LEC 08

CSE 122

Sets, For-Each Loops, Iterators

Questions during Class?

Raise hand or send here

sli.do

BEFORE WE START

Talk to your neighbors:

Did you eat breakfast today?

Music: 122 24sp Lecture Tunes 🌼

Instructors	Miya Natsuhara and Kasey Champion			
TAs	Ayush	Kyle	Colin	Chaafen
	Poojitha	Jacob	Ronald	Smriti
	Chloe	Atharva	Saivi	Ambika
	Ailsa	Rucha	Shivani	Elizabeth
	Jasmine	Megana	Kavya	Aishah
	Lucas	Eesha	Steven	Minh
	Logan	Zane	Ken	Katharine

Lecture Outline

- **Announcements**
- Practice Problem
- Sets Review
- Tradeoffs with Different Data Structures
- For-Each Loop
- Iterators

Announcements

- Programming Assignment 1 (P1) due tomorrow, Thursday, April 25!
  - Stacks, Queues, Exceptions
- Resubmission Cycle 1 was due yesterday
  - Resubmission Cycle 2 will open tomorrow
- Heads up: Quiz 1 scheduled for Tuesday, May 7
  - Reference Semantics, Stacks and Queues, Sets, Maps
- How to Use the IPL
- 122 Playground
- Programming Assignment 2 releases on Friday, April 26
  - Yes, two Programming Assignments in a row
  - BUT, you have *two weeks* to complete this assignment

Lecture Outline

- Announcements
- **Practice Problem**
- Sets Review
- Tradeoffs with Different Data Structures
- For-Each Loop
- Iterators

Practice Problem:

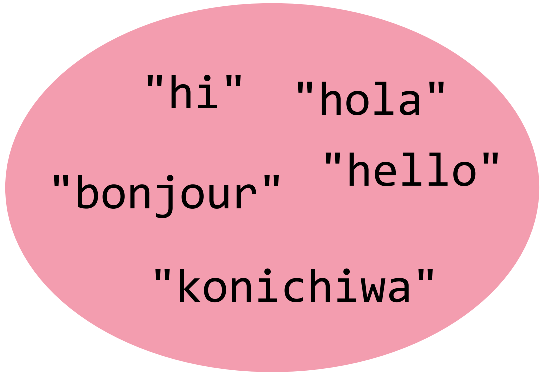
Write a program that, given a `Scanner` over a large text file (e.g., *Moby Dick* or the King James Bible), counts the number of unique words in the text.

Lecture Outline

- Announcements
- Practice Problem
- **Sets Review**
- Tradeoffs with Different Data Structures
- For-Each Loop
- Iterators

(PCM) Sets (ADT)

- A collection of unique values (no duplicates allowed) that can perform the following operations <u>efficiently</u>:
  - add
  - remove
  - search (contains)



- We don’t think of a set as having indices; we just add things to the set in general and don’t worry about order

(PCM) Sets in Java

- Set is an interface in Java
  - In `java.util`
  - Just like `List` and `Queue` are interfaces

- `HashSet` and `TreeSet` are classes that implement the `Set` interface in Java
  - HashSet: Very fast! Implemented using a “hash table” array
    - *Elements are stored in an unpredictable order*
    - Learn more about “Hashing” in CSE 332/CSE 373
  - `TreeSet`: Pretty fast! Implemented using a “binary search tree”
    - *Elements are stored in sorted order*
    - *Learn more about “Trees” in CSE 123*
  - Just like how `ArrayList` is an implementation of the `List` interface

Set Methods

Method	Description
`add(value)`	Adds the given value to the set, returns whether or not the given value was added successfully
`contains(value)`	Returns `true` if the given value is found in this set
`remove(value)`	Removes the given value from the set; returns `true` if the set contained the value, `false` if not
`clear()`	Removes all elements from the set
`size()`	Returns the number of elements in list
`isEmpty()`	Returns `true` if the set's size is 0; `false` otherwise
`toString()`	Returns a `String` representation of the set such as `"[3, 42, -7, 15]"`

Lecture Outline

- Announcements
- Practice Problem
- Sets Review
- **Tradeoffs with Different Data Structures**
- For-Each Loop
- Iterators

Choosing a Data Structure: Tradeoffs

- You got a bit of practice with this in your quiz sections on Tuesday!
  - Solving the same problem with an `ArrayList`, a `Stack`, and a `Queue`
  - Just because `ArrayList` can do all the same things `Stack` and `Queue` can, doesn't mean it's best for your problem

- Things to consider:
  - Functionality
    - If you need duplicates or indexing, `Sets` are not for you!
  - Efficiency
    - Different data structures are "good at" different things!

Lecture Outline

- Announcements
- Practice Problem
- Sets Review
- Tradeoffs with Different Data Structures
- For-Each Loop
- Iterators

For-Each Loop

- A new kind of loop!

```
Set<String> words = new HashSet<>();
for (String s : words) {
    System.out.println(s);
}
```

- BUT, you cannot *modify* the data structure inside a for-each loop
  - You will get a `ConcurrentModificationException`
  - They are “read-only”

Lecture Outline

- Announcements
- Practice Problem
- Sets Review
- Tradeoffs with Different Data Structures
- For-Each Loop
- **Iterators**

Iterators

A new object that has access to all of the elements of a given structure and can give them to you, one at a time.

Iterators

- Returned by the `iterator()` method

Methods	Description
`hasNext()`	Returns `true` if there are more elements for the iterator to return
`next()`	Returns the next element in the iteration
`remove()`	Removes and returns the element that was last returned by `next()`

- You must use the iterator's `remove()` method to remove things from what you're iterating over – otherwise you will get a `ConcurrentModificationException`