CS 2230
CS II: Data structures
Meeting 26: Map ADT and implementing it
Brandon Myers
University of Iowa
Today’s learning objectives

• Interpret code that uses the Map ADT
• Describe how to use a Map in an example situation
• Navigate a larger code base of ADTs and data structures
Airline tickets...revisited

1. TSA knows the ticket # of every person on today’s flights
2. When a person enters security, the agent checks that their ID card matches name associated with the ticket
3. Storing tickets #’s in a set doesn’t provide us the passenger’s name
Give agent the ticket to scan and your ID card

OK! 114 -> “Frog”
OK! 208 -> “Giraffe”
denied! 199 -> “Bird”

What data structure should we use to keep track of ticket #’s?
The Map ADT

public interface Map<K,V> {  
    /* Put the entry (key, value) into  
    * the map, overwriting the old value  
    * if the key is already in the map  
    */  
    void put(K key, V value);

    /* return the value for the given key  
    * or null if the key is not in the map  
    */  
    V get(K key);

    /* remove the mapping for the given key  
    * if one exists; return its value */  
    V remove(K key);

    /* return the number of entries in this Map */  
    int size();
}
### Example

Map is like Set except every **Key** has an associated **Value**

<table>
<thead>
<tr>
<th>Operation</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>put(5,A)</td>
<td>(5,A)</td>
</tr>
<tr>
<td>put(7,B)</td>
<td>(7,B)</td>
</tr>
<tr>
<td>put(2,C)</td>
<td>(2,C)</td>
</tr>
<tr>
<td>put(8,D)</td>
<td>(8,D)</td>
</tr>
<tr>
<td>put(2,E)</td>
<td>(2,E)</td>
</tr>
<tr>
<td>get(7)</td>
<td>B</td>
</tr>
<tr>
<td>get(4)</td>
<td>null</td>
</tr>
<tr>
<td>get(2)</td>
<td>E</td>
</tr>
<tr>
<td>size()</td>
<td>4</td>
</tr>
<tr>
<td>remove(5)</td>
<td>A</td>
</tr>
<tr>
<td>remove(2)</td>
<td>E</td>
</tr>
<tr>
<td>get(2)</td>
<td>null</td>
</tr>
</tbody>
</table>

Map:

\[
\emptyset \\
(5,A) \\
(7,B) \\
(2,C) \\
(8,D) \\
(5,A) \\
(7,B) \\
(2,E) \\
(8,D) \\
(5,A) \\
(7,B) \\
(2,E) \\
(8,D) \\
(5,A) \\
(7,B) \\
(2,E) \\
(8,D) \\
(7,B) \\
(8,D) \\
\]

- overwrite (2,C)
- no key 4
Interpret code that uses the Map ADT

Map<Integer, String> m =/*Map constructor */;
m.put(5, "Frog");
m.get(4); // 1
m.put(10, "Tiger");
m.put(5, "Fox");
m.size(); // 2
m.remove(4); // 3
m.get(5); // 4

what do the labeled lines return?
Give agent the ticket to scan and your ID card

OK! 114 -> “Frog”

OK! 208 -> “Giraffe”

denied! 199 -> “Bird”

person 1
person 2
person 3 (the sneak)
Describe how to use a Map in an example situation

How can we use a single Map to check IDs?

a) Map.put(ticket number, name) when reserving a ticket, Map.get(ticket number) when going through security

b) Map.put(ticket number, name) when reserving a ticket, Map.get(name) when going through security

c) Map.put(name, ticket number) when reserving a ticket, Map.get(ticket number) when going through security

d) Map.get(ticket number) when reserving a ticket, Map.put(ticket number, name) when going through security

public interface Map<K,V> {
    void put(K key, V value);
    V get(K key);
    V remove(K key);
    int size();
}

https://b.socrative.com/login/student/
room CS2230X ids 1000-2999
room CS2230Y ids 3000+
Today’s activity: Explore the Map code part1

• (download PDF on ICON)
Textbook’s Map class hierarchy

<table>
<thead>
<tr>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>int size();</td>
</tr>
<tr>
<td>boolean isEmpty();</td>
</tr>
<tr>
<td>V get(K key);</td>
</tr>
<tr>
<td>V put(K key, V value);</td>
</tr>
<tr>
<td>V remove(K key);</td>
</tr>
<tr>
<td>Iterable&lt;K&gt; keySet();</td>
</tr>
<tr>
<td>Iterable&lt;V&gt; values();</td>
</tr>
<tr>
<td>Iterable&lt;Entry&lt;K,V&gt;&gt; entrySet();</td>
</tr>
</tbody>
</table>
Today’s learning objectives

- Interpret code that uses the Map ADT
- Describe how to use a Map in an example situation
- Navigate a larger code base of ADTs and data structures
- Interpret Map implementation code
- Interpret code that uses Map
- Implement a Map method
- Implement an application that uses a Map
Setup to continue

Team | Git | Remote | Pull... | pick origin
(will be important for later questions)
9) Describe how TreeMap implements the SortedMap method firstEntry(). Be specific: you can copy code snippets and add your own comments to them. If you don't know what a method does, right click it | Navigate | go to Source.

10) How is an "abstract class" different from a "class"?

11) Of the methods in Map, which ones does AbstractMap implement and which does it leave unimplemented?
13) Now run the AirportSecurityDatabase. Did your prediction match? If not, then write what your misconception was.

14) Refer to the Map implementation class that AirportSecurityDatabase uses. What do the public methods use findIndex for?

15) What is findIndex using equals() for?

16) Why does findIndex use equals() instead of ==?

18) Now change the equals() in findIndex to ==. Run AirportSecurityDatabase and see what the new output is. Did your prediction match? If not, then write what your misconception was.
19) In UnsortedTableMap, look at put(). What does this put() implementation do when it finds the key? Why?

20) What is the purpose of the private class EntryIterator?

21) Map has two other methods that return iterators. What are they, and how does UnsortedTableMap implement them?
22) Go to AVLTreeMap.java. By navigating the code from there, find the method that actually decides which rotations to do. What is it called and what class is it in?

23) In the comments of that method, which examples correspond to the right, left, left-right, and right-left rotations?

24) How does that code (and/or code it calls) decide whether to do a right rotation or a left rotation?
Coding challenge

add the following method to the Map interface

```java
// if the key is not in the Map,
// then put the (key, value) into the map
// if the key is already in the Map,
// just return its value
V putIfAbsent(K key, V value);
```

Now, fix the new compiler errors in all the other files by implementing putIfAbsent

HINT: you only need to edit one other file
Create your own application in the apps/ package that uses a Map or SortedMap to do something interesting.
Today’s learning objectives

• Interpret code that uses the Map ADT
• Describe how to use a Map in an example situation
• Navigate a larger code base of ADTs and data structures
• Interpret Map implementation code
• Interpret code that uses Map
• Implement a Map method
• Implement an application that uses a Map
Acknowledgements

• Ticket Purchase by Andrew Jones from the Noun Project

• Airplane Ticket by Unlimiticon from the Noun Project