YEAH session #6

6 March 2014, 5:30p-6:30p
Miles Seiver
## Review session schedule

<table>
<thead>
<tr>
<th>Topic</th>
<th>Date</th>
<th>Time</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>assignment 6</strong></td>
<td>today!</td>
<td>now!</td>
<td>here!</td>
</tr>
<tr>
<td><strong>assignment 7</strong></td>
<td>Sun 16 Mar</td>
<td>7p - 8p</td>
<td>Hewlett 200</td>
</tr>
</tbody>
</table>
Interactors
JSlider slide = new JSlider(min, max, initial);
JButton button = new JButton("[button text]");
JTextField field = new JTextField("[field text]" China
JSlider slide = new JSlider(min, max, initial);
JButton button = new JButton("[button text]");
JTextField field = new JTextField("[field text]");
add(interactor, location);
addActionListeners();
public void actionPerformed(ActionEvent e) {

dобавить ActionListener();

public void actionPerformed(ActionEvent e) {

}
public void actionPerformed(ActionEvent e) {
    if (e.getSource() == someInteractorIvar) {
        // ...
    }
}
addActionListeners();

```java
public void actionPerformed(ActionEvent e) {
    if (e.getSource() == someInteractorIvar) {
        // requires instance variables
    }
}
```
```java
public void actionPerformed(ActionEvent e) {

    if (e.getSource() == someInteractorIvar) {
        
    }

    if (e.getActionCommand().equals("[text in button]") ) {
        
    }
}
```
if (e.getSource() == someInteractorIvar) {
  ...
}

if (e.getActionCommand().equals("[text in button]")) {
  ...
}

only works for buttons unless you use
someInteractor.setActionCommand("[action command]")
the assignment
Namesurfer

due Wed, 12 Mar @ 3:15pm
demo
New Java tools you will use!
New Java tools you will use!

HashMap
New Java tools you will use!

HashMap

Array
New Java tools you will use!

HashMap

Array

Interactors
NamesData.txt

Sam 58 69 99 131 168 236 278 380 467 408 466 997
Samantha 0 0 0 0 0 0 0 272 107 26 5 7 63
Samara 0 0 0 0 0 0 0 0 0 886 0
Samir 0 0 0 0 0 0 0 920 0 798 0
Sammie 537 545 351 325 333 396 565 772 930 0 0 0
Sammy 0 887 544 299 202 262 321 395 575 639 755 0
Samson 0 0 0 0 0 0 0 0 0 915 0
Samuel 31 41 46 60 61 71 83 61 52 35 28 32
Sandi 0 0 0 0 704 864 621 695 0 0 0 0
Sandra 0 942 606 50 6 12 11 39 94 168 257 962
NameSurferDatabase

Loads and manages NameSurferEntries
**NameSurferDataBase**

Loads and manages
**NameSurferEntries**

- NameSurferEntry
- NameSurferEntry
- NameSurferEntry
NamesData.txt

Sam 58 69 99 131 168 236 278 380 467 408 466
Samantha 0 0 0 0 0 0 272 107 26 5 7
Samara 0 0 0 0 0 0 0 0 0 0 886
Samir 0 0 0 0 0 0 0 920 0 798
Sammie 537 545 351 325 333 396 565 772 930 0 0
Sanny 0 887 544 299 202 262 321 395 575 639 755
Samson 0 0 0 0 0 0 0 0 0 915
Samuel 31 41 46 60 61 71 83 61 52 35 28
Sandi 0 0 0 704 864 621 695 0 0 0
Sandra 0 942 606 50 6 12 11 39 94 168 257
NamesData.txt

<table>
<thead>
<tr>
<th>Name</th>
<th>Age1</th>
<th>Age2</th>
<th>Age3</th>
<th>Age4</th>
<th>Age5</th>
<th>Age6</th>
<th>Age7</th>
<th>Age8</th>
<th>Age9</th>
<th>Age10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sam</td>
<td>58</td>
<td>69</td>
<td>99</td>
<td>131</td>
<td>168</td>
<td>236</td>
<td>278</td>
<td>380</td>
<td>467</td>
<td>408</td>
</tr>
<tr>
<td>Samantha</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>272</td>
<td>107</td>
<td>26</td>
<td>5</td>
<td>7</td>
</tr>
<tr>
<td>Samara</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>886</td>
</tr>
<tr>
<td>Samir</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>920</td>
<td>0</td>
<td>798</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sammie</td>
<td>537</td>
<td>545</td>
<td>351</td>
<td>325</td>
<td>333</td>
<td>396</td>
<td>565</td>
<td>772</td>
<td>930</td>
<td>0</td>
</tr>
<tr>
<td>Sammy</td>
<td>0</td>
<td>887</td>
<td>544</td>
<td>299</td>
<td>202</td>
<td>262</td>
<td>321</td>
<td>395</td>
<td>575</td>
<td>639</td>
</tr>
<tr>
<td>Samson</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>915</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Samuel</td>
<td>31</td>
<td>41</td>
<td>46</td>
<td>60</td>
<td>61</td>
<td>71</td>
<td>83</td>
<td>61</td>
<td>52</td>
<td>35</td>
</tr>
<tr>
<td>Sandi</td>
<td>0</td>
<td>0</td>
<td>704</td>
<td>864</td>
<td>621</td>
<td>695</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>29</td>
</tr>
<tr>
<td>Sandra</td>
<td>0</td>
<td>942</td>
<td>606</td>
<td>50</td>
<td>6</td>
<td>12</td>
<td>11</td>
<td>39</td>
<td>94</td>
<td>168</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>257</td>
</tr>
<tr>
<td>Name</td>
<td>Numbers</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>------------</td>
<td>---------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sam</td>
<td>58 69 99 131 168 236 278 380 467 408 466</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Samantha</td>
<td>0 0 0 0 0 0 272 107 26 5 7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Samara</td>
<td>0 0 0 0 0 0 0 0 0 886</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Samir</td>
<td>0 0 0 0 0 0 0 920 0 798</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sammie</td>
<td>537 545 351 325 333 396 565 772 930 0 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sammy</td>
<td>0 987 544 299 202 262 321 395 575 639 755</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Samson</td>
<td>0 0 0 0 0 0 0 0 0 915</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Samuel</td>
<td>31 41 46 60 61 71 83 61 52 35 28</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sandi</td>
<td>0 0 0 0 704 864 621 695 0 0 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sandra</td>
<td>0 942 606 50 6 12 11 39 94 168 257</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**NameSurferDatabase**

- Loads and manages
- NameSurferEntries

- NameSurferEntry
- NameSurferEntry
- NameSurferEntry
NamesData.txt

Sam 58 69 99 131 168 236 276 380 467 408 466
Samantha 0 0 0 0 0 0 272 107 26 5 7
Samarra 0 0 0 0 0 0 0 0 886
Samir 0 0 0 0 0 0 920 0 798
Sammie 537 545 351 325 333 396 565 772 930 0 0
Sammy 0 887 544 299 202 262 321 395 575 639 755
Samson 0 0 0 0 0 0 0 0 915
Samuel 31 41 46 60 61 71 83 61 52 35 28
Sandi 0 0 0 704 864 621 695 0 0 0
Sandra 0 942 606 50 6 12 11 39 94 168 257

