Dictionaries

Chapter 19

Contents

Objectives

- Describe operations of ADT dictionary
- Distinguish between a dictionary and a list
- Use a dictionary in a program

Specifications ADT Dictionary

- Synonyms for Dictionary
  - Map
  - Table
  - Associative table
- Contains
  - Search key
  - Value

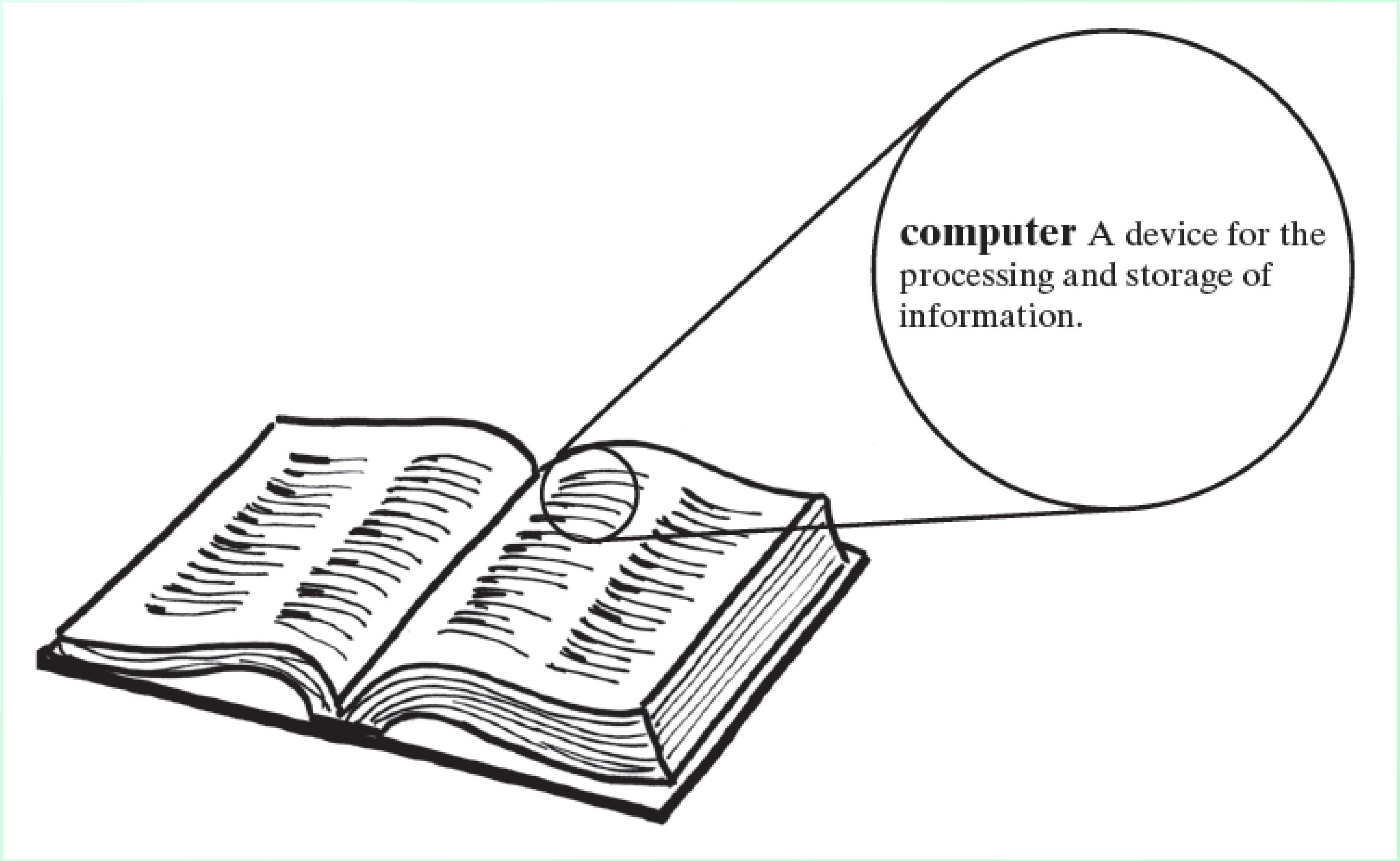

computer A device for the processing and storage of information.

