From Comparable to Comparator

- When a class implements `Comparable` then ...
  - `Comparable<...>` types the parameter to `compareTo`
  - Return `< 0, == 0, > 0` according to results of comparison

- Suppose we want to change how Strings compare
  - Or change class `Foo` implements `Comparable<Foo>`
  - What if we need more than one way to compare `Foo`'s?

- New interface to the rescue: Comparator
  - Change how String or Foo or ... compares
  - Comparator object is argument to sort, for example

java.util.Comparator

- How does sorting work in general and in Java?
  - Characteristics of Java library sort methods
  - What can be sorted?
  - How do you change how sorting works?

- Use APT ClientsList as an example to explore Comparator
  - Creating new Comparator: nested class
    - Should it be public? Private? Matter?
  - Comparator could anonymous, but then issues.

- What does it mean to implement `Comparable`?
  - Other Java interfaces: `cloneable`, `serializable`, ...