NameSurferDataBase
- Loads and manages
- NameSurferEntries
  - NameSurferEntry
  - NameSurferEntry
  - NameSurferEntry

NameSurfer
- Main program.
  - Receives user input,
  - reads from the
database, and tells
the graph what to display.
NamesData.txt

Sam 58 69 99 131 168 236 276 380 467 408 466
Samantha 0 0 0 0 0 0 272 107 26 5 7
Samara 0 0 0 0 0 0 0 0 0 886
Samir 0 0 0 0 0 0 0 920 0 798
Sammie 537 545 351 325 333 396 565 772 930 0 0
Sammie 547 544 299 202 262 321 395 575 639 755
Samson 0 0 0 0 0 0 0 0 0 915
Samuel 31 41 46 60 61 71 83 61 52 35 28
Sandi 0 0 0 0 704 864 621 695 0 0 0
Sandra 0 942 606 50 6 12 11 39 94 168 257

NameSurferDataBase

Loads and manages
NameSurferEntries

Asks for
NameSurferEntries

NameSurfer

Main program.
Receives user input,
reads from the
database, and tells the
graph what to display.
## NamesData.txt

<table>
<thead>
<tr>
<th>Name</th>
<th>ID1</th>
<th>ID2</th>
<th>ID3</th>
<th>ID4</th>
<th>ID5</th>
<th>ID6</th>
<th>ID7</th>
<th>ID8</th>
<th>ID9</th>
<th>ID10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sam</td>
<td>58</td>
<td>69</td>
<td>99</td>
<td>131</td>
<td>168</td>
<td>236</td>
<td>276</td>
<td>380</td>
<td>467</td>
<td>408</td>
</tr>
<tr>
<td>Samantha</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>272</td>
<td>107</td>
<td>26</td>
<td>5</td>
</tr>
<tr>
<td>Samara</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>886</td>
<td></td>
</tr>
<tr>
<td>Samir</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>920</td>
<td>0</td>
<td>798</td>
</tr>
<tr>
<td>Sammie</td>
<td>537</td>
<td>545</td>
<td>351</td>
<td>325</td>
<td>333</td>
<td>396</td>
<td>565</td>
<td>772</td>
<td>930</td>
<td>0</td>
</tr>
<tr>
<td>Sammy</td>
<td>0</td>
<td>887</td>
<td>544</td>
<td>299</td>
<td>202</td>
<td>262</td>
<td>321</td>
<td>395</td>
<td>575</td>
<td>639</td>
</tr>
<tr>
<td>Samson</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>915</td>
<td></td>
</tr>
<tr>
<td>Samuel</td>
<td>31</td>
<td>41</td>
<td>46</td>
<td>60</td>
<td>61</td>
<td>71</td>
<td>83</td>
<td>61</td>
<td>52</td>
<td>35</td>
</tr>
<tr>
<td>Sandi</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>704</td>
<td>864</td>
<td>621</td>
<td>695</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Sandra</td>
<td>942</td>
<td>606</td>
<td>50</td>
<td>6</td>
<td>12</td>
<td>11</td>
<td>39</td>
<td>94</td>
<td>168</td>
<td>257</td>
</tr>
</tbody>
</table>

---

**NameSurfer**

Main program. Receives user input, reads from the database, and tells the graph what to display.
NameSurferDatabase

- Loads and manages NameSurferEntries

NameSurfer

- Main program
- Receives user input, reads from the database, and tells the graph what to display

NameSurferGraph

- Graphs NameSurferEntries and keeps track of which entries need to be displayed

NamesData.txt

- Sam 58 69 99 131 168 236 278 380 467 408 466
- Samantha 0 0 0 0 0 0 272 107 26 5 7
- Samara 0 0 0 0 0 0 0 0 0 0 886
- Samir 0 0 0 0 0 0 0 0 20 0 0 798
- Sammie 537 545 351 325 333 396 565 772 930 0 0
- Sammy 0 647 544 299 202 262 321 395 575 639 755
- Samson 0 0 0 0 0 0 0 0 915
- Samuel 31 41 46 60 61 71 83 61 52 35 28
- Sandi 0 0 0 0 704 864 621 695 0 0 0
- Sandra 0 942 606 50 6 12 11 39 94 168 257
- ...
NamesData.txt

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Sam</td>
<td>89</td>
<td>89</td>
<td>89</td>
<td>89</td>
<td>89</td>
<td>89</td>
<td>89</td>
<td>89</td>
<td>89</td>
<td>89</td>
<td>89</td>
</tr>
<tr>
<td>Samara</td>
<td>89</td>
<td>89</td>
<td>89</td>
<td>89</td>
<td>89</td>
<td>89</td>
<td>89</td>
<td>89</td>
<td>89</td>
<td>89</td>
<td>89</td>
</tr>
<tr>
<td>Samir</td>
<td>89</td>
<td>89</td>
<td>89</td>
<td>89</td>
<td>89</td>
<td>89</td>
<td>89</td>
<td>89</td>
<td>89</td>
<td>89</td>
<td>89</td>
</tr>
<tr>
<td>Samnn</td>
<td>89</td>
<td>89</td>
<td>89</td>
<td>89</td>
<td>89</td>
<td>89</td>
<td>89</td>
<td>89</td>
<td>89</td>
<td>89</td>
<td>89</td>
</tr>
<tr>
<td>Samson</td>
<td>89</td>
<td>89</td>
<td>89</td>
<td>89</td>
<td>89</td>
<td>89</td>
<td>89</td>
<td>89</td>
<td>89</td>
<td>89</td>
<td>89</td>
</tr>
<tr>
<td>Samuel</td>
<td>89</td>
<td>89</td>
<td>89</td>
<td>89</td>
<td>89</td>
<td>89</td>
<td>89</td>
<td>89</td>
<td>89</td>
<td>89</td>
<td>89</td>
</tr>
<tr>
<td>Sandi</td>
<td>89</td>
<td>89</td>
<td>89</td>
<td>89</td>
<td>89</td>
<td>89</td>
<td>89</td>
<td>89</td>
<td>89</td>
<td>89</td>
<td>89</td>
</tr>
<tr>
<td>Sandra</td>
<td>89</td>
<td>89</td>
<td>89</td>
<td>89</td>
<td>89</td>
<td>89</td>
<td>89</td>
<td>89</td>
<td>89</td>
<td>89</td>
<td>89</td>
</tr>
</tbody>
</table>

NameSurfer

Main program. Receives user input, reads from the database, and tells the graph what to display.

NameSurferDatabase

Loads and manages NameSurferEntries

NameSurferEntry

NameSurferGraph

Graphs NameSurferEntries and keeps track of which entries need to be displayed.
NameSurferDatabase

NameSurferDatabase
- Loads and manages NameSurferEntries

NameSurfer
- Asks for NameSurferEntries
- Main program.
  - Receives user input, reads from the database, and tells the graph what to display.

NameSurferGraph
-Graphs NameSurferEntries and keeps track of which entries need to be displayed.

