ArrayList

- Constructors
  - ArrayList lst1 = new ArrayList();
  - ArrayList lst2 = new ArrayList(int initialSize);
  - ArrayList<String> strList = new ArrayList();

- Parameterized type
  - use if you know the type of the list and the list type is not mixed

- Methods
  - size() // Returns the num of items held.
  - add(Object o) // Appends o to end.
  - add(int idx, Object o) // Inserts o at pos idx.
  - remove(int idx) // Removes item at pos idx.
  - remove(int idx) // Gets item at idx. No removal.
  - set(int idx, Object o) // Replaces item at idx with o.
  - clear() // Removes all items.
  - isEmpty() // true if empty.
  - toArray() // Returns an array that contains the contents of the list

Removing items from ArrayList while iterating

- When an item is removed from an ArrayList, the list shrinks and the indices are renumbered behind the removed item
- Why doesn't this removal work?
  - for (int i=0; i<lst.size(); i++) {
    lst.remove(i);
  }
- Must remove from the back to the front
  - for (int lst.size()-1; i>=0; i--) {
    lst.remove(i);
  }

Word class

class Word {
  String word;
  int freq;

  Word (String word) {
    this.word = word;
    freq = 1;
  }

  void inc() {
    freq++;
  }

  String getWord() {
    return word;
  }

  int getFreq() {
    return freq;
  }

  String toString() {
    return "<"+word+","+freq+">";
  }
}

Filter

String raw:
String delimiters = " ,.?!;:-’"()*![]{}|\~`@#$%^&";
String[] fileText, words, uniqueWords;
int[] freqs;

void setup() {
  fileText = loadStrings("EliotLoveSong.txt");
  println("Read " + fileText.length + " lines.");
  raw = join(fileText, " ");
  raw = raw.toLowerCase();
  words = splitTokens(raw, delimiters);
  println("Found " + words.length + " words.");
  freqs = makeUnique(words);
  println("Found "+uniqueWords.length+" unique words.");
}

makeUnique

void makeUnique(String[] words) {
  uniqueWords = new ArrayList();
  for (int i=0; i<words.length; i++) {
    int idx = contains(words[i], uniqueWords);
    if (idx < 0) {
      uniqueWords.add(new Word(words[i]));
    } else {
      uniqueWords.get(idx).inc();
    }
  }
}

int contains(String t, ArrayList<Word> lst) {
  for (int i=0; i<lst.size(); i++) {
    if (t.equals(lst.get(i).getWord()))
      return i;
  }
  return -1;
}
Stop words removal

- The most common short function words
  - the, is, a, at, which, on, etc
- Usually filtered out
- The list is not unique or definitive

```java
fileText = loadStrings("stopwords.txt");
stopwords = new ArrayList(fileText.length);
for (int i=0; i < fileText.length; i++) {
    stopwords.add(fileText[i].toLowerCase());
}
```

makeUnique without stop words

```java
void makeUnique(String[] words) {
    uniqueWords = new ArrayList();
    for (int i=0; i < words.length; i++) {
        int idx = contains(words[i], uniqueWords);
        if (idx < 0) {
            uniqueWords.add(new Word(words[i]));
            if (!stopwords.contains(words[i])) {
                uniqueWords.get(idx).inc();
            } else {
                uniqueWords.get(idx).inc();
            }
        }
    }
}
```

Sorting

- Any process of arranging items in sequence
- Build-in sort()
  - Works on arrays of simple types, i.e. int, float and String
    - float[] a = {3.4, 3.6, 2.0, 7.1};
    - a = sort(a);
    - String[] s = {"deer", "elephant", "bear", "aardvark", "cat");
    - s = sort(s, 3);
- Convenient, but not very flexible
- Recall that we have an ArrayList<Word>

Implement your own sort

- Many sorting algorithms
- Bubble Sort
  - Looks at items in successive pairs
  - Swap if in the wrong order
- Selection Sort
  - Scan a list start to end and find the value that should come first
  - Swap that item into the first position
  - Repeat scanning and swapping starting at the second position in the list
- Insertion Sort