Figure 19-1 An English dictionary

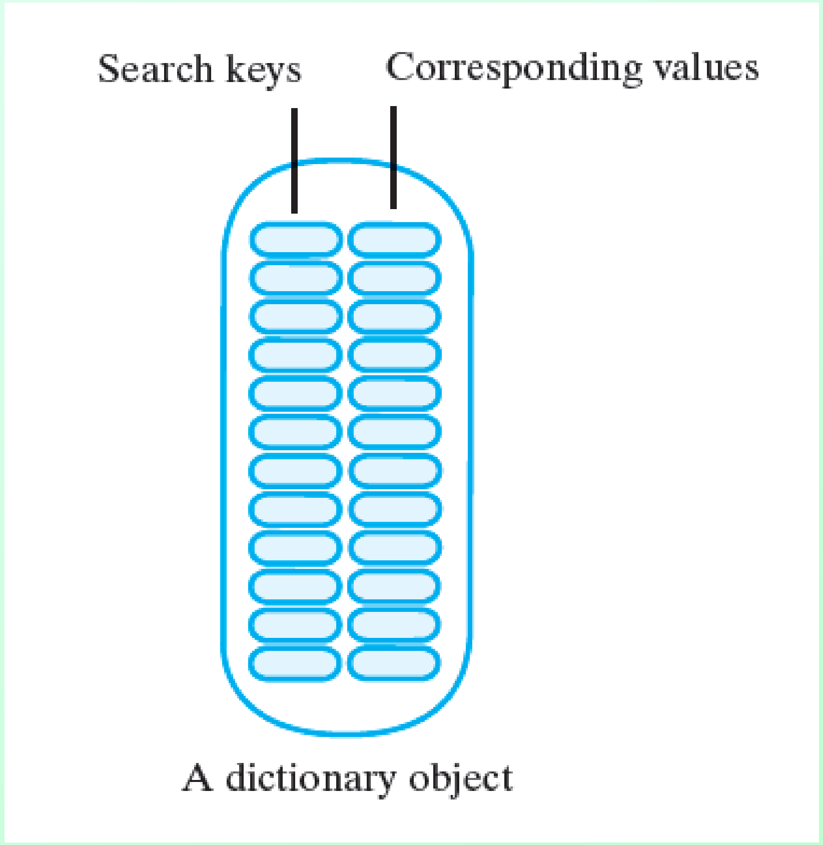


Figure 19-2 An instance of an ADT dictionary has search keys paired with corresponding values

Dictionary Operations

- Add new entry, given search key and associated value
- Remove an entry, given associated search key
- Retrieve value associated with given search key
- See whether dictionary contains a given search key

Dictionary Operations

- Traverse all search keys in dictionary
- Traverse all values in dictionary
- Detect whether dictionary is empty
- Get number of entries in dictionary
- Remove all entries from dictionary

Abstract Data Type: **`Dictionary`**

Operations

- **`add(key, value)`**
- **`remove(key)`**
- **`getValue(key)`**
- **`contains(key)`**
- **`getKeyIterator()`**
- **`getValueIterator()`**
- **`isEmpty()`**
- **`getSize()`**
- **`clear()`**

Refining the Specifications

- Distinct search keys
  - Method **`add`** must deal with finding a duplicate search key
- Duplicate search keys
  - Methods **`remove`** and **`getValue`** must deal with the duplicates

A Java Interface

- Listing 19-1 contains a Java interface for the ADT dictionary
- Iterators specified allow traversal of …
  - All search keys in dictionary without traversing values
  - All values search keys
  - All search es in parallel

Note: Code listing files must be in same folder as PowerPoint files for links to work