NameSurferEntries
- NameSurferEntry
- NameSurferEntry
- NameSurferEntry

NameSurferEntries
- Returns NameSurferEntries

NameSurferEntries
- Hands over NameSurferEntries that need to be displayed.
NameSurferDatabase

**NameSurferDatabase**
- Loads and manages NameSurferEntries
  - NameSurferEntry
  - NameSurferEntry
  - NameSurferEntry

**NameSurfer**
- Main program.
  - Receives user input, reads from the database, and tells the graph what to display.

**NameSurferGraph**
- Graphs NameSurferEntries and keeps track of which entries need to be displayed.

The diagram illustrates the interaction between NameSurferDatabase and NameSurfer, with NameSurferGraph receiving entries to be displayed.
public class NameSurferDataBase implements NameSurferConstants {

    /* Constructor: NameSurferDataBase(filename) */

    /**
     * Creates a new NameSurferDataBase and initializes it using the
     * data in the specified file. The constructor throws an error
     * exception if the requested file does not exist or if an error
     * occurs as the file is being read.
     */

    public NameSurferDataBase(String filename) {
        // You fill this in //
    }

    /* Method: findEntry(name) */

    /**
     * Returns the NameSurferEntry associated with this name, if one
     * exists. If the name does not appear in the database, this
     * method returns null.
     */

    public NameSurferEntry findEntry(String name) {
        // You need to turn this stub into a real implementation //
        return null;
    }
}
public class NameSurferDataBase implements NameSurfererConstants {

    /* Constructor: NameSurferDataBase(filename) */
    public NameSurferDataBase(String filename) {
        // You fill this in //
    }

    /* Method: findEntry(name) */
    public NameSurferEntry findEntry(String name) {
        // You need to turn this stub into a real implementation //
        return null;
    }
}
public class NameSurferDataBase implements NameSurfererConstants {

    /* Constructor: NameSurferDataBase(filename) */
    /**
     * Creates a new NameSurferDataBase and initializes it using the data in the specified file. The constructor throws an error exception if the requested file does not exist or if an error occurs as the file is being read.
     */
    public NameSurferDataBase(String filename) {
        // You fill this in //
    }

    /* Method: findEntry(name) */
    /**
     * Returns the NameSurferEntry associated with this name, if one exists. If the name does not appear in the database, this method returns null.
     */
    public NameSurferEntry findEntry(String name) {
        // You need to turn this stub into a real implementation //
        return null;
    }
}
public class NameSurferDataBase implements NameSurferConstants {

    /* Constructor: NameSurferDataBase(filename) */
    /**
     * Creates a new NameSurferDataBase and initializes it using the 
     * data in the specified file. The constructor throws an error 
     * exception if the requested file does not exist or if an error 
     * occurs as the file is being read.
     */
    public NameSurferDataBase(String filename) {
        // You fill this in //
    }

    /* Method: findEntry(name) */
    /**
     * Returns the NameSurferEntry associated with this name, if one 
     * exists. If the name does not appear in the database, this 
     * method returns null.
     */
    public NameSurferEntry findEntry(String name) {
        // You need to turn this stub into a real implementation //
        return null;
    }
}
public class NameSurferDataBase implements NameSurferConstants {

/**
 * Constructor: NameSurferDataBase(filename) */
/**
 * Creates a new NameSurferDataBase and initializes it using the
 * data in the specified file. The constructor throws an error
 * exception if the requested file does not exist or if an error
 * occurs as the file is being read.
 */

    public NameSurferDataBase(String filename) {
        // You fill this in //
    }

    /**
     * Method: findEntry(name) */
/**
   * Returns the NameSurferEntry associated with this name, if one
   * exists. If the name does not appear in the database, this
   * method returns null.
   */

        public NameSurferEntry findEntry(String name) {
            // You need to turn this stub into a real implementation //
            return null;
    }
}
NameSurferDatabase.java

```
NamesData.txt

Sam 58 69 99 131 168 236 278 380 467 408 466 997
Samanta 0 0 0 0 0 0 272 107 26 5 7 63
Samara 0 0 0 0 0 0 0 0 0 886 0
Samir 0 0 0 0 0 0 0 920 0 798 0
Sammie 537 545 351 325 333 396 565 772 930 0 0 0
Sammy 0 887 544 299 202 262 321 395 575 639 755 0
Samson 0 0 0 0 0 0 0 0 0 915 0
Samuel 31 41 46 60 61 71 83 61 52 35 28 32
Sandi 0 0 0 0 704 864 621 695 0 0 0 0
Sandra 0 942 606 50 6 12 11 39 94 168 257 962
```

`NameSurferDatabase` loads and manages `NameSurferEntries`. It asks for `NameSurferEntries` and returns them.
rank 0 means the name did not appear in the top 1000 names for that year
NameSurferDatabase.java

String line = rd.readLine();

rank 0 means the name did not appear in the top 1000 names for that year
String line = rd.readLine();
NameSurferEntry entry = new NameSurferEntry(line);

rank 0 means the name did not appear in the top 1000 names for that year
String line = rd.readLine();
NameSurferEntry entry = new NameSurferEntry(line);
//Store this NameSurferEntry so it can be retrieved

rank 0 means the name did not appear in the top 1000 names for that year
NameSurferEntry

NameSurferDatabase
Loads and manages NameSurferEntries

NameSurfer
Main program.
Receives user input, reads from the database, and tells the graph what to display.

NameSurferGraph
Graphs NameSurferEntries and keeps track of which entries need to be displayed.

NameSurferGraph
Hands over NameSurferEntries that need to be displayed.

NameSurfer
Returns NameSurferEntries

Asks for NameSurferEntries
NameSurferEntry

NameSurferDataBase
  Loads and manages NameSurferEntries
    NameSurferEntry
    NameSurferEntry
    NameSurferEntry

NameSurfer
  Main program.
  Receives user input, reads from the database, and tells the graph what to display.
  Returns NameSurferEntries

NameSurferGraph
  Graphs NameSurferEntries and keeps track of which entries need to be displayed.
  Hands over NameSurferEntries that need to be displayed.
public class NameSurferEntry implements NameSurferConstants {

    /* Constructor: NameSurferEntry(line) */
    /**
     * Creates a new NameSurferEntry from a data line as it appears
     * in the data file. Each line begins with the name, which is
     * followed by integers giving the rank of that name for each
     * decade.
     */
    public NameSurferEntry(String line) {
        // You fill this in
    }

    /* Method: getName() */
    /**
     * Returns the name associated with this entry.
     */
    public String getName() {
        // You need to turn this stub into a real implementation
        return null;
    }

    /* Method: getRank(decade) */
    /**
     * Returns the rank associated with an entry for a particular
     * decade. The decade value is an integer indicating how many
     * decades have passed since the first year in the database,
     * which is given by the constant START_DECADE. If a name does
     * not appear in a decade, the rank value is 0.
     */
    public int getRank(int decade) {
        // You need to turn this stub into a real implementation
        return 0;
    }

