What can you put into an ArrayList?

- Any Object
- Use a wrapper class (see java.lang.*)
  - int, double, char, boolean...
  - Integer, Double, Character, Boolean
- Can have your cake and eat it too
  ```java
  ArrayList<Integer> list = new ArrayList<Integer>();
  for (int k = 0; k < 10; k++) {
    list.add(k*k);
  }
  for (Integer jj : list) {
    System.out.println(jj);
  }
  ```
- All made practical by Version 5 of Java

Exploring ArrayLists

- Look at the Java 6 API
- Note interfaces implemented
  - Serializable, Cloneable, Iterable
  - Collection, List, RandomAccess
- Note other descriptive text
  - Regarding performance
  - Constructors
  - Methods
  - Don’t forget methods in parent classes

Exploring ArrayLists

- Some Commonly Used Methods
  - boolean add(E o) // append
  - void add(int index, E element) // insert
  - void Clear()
  - boolean contains(Object elem)
  - E get(int index)
  - int indexOf(Object elem)
  - boolean remove(Object o)
  - E remove(int index)
  - E set(int index, E elem) // replace
  - int size()
What is a char?

- Differences between unicode and ASCII
  - Why is unicode used? Why should we care? What should we know? How many of the details are important?

- A char value can be treated like an int value
  - Add integer to it, cast back to char
  - Subtract character from it, get int back

  ```
  counters[s.charAt(k) - 'A']++;
  ```

  - Anatomy of the statement above??

View Model Communication

- View interacts with user
  - Load file
    - What method in model called?
  - Start new game
    - What method in model called?
  - User guesses word (when?)
    - What method in model called?
  - User responds with # letters in common (when?)
    - What method in model called?

Model View Communication

- Informative messages
  - I know 5432 words
    - showMessage(…)
- Messages requiring interaction (modal dialog)
  - You guessed my word
  - You’ve guessed that word before (or ...?)
    - showModalInfo(…)
- Model changes and notifies view
  - I’m thinking of a secret word
  - Your word has 4 letters in common, you’ve used one guess
    - processModelResponse(…)

Inheritance and models

- We’ll see other models in other programs
  - What interface in common?
  - AbstractModel implements code and calls unimplemented methods
    - Who implements these methods?
    - Hollywood principle, template pattern