Group Members
Eliot Cole
Kelvin Wilch
Bill Ngoga
Liam McGibbon

Writeup for Song and SongCollection Classes

This code defines two classes, Song and SongCollection, used to represent and manage a collection of songs.

Song Class

- Represents a single song with its artist, title, and lyrics.
- **Fields:**
  - `artist`: String containing the artist name.
  - `title`: String containing the song title.
  - `lyrics`: String containing the song lyrics with line feeds included.
- **Methods:**
  - **Constructor (Parameterized)**: `Song(String artist, String title, String lyrics)` - Creates a new Song object with the provided artist, title, and lyrics.
  - `getArtist()`: Returns the artist name.
  - `getTitle()`: Returns the song title.
  - `getLyrics()`: Returns the song lyrics.
  - `toString()`: Returns a formatted string with artist and title in the form "artist, "title"."
  - `compareTo(Object that)`: Implements the Comparable interface and defines the sorting order for songs. It sorts by artist first (case-insensitive) and then by title (case-insensitive). This ensures songs from the same artist are grouped together alphabetically.

SongCollection Class

- Represents a collection of Song objects.
- **Field:**
  - `songs`: An array of Song objects holding the song collection.
- **Constructor:**
  - `SongCollection(String filename)`: Takes a filename as input and reads song data from it. It throws a FileNotFoundException if the file is not found.
    - Uses a `try-catch` block to handle the file reading process.
    - Creates a temporary `ArrayList` to store songs during file processing.
    - Opens the file using `FileInputStream` and creates a `Scanner` object to read the file line by line.
    - Sets the scanner delimiter to quotes (") for proper parsing.
    - Iterates through the file, skipping the first token (assuming it's a quote).
    - Reads artist, title, and lyrics within quotes.
    - Creates a new Song object and adds it to the temporary list.
    - Catches `FileNotFoundException` and prints an error message if the file is not found.

Group Members
Eliot Cole
Kelvin Wilch
Bill Ngoga
Liam McGibbon

    - Converts the temporary `ArrayList` to a Song array (`songs`).
    - Sorts the `songs` array using `Arrays.sort()`. This leverages the `compareTo()` method in the Song class for sorting.

- **Methods:**
    - `getAllSongs()`: Returns the songs array containing the song collection.
    - **Main (static)**: This is a unit testing method that demonstrates how to create a SongCollection object, read songs from a file, and potentially display the number of songs (commented out in the provided code). It currently shows the first 10 elements using a stream operation.

Output

```
run:
testing getArtist: Professor B
testing getTitle: Small Steps
testing getLyrics:
Write your programs in small steps
small steps, small steps
Write your programs in small steps
Test and debug every step of the way.

testing toString:

Song 1: Professor B, "Small Steps"
Song 2: Brian Dill, "Ode to Bobby B"
Song 3: Professor B, "Debugger Love"
testing compareTo:
Song1 vs Song2 = 14
Song2 vs Song1 = -14
Song1 vs Song3 = 15
Song3 vs Song1 = -15
Song1 vs Song1 = 0
```