    /* Method: toString() */
    /**
     * Returns a string that makes it easy to see the value of a
     * NameSurferEntry.
     */
    public String toString() {
        // You need to turn this stub into a real implementation
        return "";
    }
}

public class NameSurferEntry implements NameSurferConstants {

    /* Constructor: NameSurferEntry(line) */
    /**
     * Creates a new NameSurferEntry from a data line as it appears
     * in the data file. Each line begins with the name, which is
     * followed by integers giving the rank of that name for each
     * decade.
     */
    public NameSurferEntry(String line) {
        // You fill this in
    }

    /* Method: getName() */
    /**
     * Returns the name associated with this entry.
     */
    public String getName() {
        // You need to turn this stub into a real implementation
        return null;
    }

    /* Method: getRank(decade) */
    /**
     * Returns the rank associated with an entry for a particular
     * decade. The decade value is an integer indicating how many
     * decades have passed since the first year in the database,
     * which is given by the constant START_DECADE. If a name does
     * not appear in a decade, the rank value is 0.
     */
    public int getRank(int decade) {
        // You need to turn this stub into a real implementation
        return 0;
    }

    /* Method: toString() */
    /**
     * Returns a string that makes it easy to see the value of a
     * NameSurferEntry.
     */
    public String toString() {
        // You need to turn this stub into a real implementation
        return ""
    }
}
public class NameSurferEntry implements NameSurferConstants {

    /* Constructor: NameSurferEntry(line) */
    /**
     * Creates a new NameSurferEntry from a data line as it appears
     * in the data file. Each line begins with the name, which is
     * followed by integers giving the rank of that name for each
     * decade.
     */
    public NameSurferEntry(String line) {
        // You fill this in
    }

    /* Method: getName() */
    /**
     * Returns the name associated with this entry.
     */
    public String getName() {
        // You need to turn this stub into a real implementation
        return null;
    }

    /* Method: getRank(decade) */
    /**
     * Returns the rank associated with an entry for a particular
     * decade. The decade value is an integer indicating how many
     * decades have passed since the first year in the database,
     * which is given by the constant START_DECADE. If a name does
     * not appear in a decade, the rank value is 0.
     */
    public int getRank(int decade) {
        // You need to turn this stub into a real implementation
        return 0;
    }

    /* Method: toString() */
    /**
     * Returns a string that makes it easy to see the value of a
     * NameSurferEntry.
     */
    public String toString() {
        // You need to turn this stub into a real implementation
        return ""
    }
}
public class NameSurferEntry implements NameSurferConstants {

    /* Constructor: NameSurferEntry(line) */
    /**
     * Creates a new NameSurferEntry from a data line as it appears
     * in the data file. Each line begins with the name, which is
     * followed by integers giving the rank of that name for each
     * decade.
     */
    public NameSurferEntry(String line) {
        // You fill this in //
    }

    /* Method: getName() */
    /**
     * Returns the name associated with this entry.
     */
    public String getName() {
        // You need to turn this stub into a real implementation //
        return null;
    }

    /* Method: getRank(decade) */
    /**
     * Returns the rank associated with an entry for a particular
     * decade. The decade value is an integer indicating how many
     * decades have passed since the first year in the database,
     * which is given by the constant START_DECADE. If a name does
     * not appear in a decade, the rank value is 0.
     */
    public int getRank(int decade) {
        // You need to turn this stub into a real implementation //
        return 0;
    }

    /* Method: toString() */
    /**
     * Returns a string that makes it easy to see the value of a
     * NameSurferEntry.
     */
    public String toString() {
        // You need to turn this stub into a real implementation //
        return "";
    }
}
public class NameSurferEntry implements NameSurferConstants {

    /* Constructor: NameSurferEntry(line) */
    /**
     * Creates a new NameSurferEntry from a data line as it appears
     * in the data file. Each line begins with the name, which is
     * followed by integers giving the rank of that name for each
     * decade.
     */
    public NameSurferEntry(String line) {
        // You fill this in //
    }

    /* Method: getName() */
    /**
     * Returns the name associated with this entry.
     */
    public String getName() {
        // You need to turn this stub into a real implementation //
        return null;
    }

    /* Method: getRank(decade) */
    /**
     * Returns the rank associated with an entry for a particular
     * decade. The decade value is an integer indicating how many
     * decades have passed since the first year in the database,
     * which is given by the constant START_DECADE. If a name does
     * not appear in a decade, the rank value is 0.
     */
    public int getRank(int decade) {
        // You need to turn this stub into a real implementation //
        return 0;
    }

    /* Method: toString() */
    /**
     * Returns a string that makes it easy to see the value of a
     * NameSurferEntry.
     */
    public String toString() {
        // You need to turn this stub into a real implementation //
        return "";
    }
}
String getName()
int getRank(int decade)
String toString()
String getName()
int getRank(int decade)
String toString()
String getName()
int getRank(int decade)
String toString()
NameSurferEntry.java

“Sam 58 69 99 131 168 236 278 380 467 408 466 997”

String getName()
int getRank(int decade)
String toString()
NameSurferEntry.java

“Sam 58 69 99 131 168 236 278 380 467 408 466 997”

rankData

58 69 99 131 168 236 278 380 467 408 466 997

name

“Sam”

String getName()
int getRank(int decade)
String toString()
NameSurferGraph

**NameSurferDatabase**
- Loads and manages NameSurferEntries
  - NameSurferEntry
  - NameSurferEntry
  - NameSurferEntry

**NameSurfer**
- Main program
  - Receives user input, reads from the database, and tells the graph what to display.
  - Returns NameSurferEntries

**NameSurferGraph**
- Graphs NameSurferEntries and keeps track of which entries need to be displayed.
- Hands over NameSurferEntries that need to be displayed.
NameSurferGraph

**NameSurferDataBase**
- Loads and manages NameSurferEntries
  - NameSurferEntry
  - NameSurferEntry
  - NameSurferEntry

**NameSurfer**
- Main program. Receives user input, reads from the database, and tells the graph what to display.

**NameSurferGraph**
- Graphs NameSurferEntries and keeps track of which entries need to be displayed.
public class NameSurferGraph extends GCanvas
    implements NameSurferConstants, ComponentListener {
/**
 * Creates a new NameSurferGraph object that displays the data.
 */
public NameSurferGraph() {
    addComponentListener(this);
    // You fill in the rest //
}

/**
 * Clears the list of name surfer entries stored inside this class.
 */
public void clear() {
    // You fill this in //
}

/* Method: addEntry(entry) */
/**
 * Adds a new NameSurferEntry to the list of entries on the display.
 * Note that this method does not actually draw the graph, but
 * simply stores the entry; the graph is drawn by calling update.
 */
public void addEntry(NameSurferEntry entry) {
    // You fill this in //
}

/**
 * Updates the display image by deleting all the graphical objects
 * from the canvas and then reassembling the display according to
 * the list of entries. Your application must call update after
 * calling either clear or addEntry; update is also called whenever
 * the size of the canvas changes.
 */
public void update() {
    // You fill this in //
}

/* Implementation of the ComponentListener interface */
public void componentHidden(ComparatorEvent e) { }
public void componentMoved(ComparatorEvent e) { }
public void componentResized(ComparatorEvent e) { update(); }
public void componentShown(ComparatorEvent e) { }
public class NameSurferGraph extends GCanvas implements NameSurferConstants, ComponentListener {
    /**
     * Creates a new NameSurferGraph object that displays the data.
     */
    public NameSurferGraph() {
        addComponentListener(this);
        // You fill in the rest
    }

