ArrayList
ArrayList

• ArrayList class
  – ArrayList has two significant advantages over arrays
    • Array lists can grow or shrink as needed
    • The ArrayList class supplies many methods for common tasks, such as inserting or deleting
ArrayList

- ArrayList is a generic class. It takes a type parameter, which can be any class.

- To create an empty ArrayList of Rectangle, do
  - `ArrayList<Rectangle> r = new ArrayList<Rectangle>();`
  - to add an element, `r.add(new Rectangle(2,3));`
  - to access an element: `Rectangle r0 = r.get(0);`
ArrayList<T>: other methods

- add(int i, T t): insert an element t at index i
- remove(int i): remove the element at index i
- set(int i, T t): replace the element at index i with t
- get(int i): returns the element at index i
- size(): return the number of elements in the current ArrayList
Wrapper classes

- Since numbers are not objects in Java, they cannot be put directly into an ArrayList. To store numbers in an array list, you must turn them into objects by using wrapper classes.
  - char → Character
  - double → Double
  - int → Integer
Auto-boxing (unboxing)

• Fortunately, conversion between primitive types and their corresponding wrapper classes is automatic. For example,
  – Double d = 29.95; // auto-boxing; same as
    // Double d = new Double(29.95)
  – double x = d; // auto-unboxing; same as x = d.doubleValue();
  – d = d + 1; // legal

• In order to collect numbers in an ArrayList, do
  – ArrayList<Double> values = new ArrayList<Double>();
  – values.add(29.95); // same as: values.add(new Double(29.95));
  – double x = values.get(0);