9/26/2007
Do you have any questions about the quiz?
Let's look at some solutions to the interclass problem.
Alice has a second data structure called an array. Arrays are pretty much just like lists except that they have a fixed length.

With an array you can only retrieve values or overwrite values.

The indexes in an array, like those in a list, start at zero.

There are fewer array operations than list operations.

- [] - to get an element
- set - to set an element
- length – to tell us how many slots on in the array.

Arrays provide efficient “random access”.
For your project you might want some things to happen a bit differently every time it runs. For this you need random numbers.

The world has two methods to return pseudo-random boolean and double values. (For our purposes a double is just a number.)
If you tried to animate your list by giving the objects on your list a walk method you probably found there was a problem. Alice won't let you say `item_from_list.walk`. This is because `item_from_list` could be any object in the Alice world and not all objects in Alice have a walk method.

How then do we animate objects using a list/array in Alice? The `partNamed` method lets us pull of sub-parts of an object to move for the animation. This is more tedious than I would like, but it follows the strong static typing of Java and the lack of subtyping in Alice.
Is there any part of your project where you might use random numbers? If so, when?
You can also let me know if you have any questions about lists or arrays.
Interclass Problem – Do problem 5.5 of 5.7.