    /**
     * Clears the list of name surfer entries stored inside this class.
     */
    public void clear() {
        // You fill this in
    }

    /* Method: addEntry(entry) */
    /**
     * Adds a new NameSurferEntry to the list of entries on the display.
     * Note that this method does not actually draw the graph, but
     * simply stores the entry; the graph is drawn by calling update.
     */
    public void addEntry(NameSurferEntry entry) {
        // You fill this in
    }

    /**
     * Updates the display image by deleting all the graphical objects
     * from the canvas and then reassembling the display according to
     * the list of entries. Your application must call update after
     * calling either clear or addEntry; update is also called whenever
     * the size of the canvas changes.
     */
    public void update() {
        // You fill this in
    }

    /* Implementation of the ComponentListener interface */
    public void componentHidden(ComponentEvent e) { }
    public void componentMoved(ComponentEvent e) { }
    public void componentResized(ComponentEvent e) { update(); }
    public void componentShown(ComponentEvent e) { }
}
public class NameSurferGraph extends GCanvas
    implements NameSurferConstants, ComponentListener {

    /**
     * Creates a new NameSurferGraph object that displays the data.
     */
    public NameSurferGraph() {
        addComponentListener(this);
        // You fill in the rest //
    }

    /**
     * Clears the list of name surfer entries stored inside this class.
     */
    public void clear() {
        // You fill this in //
    }

    /* Method: addEntry(entry) */
    /*
     * Adds a new NameSurferEntry to the list of entries on the display.
     * Note that this method does not actually draw the graph, but
     * simply stores the entry; the graph is drawn by calling update.
     */
    public void addEntry(NameSurferEntry entry) {
        // You fill this in //
    }

    /**
     * Updates the display image by deleting all the graphical objects
     * from the canvas and then reassembling the display according to
     * the list of entries. Your application must call update after
     * calling either clear or addEntry; update is also called whenever
     * the size of the canvas changes.
     */
    public void update() {
        // You fill this in //
    }

    /* Implementation of the ComponentListener interface */
    public void componentHidden( ComponentEvent e ) { } 
    public void componentMoved( ComponentEvent e ) { }
    public void componentResized( ComponentEvent e ) { update(); } 
    public void componentShown( ComponentEvent e ) { } 
}
public class NameSurferGraph extends GCanvas
    implements NameSurferConstants, ComponentListener {
    /**
     * Creates a new NameSurferGraph object that displays the data.
     */
    public NameSurferGraph() {
        addComponentListener(this);
        // You fill in the rest //
    }

    /**
     * Clears the list of name surfer entries stored inside this class.
     */
    public void clear() {
        // You fill this in //
    }

    /* Method: addEntry(entry) */
    /**
     * Adds a new NameSurferEntry to the list of entries on the display.
     * Note that this method does not actually draw the graph, but
     * simply stores the entry; the graph is drawn by calling update.
     */
    public void addEntry(NameSurferEntry entry) {
        // You fill this in //
    }

    /**
     * Updates the display image by deleting all the graphical objects
     * from the canvas and then reassembling the display according to
     * the list of entries. Your application must call update after
     * calling either clear or addEntry; update is also called whenever
     * the size of the canvas changes.
     */
    public void update() {
        // You fill this in //
    }

    /* Implementation of the ComponentListener interface */
    public void componentHidden(ComponentEvent e) { }
    public void componentMoved(ComponentEvent e) { }
    public void componentResized(ComponentEvent e) { update(); }
    public void componentShown(ComponentEvent e) { }
}
public class NameSurferGraph extends GCanvas implements NameSurferConstants, ComponentListener {
    /**
     * Creates a new NameSurferGraph object that displays the data.
     */
    public NameSurferGraph() {
        addComponentListener(this);
        // You fill in the rest //
    }

    /**
     * Clears the list of name surfer entries stored inside this class.
     */
    public void clear() {
        // You fill this in //
    }

    /* Method: addEntry(entry) */
    /**
     * Adds a new NameSurferEntry to the list of entries on the display.
     * Note that this method does not actually draw the graph, but
     * simply stores the entry; the graph is drawn by calling update.
     */
    public void addEntry(NameSurferEntry entry) {
        // You fill this in //
    }

    ///
    * Updates the display image by deleting all the graphical objects
    * from the canvas and then reassembling the display according to
    * the list of entries. Your application must call update after
    * calling either clear or addEntry; update is also called whenever
    * the size of the canvas changes.
    ///
    public void update() {
        // You fill this in //
    }

    /* Implementation of the ComponentListener interface */
    public void componentHidden(ComponentEvent e) { }
    public void componentMoved(ComponentEvent e) { }
    public void componentResized(ComponentEvent e) { update(); }
    public void componentShown(ComponentEvent e) { }
}
Name: Samantha

entries

Sam NameSurferEntry
entries

- Sam NameSurferEntry
- Samantha NameSurferEntry
Name: Samantha

Entries:
- Sam NameSurferEntry
  (58 69 99 131 168 236 278 380 467 408 466 997)
- Samantha NameSurferEntry
  (0 0 0 0 0 0 272 107 26 5 7 63)
entries

Sam NameSurferEntry
(58 69 99 131 168 236 278 380 467 408 466 997)

Samantha NameSurferEntry
(0 0 0 0 0 0 272 107 26 5 7 63)

from NamesurferConstants

/** The first decade in the database */
public static final int START_DECADE = 1900;

/** The number of decades */
public static final int NDECADES = 12;

/** The maximum rank in the database */
public static final int MAX_RANK = 1000;

/** The number of pixels to reserve at the top and bottom */
public static final int GRAPH_MARGIN_SIZE = 20;
entries

<table>
<thead>
<tr>
<th>Name</th>
<th>NameSurferEntry</th>
<th>(58 69 99 131 168 236 278 380 467 408 466 997)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sam</td>
<td>NameSurferEntry</td>
<td></td>
</tr>
<tr>
<td>Samantha</td>
<td>NameSurferEntry</td>
<td></td>
</tr>
</tbody>
</table>

```java
/** The first decade in the database */
public static final int START_DECADE = 1900;

/** The number of decades */
public static final int NDECADES = 12;

/** The maximum rank in the database */
public static final int MAX_RANK = 1000;

/** The number of pixels to reserve at the top and bottom */
public static final int GRAPH_MARGIN_SIZE = 20;
```
getNameSurferEntry

entries

Sam NameSurferEntry
(58 69 99 131 168 236 278 380 467 408 466 997)

Samantha NameSurferEntry
(0 0 0 0 0 272 107 26 5 7 63)

from NameSurferConstants

/**< The first decade in the database */
public static final int START_DECade = 1900;

/**< The number of decades */
public static final int NDECADES = 12;

/**< The maximum rank in the database */
public static final int MAX_RANK = 1000;

/**< The number of pixels to reserve at the top and bottom */
public static final int GRAPH_MARGIN_SIZE = 20;
entries

