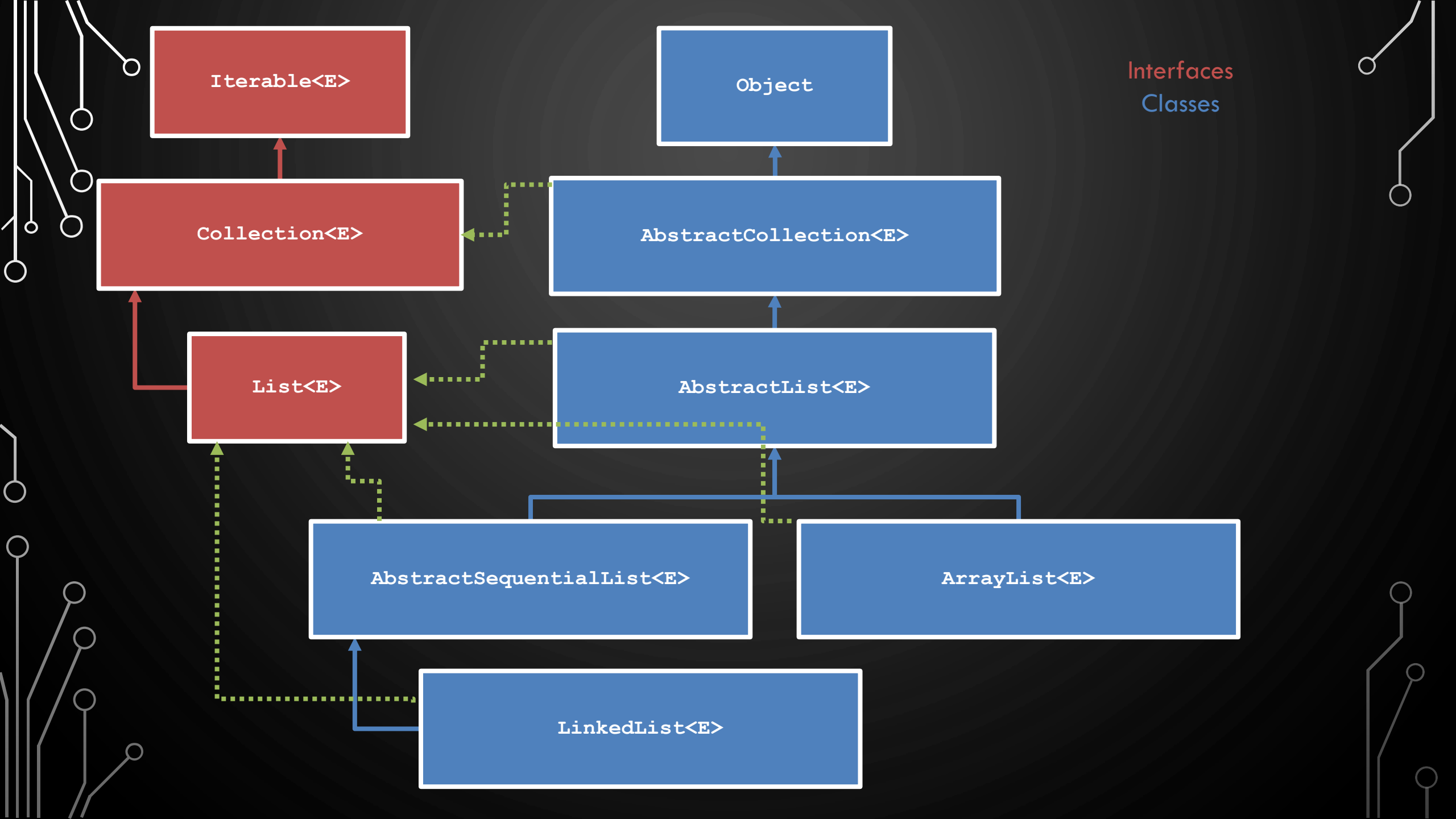
JAVA LISTS

SUMMARY OF CLASSES CONCERNING LISTS

- `Vector<E>` - Growable-array using incremental strategy (sort of deprecated)
- `ArrayList<E>` - Growable-array using doubling strategy (supports List)
- `LinkedList<E>` - Doubly linked list (supports List, Deque, Stack, and Queue)

- Others outside the scope of this course
- To find how to use them, go to the Java API!


Iterable<E>
Object
Interfaces
Classes
Collection<E>
AbstractCollection<E>
List<E>
AbstractList<E>
AbstractSequentialList<E>
ArrayList<E>
LinkedList<E>

EXAMPLE OF USING ARRAYLIST<E>

```
1. Scanner s = new Scanner(new File("numbers.txt"));
2. ArrayList<Integer> numbers = new ArrayList<Integer>();
3. while(s.hasNextInt())
4.   numbers.add(s.nextInt());
5. …elsewhere…
6. int sum = 0;
7. for(int n = 0; n < numbers.size(); ++n)
8.   sum += numbers.get(n);
```

CHOOSING ARRAYLIST VS LINKEDLIST

- General guideline – 95% of the time ArrayList should be the go-to List – note this is a made up statistic, based on my experience.
- Educated guess
  - Start with array list for quick implementation
  - If linked list provides better big-oh complexity switch to it
  - Otherwise, you need to experiment with both to make best selection

PROBLEMS

- Linear regression. Lets help the sciences by creating a simple program for linear regression modeling. Look here for how we compute correlation coefficients. Here is experimental data.
- For a given data file, find the correlation coefficient between all pairs of columns. Find the most correlated items.
- I recommend trying the solve this problem for x08.txt
- Lets discuss together how to break the problem down into manageable pieces.
  - Use my starter code for parsing the file. You need to modify the parse to put the data into your data structure.