Example: Text Processing & Visualization

Obama Nobel Peace Prize Acceptance Speech: December 2009

The other is a conflict that America did not seek; one in an effort to defend ourselves and all nations from further attacks. A world needed insatutions to prevent another World War. And so, World War II was a conflict in which the total number of civilians who died exceeded the number of soldiers who perished. Blurred. In the span of thirty years, such carnage would just than the defeat of the Third Reich and the Axis powers, but perhaps the most profound issue surrounding my receipt of this prize — Schweitzer...?
We can do that - this is the story of human progress; that is the hope of all the armed conflict - filled with difficult questions about the relationship between war and will kill. Some will be killed. And so I come here with an acute sense of the cost of beginning, and not the end, of my labors on the world stage. Compared to some of the mere prisoners of fate. Our actions matter, and can bend history in the direction of I receive this honor with deep gratitude and great humility. It is an award that speaks to...
Chris Rock @ Oscars

Sales Data
(US $ in thousands)

<table>
<thead>
<tr>
<th>Region</th>
<th>Jan</th>
<th>Feb</th>
<th>Mar</th>
<th>Apr</th>
<th>May</th>
<th>Jun</th>
<th>Jul</th>
<th>Aug</th>
<th>Sep</th>
<th>Oct</th>
<th>Nov</th>
<th>Dec</th>
</tr>
</thead>
<tbody>
<tr>
<td>Domestic</td>
<td>1983</td>
<td>2343</td>
<td>2593</td>
<td>2283</td>
<td>2574</td>
<td>2838</td>
<td>2382</td>
<td>2634</td>
<td>2938</td>
<td>2739</td>
<td>3413</td>
<td></td>
</tr>
<tr>
<td>International</td>
<td>574</td>
<td>636</td>
<td>573</td>
<td>593</td>
<td>644</td>
<td>679</td>
<td>593</td>
<td>139</td>
<td>599</td>
<td>583</td>
<td>602</td>
<td>690</td>
</tr>
</tbody>
</table>

Top Medals in Olympics by Country (1992-2012)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>United States of America</td>
<td>104</td>
<td>110</td>
<td>103</td>
<td>92</td>
<td>101</td>
<td>108</td>
</tr>
<tr>
<td>People’s Republic of China</td>
<td>88</td>
<td>100</td>
<td>63</td>
<td>59</td>
<td>50</td>
<td>54</td>
</tr>
<tr>
<td>Russian Federation</td>
<td>82</td>
<td>72</td>
<td>92</td>
<td>88</td>
<td>63</td>
<td>112</td>
</tr>
<tr>
<td>Great Britain</td>
<td>85</td>
<td>47</td>
<td>30</td>
<td>28</td>
<td>15</td>
<td>20</td>
</tr>
<tr>
<td>Australia</td>
<td>35</td>
<td>46</td>
<td>58</td>
<td>41</td>
<td>27</td>
<td></td>
</tr>
<tr>
<td>Germany</td>
<td>44</td>
<td>43</td>
<td>56</td>
<td>85</td>
<td>82</td>
<td></td>
</tr>
<tr>
<td>France</td>
<td>34</td>
<td>40</td>
<td>33</td>
<td>37</td>
<td>29</td>
<td></td>
</tr>
<tr>
<td>Republic of Korea</td>
<td>28</td>
<td>31</td>
<td>28</td>
<td>27</td>
<td>29</td>
<td></td>
</tr>
<tr>
<td>Japan</td>
<td>28</td>
<td>25</td>
<td>18</td>
<td>14</td>
<td>22</td>
<td></td>
</tr>
<tr>
<td>Italy</td>
<td>28</td>
<td>27</td>
<td>32</td>
<td>34</td>
<td>35</td>
<td>19</td>
</tr>
</tbody>
</table>

Top Medals in Olympics (1992-2012)

Top Medals in Olympics (1992-2012)

Example: Visualizing Time Series

Raw Data
1,4,2010,213,4,3,214,50,214,01,17832300
1,5,2010,214,6,0,215,58,214,38,21496800
1,6,2010,214,7,215,23,210,87,18790000
1,7,2010,214,8,215,18,210,23,18790000
1,8,2010,215,5,215,18,211,4,17832300
12,20,2010,213,9,325,44,324,68,8823000
12,20,2010,213,10,326,45,325,47,52693000
12,20,2010,213,11,326,45,325,29,52694400
12,20,2010,213,12,325,46,325,71,52664800
12,31,2010,213,12,325,48,322,56,69110000
Visualizing Time Series

Data Visualization - Horizons