<table>
<thead>
<tr>
<th>Name</th>
<th>NameSurferEntry</th>
<th>(58 69 99 131 168 236 278 380 408 466 997)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sam</td>
<td>NameSurferEntry</td>
<td>(58 69 99 131 168 236 278 380 408 466 997)</td>
</tr>
<tr>
<td>Samantha</td>
<td>NameSurferEntry</td>
<td>(0 0 0 0 0 0 272 107 26 5 7 63)</td>
</tr>
</tbody>
</table>

```java
/** The first decade in the database */
public static final int START_DECade = 1900;

/** The number of decades */
public static final int NDECADEs = 12;

/** The maximum rank in the database */
public static final int MAX_RANK = 1000;

/** The number of pixels to reserve at the top and bottom */
public static final int GRAPH_MARGIN_SIZE = 20;
```
Sam NameSurferEntry
(58 69 99 131 168 236 278 380 408 466 997)

Samantha NameSurferEntry
(0 0 0 0 0 0 272 107 26 5 7 63)

/** The first decade in the database */
public static final int START_DECade = 1900;

/** The number of decades */
public static final int NDECADES = 12;

/** The maximum rank in the database */
public static final int MAX_RANK = 1000;

/** The number of pixels to reserve at the top and bottom */
public static final int GRAPH_MARGIN_SIZE = 20;
entries

<table>
<thead>
<tr>
<th>Name</th>
<th>Sam</th>
<th>NameSurferEntry</th>
<th>(58 69 99 131 168 236 278 380 467 408 466 997)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Samantha</td>
<td>Name</td>
<td>NameSurferEntry</td>
<td>(0 0 0 0 0 0 272 107 26 5 7 63)</td>
</tr>
</tbody>
</table>

from NameSurferConstants

```java
/** The first decade in the database */
public static final int START_DECADE = 1900;

/** The number of decades */
public static final int NDECADES = 12;

/** The maximum rank in the database */
public static final int MAX_RANK = 1000;

/** The number of pixels to reserve at the top and bottom */
public static final int GRAPH_MARGIN_SIZE = 20;
```
entries

Sam NameSurferEntry (58 69 99 131 168 236 278 380 467 408 466 997)

Samantha NameSurferEntry (0 0 0 0 0 272 107 26 5 7 63)

```java
/** The first decade in the database */
public static final int START_DECade = 1900;

/** The number of decades */
public static final int NDECADES = 12;

/** The maximum rank in the database */
public static final int MAX_RANK = 1000;

/** The number of pixels to reserve at the top and bottom */
public static final int GRAPH_MARGIN_SIZE = 20;
```

from NameSurferConstants
entries

Sam NameSurferEntry
(58 69 99 131 168 236 278 380 408 466 997)

Samantha NameSurferEntry
(0 0 0 0 0 0 272 107 26 5 7 63)

from NameSurferConstants

/** The first decade in the database */
public static final int START_DECADE = 1900;

/** The number of decades */
public static final int NDECADES = 12;

/** The maximum rank in the database */
public static final int MAX_RANK = 1000;

/** The number of pixels to reserve at the top and bottom */
public static final int GRAPH_MARGIN_SIZE = 20;
entries

<table>
<thead>
<tr>
<th>Name</th>
<th>NameSurferEntry</th>
<th>(58 69 99 131 168 236 278 380 408 466 997)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sam</td>
<td>NameSurferEntry</td>
<td>(58 69 99 131 168 236 278 380 408 466 997)</td>
</tr>
<tr>
<td>Samantha</td>
<td>NameSurferEntry</td>
<td>(0 0 0 0 0 0 272 107 26 5 7 63)</td>
</tr>
</tbody>
</table>

```java
/** The first decade in the database */
public static final int START_DECADEx = 1900;

/** The number of decades */
public static final int NDECADEx = 12;

/** The maximum rank in the database */
public static final int MAX_RANK = 1000;

/** The number of pixels to reserve at the top and bottom */
public static final int GRAPH_MARGIN_SIZE = 20;
```
entries

Sam NameSurferEntry
(58 69 99 131 168 236 278 380 467 408 466 997)

Samantha NameSurferEntry
(0 0 0 0 0 0 272 107 26 5 7 63)

```java
/** The first decade in the database */
public static final int START_DECADE = 1900;

/** The number of decades */
public static final int NDECADES = 12;

/** The maximum rank in the database */
public static final int MAX_RANK = 1000;

/** The number of pixels to reserve at the top and bottom */
public static final int GRAPH_MARGIN_SIZE = 20;
```
<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sam NameSurferEntry</td>
<td>(58 69 99 131 168 236 278 380 467 408 466 997)</td>
<td>entries</td>
</tr>
<tr>
<td>Samantha NameSurferEntry</td>
<td>(0 0 0 0 0 0 272 107 26 5 7 63)</td>
<td>from NameSurferConstants</td>
</tr>
</tbody>
</table>

```java
/** The first decade in the database */
public static final int START_DECade = 1900;

/** The number of decades */
public static final int NDECADES = 12;

/** The maximum rank in the database */
public static final int MAX_RANK = 1000;

/** The number of pixels to reserve at the top and bottom */
public static final int GRAPH_MARGIN_SIZE = 20;
```
**NameSurferConstants**

```java
/** The first decade in the database */
public static final int START_DECADE = 1900;

/** The number of decades */
public static final int NDECADES = 12;

/** The maximum rank in the database */
public static final int MAX_RANK = 1000;

/** The number of pixels to reserve at the top and bottom */
public static final int GRAPH_MARGIN_SIZE = 20;
```
**entries**

<table>
<thead>
<tr>
<th>Sam NameSurferEntry</th>
<th>(58 69 99 131 168 236 278 380 467 408 466 997)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Samantha NameSurferEntry</td>
<td>(0 0 0 0 0 0 272 107 26 5 7 63)</td>
</tr>
</tbody>
</table>

```java
/** The first decade in the database */
public static final int START_DECade = 1900;

/** The number of decades */
public static final int N_DECades = 12;

/** The maximum rank in the database */
public static final int MAX_RANK = 1000;

/** The number of pixels to reserve at the top and bottom */
public static final int GRAPH_MARGIN_SIZE = 20;
```
entries

Sam NameSurferEntry
(58 69 99 131 168 236 278 380 467 408 466 997)

Samantha NameSurferEntry
(0 0 0 0 0 0 272 107 26 5 7 63)

```java
/** The first decade in the database */
public static final int START_DECade = 1900;

/** The number of decades */
public static final int NDECADES = 12;

/** The maximum rank in the database */
public static final int MAX_RANK = 1000;

/** The number of pixels to reserve at the top and bottom */
public static final int GRAPH_MARGIN_SIZE = 20;
```
entries

Sam NameSurferEntry
(58 69 99 131 168 236 278 380 467 408 466 997)

Samantha NameSurferEntry
(0 0 0 0 0 0 272 107 26 5 7 63)

/** The first decade in the database */
public static final int START_DECade = 1900;

/** The number of decades */
public static final int NDECADES = 12;

/** The maximum rank in the database */
public static final int MAX_RANK = 1000;

