java
public interface Drawable
{
    public void draw();
}
public interface Drawing
{
    public void drew();
}

// corresponding classes
public class Rectangle
    implements Drawable,Drawing
{
    // ...
    public void draw()
    {
        // ...
    }
}

public class Circle
    implements Drawable
{
    // ...
    public void draw()
    {
        // ...
    }
}
```

Lists interface (generic type)

```java
List<String> names = ...;
```

```
names.add("Sara");

List<Jeroo> jeroos = ...;

jeroos.add(new Jeroo());
```

ArrayList

- ArrayList is a class that implements the List interface.
- resizable array

```java
List<Integer> list = new ArrayList<Integer>();
list.add(-2);

int x = list.get(1); // get 2nd item in list

list.set(1, 4); // change 2nd item to value of 4
```

	Some List Interface Methods
Method Name	**Purpose**
`add(<some value>)`	adds an item to the list
`get(int <some index>)`	returns the item stored at this index
`set(int <some index>, <some value>)`	sets the item at some index to be some value
`clear()`	removes all elements from the list
`isEmpty()`	returns `true` if a list doesn’t have any values stored in it, otherwise `false`
`remove(int <some index>)`	removes element at the specified index from the list
`size()`	returns the number of elements in the list
`contains(<some value>)`	returns `true` if the value is in the list, otherwise `false`
`add(<some index>, <some value>)`	inserts an item into the list at the specified position, moving other items back by one to make room

Nested for-loops - a for-loop inside a for-loop