/** The number of pixels to reserve at the top and bottom */
public static final int GRAPH_MARGIN_SIZE = 20;
entries

Sam NameSurferEntry
(58 69 99 131 168 236 278 380 467 408 466 997)

Samantha NameSurferEntry
(0 0 0 0 0 0 272 107 26 5 7 63)

from NameSurferConstants

/** The first decade in the database */
public static final int START_DECADE = 1900;

/** The number of decades */
public static final int NDECADES = 12;

/** The maximum rank in the database */
public static final int MAX_RANK = 1000;

/** The number of pixels to reserve at the top and bottom */
public static final int GRAPH_MARGIN_SIZE = 20;
entries

Sam NameSurferEntry
(58 69 99 131 168 236 278 380 467 408 466 997)

Samantha NameSurferEntry
(0 0 0 0 0 0 272 107 26 5 7 63)

from NameSurferConstants

```java
/** The first decade in the database */
public static final int START_DECade = 1900;

/** The number of decades */
public static final int NDECADES = 12;

/** The maximum rank in the database */
public static final int MAX_RANK = 1000;

/** The number of pixels to reserve at the top and bottom */
public static final int GRAPH_MARGIN_SIZE = 20;
```
entries

**Sam** NameSurferEntry
(58 69 99 131 168 236 278 380 467 408 466 997)

**Samantha** NameSurferEntry
(0 0 0 0 0 0 272 107 26 5 7 63)

```java
/** The first decade in the database */
public static final int START_DECade = 1900;

/** The number of decades */
public static final int NDECAses = 12;

/** The maximum rank in the database */
public static final int MAX_RANK = 1000;

/** The number of pixels to reserve at the top and bottom */
public static final int GRAPH_MARGIN_SIZE = 20;
```
entries

Sam NameSurferEntry
(58 69 99 131 168 236 278 380 467 408 466 997)

Samantha NameSurferEntry
(0 0 0 0 0 0 272 107 26 5 7 63)

```java
/** The first decade in the database */
public static final int START_DECADE = 1900;

/** The number of decades */
public static final int NDECADES = 12;

/** The maximum rank in the database */
public static final int MAX_RANK = 1000;

/** The number of pixels to reserve at the top and bottom */
public static final int GRAPH_MARGIN_SIZE = 20;
```
entries

<table>
<thead>
<tr>
<th>Name</th>
<th>NameSurferEntry</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sam</td>
<td>(58 69 99 131 168 236 278 380 467 408 466 997)</td>
</tr>
<tr>
<td>Samantha</td>
<td>(0 0 0 0 0 0 272 107 26 5 7 63)</td>
</tr>
</tbody>
</table>

from NameSurferConstants

```java
/** The first decade in the database */
public static final int START_DECade = 1900;

/** The number of decades */
public static final int NDECades = 12;

/** The maximum rank in the database */
public static final int MAX_RANK = 1000;

/** The number of pixels to reserve at the top and bottom */
public static final int GRAPH_MARGIN_SIZE = 20;
```
NameSurferConstants
public interface NameSurferConstants {

    /** The width of the application window */
    public static final int APPLICATION_WIDTH = 800;

    /** The height of the application window */
    public static final int APPLICATION_HEIGHT = 600;

    /** The name of the file containing the data */
    public static final String NAMES_DATA_FILE = "names-data.txt";

    /** The first decade in the database */
    public static final int START_DECADE = 1900;

    /** The number of decades */
    public static final int NDECADES = 12;

    /** The maximum rank in the database */
    public static final int MAX_RANK = 1000;

    /** The number of pixels to reserve at the top and bottom */
    public static final int GRAPH_MARGIN_SIZE = 20;

}
NameSurfer

NameSurferDatabase

- Loads and manages NameSurferEntries
  - NameSurferEntry
  - NameSurferEntry
  - NameSurferEntry

NameSurfer

- Main program.
  - Receives user input, reads from the database, and tells the graph what to display.
- Returns NameSurferEntries

Hands over NameSurferEntries that need to be displayed.

NameSurferGraph

- Graphs NameSurferEntries and keeps track of which entries need to be displayed.
NameSurfer

NameSurferDatabase
- Loads and manages NameSurferEntries
  - NameSurferEntry
  - NameSurferEntry
  - NameSurferEntry

Asks for NameSurferEntries

NameSurfer
- Main program. Receives user input, reads from the database, and tells the graph what to display.

Returns NameSurferEntries

NameSurferGraph
- Graphs NameSurferEntries and keeps track of which entries need to be displayed.

Hands over NameSurferEntries that need to be displayed.
public class NameSurfer extends Program implements NameSurferConstants {

    /* Method: init() */
    /**
     * This method has the responsibility for reading in the database
     * and initializing the interactors at the top of the window.
     */
    public void init() {
        // You fill this in, along with any helper methods //
    }

    /* Method: actionPerformed(e) */
    /**
     * This class is responsible for detecting when the buttons are
     * clicked, so you will have to define a method to respond to
     * button actions.
     */
    public void actionPerformed(ActionEvent e) {
        // You fill this in //
    }
}
To add the graph to the screen

private NameSurferGraph graph;

graph = new NameSurferGraph();
add(graph);
To add the graph to the screen

```java
private NameSurferGraph graph;

graph = new NameSurferGraph();
add(graph);
```
NameSurferDatabase

Loads and manages NameSurferEntries
NameSurferDatabase

Loads and manages NameSurferEntries

NameSurferEntry

NameSurferEntry

NameSurferEntry
<table>
<thead>
<tr>
<th>Name</th>
<th>SurferID</th>
<th>SurferFirst</th>
<th>SurferLast</th>
<th>SurferAge</th>
<th>SurferGender</th>
<th>SurferCity</th>
<th>SurferState</th>
<th>SurferPhone</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sam</td>
<td>58</td>
<td>69</td>
<td>99</td>
<td>131</td>
<td>168</td>
<td>236</td>
<td>278</td>
<td>380</td>
</tr>
<tr>
<td>Samm</td>
<td>537</td>
<td>545</td>
<td>351</td>
<td>325</td>
<td>333</td>
<td>396</td>
<td>565</td>
<td>772</td>
</tr>
<tr>
<td>Sammy</td>
<td>008</td>
<td>544</td>
<td>299</td>
<td>202</td>
<td>262</td>
<td>321</td>
<td>395</td>
<td>575</td>
</tr>
<tr>
<td>Samuel</td>
<td>31</td>
<td>41</td>
<td>46</td>
<td>60</td>
<td>61</td>
<td>71</td>
<td>83</td>
<td>61</td>
</tr>
<tr>
<td>Sandi</td>
<td>000</td>
<td>000</td>
<td>704</td>
<td>864</td>
<td>621</td>
<td>695</td>
<td>000</td>
<td>000</td>
</tr>
<tr>
<td>Sandra</td>
<td>0942</td>
<td>606</td>
<td>50</td>
<td>6</td>
<td>12</td>
<td>11</td>
<td>39</td>
<td>94</td>
</tr>
</tbody>
</table>

**NameSurferDatabase**

- Loads and manages
- **NameSurferEntries**
  - NameSurferEntry
  - NameSurferEntry
  - NameSurferEntry
<table>
<thead>
<tr>
<th>Name</th>
<th>SurferID</th>
<th>SurferID</th>
<th>SurferID</th>
<th>SurferID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sam</td>
<td>58 69 99</td>
<td>131 168 236 278</td>
<td>380 467 408 466</td>
<td></td>
</tr>
<tr>
<td>Samantha</td>
<td>0 0 0 0 0 0 0 272 107 26 5 7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Samara</td>
<td>0 0 0 0 0 0 0 0 0 886</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Samir</td>
<td>0 0 0 0 0 0 0 0 0 0 920 978</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sammie</td>
<td>537 545 351 325 333 396 565 772 930 0 0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sammy</td>
<td>0 887 544 299 202 262 321 395 575 639 755</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Samson</td>
<td>0 0 0 0 0 0 0 0 0 0 915</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Samuel</td>
<td>31 41 46 46 60 61 71 83 61 52 35 28</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sandi</td>
<td>0 0 0 0 0 704 864 621 695 0 0 0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sandra</td>
<td>0 942 606 50 6 12 11 39 94 168 257</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

NameSurferDatabase

Loads and manages NameSurferEntries

- NameSurferEntry
- NameSurferEntry
- NameSurferEntry
<table>
<thead>
<tr>
<th>Name</th>
<th>Score1</th>
<th>Score2</th>
<th>Score3</th>
<th>Score4</th>
<th>Score5</th>
<th>Score6</th>
<th>Score7</th>
<th>Score8</th>
<th>Score9</th>
<th>Score10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sam</td>
<td>58</td>
<td>69</td>
<td>99</td>
<td>131</td>
<td>168</td>
<td>236</td>
<td>278</td>
<td>380</td>
<td>467</td>
<td>408</td>
</tr>
<tr>
<td>Samantha</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>272</td>
<td>107</td>
<td>26</td>
<td>5</td>
<td>7</td>
</tr>
<tr>
<td>Samara</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>886</td>
<td></td>
</tr>
<tr>
<td>Samir</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>920</td>
<td>0</td>
<td>798</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sammie</td>
<td>537</td>
<td>545</td>
<td>351</td>
<td>325</td>
<td>333</td>
<td>396</td>
<td>565</td>
<td>772</td>
<td>930</td>
<td>0</td>
</tr>
<tr>
<td>Sammy</td>
<td>0</td>
<td>887</td>
<td>544</td>
<td>299</td>
<td>202</td>
<td>262</td>
<td>321</td>
<td>395</td>
<td>575</td>
<td>639</td>
</tr>
<tr>
<td>Samson</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>915</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Samuel</td>
<td>31</td>
<td>41</td>
<td>46</td>
<td>60</td>
<td>61</td>
<td>71</td>
<td>83</td>
<td>61</td>
<td>52</td>
<td>35</td>
</tr>
<tr>
<td>Sandi</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>704</td>
<td>864</td>
<td>621</td>
<td>695</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Sandra</td>
<td>942</td>
<td>606</td>
<td>50</td>
<td>6</td>
<td>12</td>
<td>11</td>
<td>39</td>
<td>94</td>
<td>168</td>
<td>257</td>
</tr>
</tbody>
</table>

NameSurferData.txt

NameSurferDataBase
Loads and manages NameSurferEntry

NameSurferEntry

NameSurferEntry

NameSurferEntry
NameSurfer

Main program. Receives user input, reads from the database, and tells the graph what to display.
NamesData.txt

Sam 58 69 99 131 168 236 276 380 467 408 466
Samantha 0 0 0 0 0 0 272 107 26 5 7
Samara 0 0 0 0 0 0 0 0 886
Samir 0 0 0 0 0 0 920 0 798
Sammie 537 545 351 325 333 396 565 772 930 0 0
Sammie 0 887 544 299 202 262 321 395 575 639 755
Samson 0 0 0 0 0 0 0 0 915
Samuel 31 41 46 60 61 71 83 61 52 35 28
Sandi 0 0 0 0 704 864 621 695 0 0 0
Sandra 0 942 606 50 6 12 11 39 94 168 257

NameSurferDataBase

Loads and manages
NameSurferEntries

NameSurfer

Main program. Receives user input, reads from the database, and tells the graph what to display.
NamesData.txt

Sam 58 69 99 131 168 236 278 380 467 408 466
Samantha 0 0 0 0 0 0 272 107 26 5 7
Samara 0 0 0 0 0 0 0 0 886
Samir 0 0 0 0 0 0 0 920 0 798
Sammie 537 545 351 325 333 396 565 772 930 0 0
Sammy 0 887 544 299 202 262 321 395 575 639 755
Samson 0 0 0 0 0 0 0 0 915
Samuel 31 41 46 60 61 71 83 61 52 35 28
Sandi 0 0 0 0 704 864 621 695 0 0 0
Sandra 0 942 606 50 6 12 11 39 94 168 257
NamesData.txt

Sam 58 69 99 131 168 236 278 380 467 408 466
Samantha 0 0 0 0 0 0 0 0 272 107 26 5 7
Samara 0 0 0 0 0 0 0 0 0 886
Samir 0 0 0 0 0 0 0 0 920 0 798
Sammie 537 545 351 325 333 396 565 772 930 0 0
Sammy 0 887 544 299 202 262 321 395 575 639 755
Samson 0 0 0 0 0 0 0 0 0 915
Samuel 31 41 46 60 61 71 83 61 52 35 28
Sandi 0 0 0 0 704 864 621 695 0 0 0
Sandra 0 942 606 50 6 12 11 39 94 168 257

NameSurferDatabase

Loads and manages NameSurferEntries

NameSurfer

Main program. Receives user input, reads from the database, and tells the graph what to display.

NameSurferGraph

Graphs NameSurferEntries and keeps track of which entries need to be displayed.

Hands over NameSurferEntries that need to be displayed.
NamesData.txt
Sam 58 69 99 131 168 236 276 380 467 408 466
Samanta 0 0 0 0 0 272 107 26 57
Samara 0 0 0 0 0 0 0 0 0 886
Samir 0 0 0 0 0 0 0 0 920 0 798
Sammie 537 545 351 325 333 396 565 772 930 0 0
Samm 0 887 544 299 202 262 321 395 575 639 755
Samson 0 0 0 0 0 0 0 0 915
Samuel 31 41 46 60 61 71 83 61 52 35 28
Sandi 0 0 0 0 704 864 621 695 0 0 0
Sandra 942 606 50 6 12 11 39 94 168 257

NameSurferDatabase
- Loads and manages NameSurferEntries
- Main program.
  - Receives user input, reads from the database, and tells the graph what to display.
  - Hands over NameSurferEntries that need to be displayed.
  - Graphs NameSurferEntries and keeps track of which entries need to be displayed.

NameSurferGraph
Tricky parts
Tricky parts

• null pointer exceptions
Tricky parts

- null pointer exceptions
- OutOfBoundsException
Tricky parts

- null pointer exceptions
- OutOfBoundsException
- off-by-one drawing
Tricky parts

- null pointer exceptions
- OutOfBoundsException
- off-by-one drawing
- resizing using update()