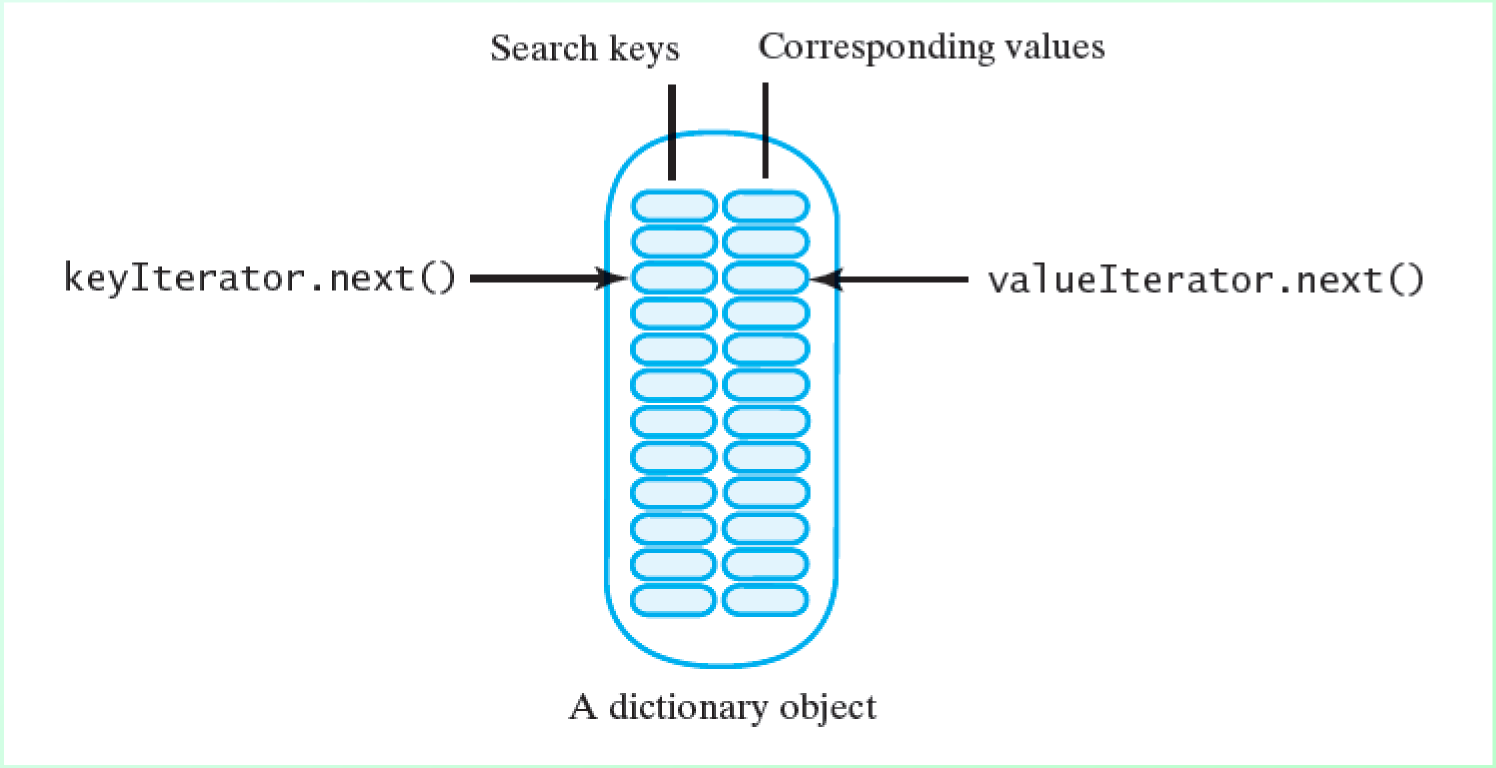


Figure 19-3 Two iterators that traverse a dictionary's keys and values in parallel

Question 1 If the class Dictionary implements DictionaryInterface, write a Java statement that creates an empty dictionary myDictionary. This dictionary will contain the names and telephone numbers of your friends. Assume that the names are the search keys, and you have the class Name to represent them. Let each telephone number be a string.

Question 2 Write a Java statement that adds your name and telephone number to the dictionary that you created in Question 1.

Question 1 If the class Dictionary implements DictionaryInterface, write a Java statement that creates an empty dictionary myDictionary. This dictionary will contain the names and telephone numbers of your friends. Assume that the names are the search keys, and you have the class Name to represent them. Let each telephone number be a string.

```
DictionaryInterface<Name, String> myDictionary = new  Dictionary<Name, String>();
```

Question 2 Write a Java statement that adds your name and telephone number to the dictionary that you created in Question 1.

```
myDictionary.add(new  Name("Joe", "Java"), "555-1234");
```

Question 3 Write Java statements that display either Steve Bruemmer’s telephone number, if he is in the dictionary from Question 1, or an error message if he is not.

Question 1 dictionary:

```
DictionaryInterface<Name, String> myDictionary = new  Dictionary<Name, String>();
```

Question 3 Write Java statements that display either Steve Bruemmer's telephone number, if he is in the dictionary from Question 1, or an error message if he is not.

Question 1 dictionary:

```
DictionaryInterface<Name, String> myDictionary = new  Dictionary<Name, String>();
```

```
Name steve = new  Name("Steve", "Bruemmer");
if (myDictionary.contains(steve))
   System.out.println("Steve's phone number is " + myDictionary.getValue(steve));
else
   System.out.println("Steve is not in the dictionary");
```

or

```
String phoneNumber = myDictionary.getValue( new  Name("Steve", "Bruemmer"));
if (phoneNumber == null)
   System.out.println("Steve is not in the dictionary");
else
   System.out.println("Steve's phone number is " + phoneNumber);
```

A Directory of Telephone Numbers

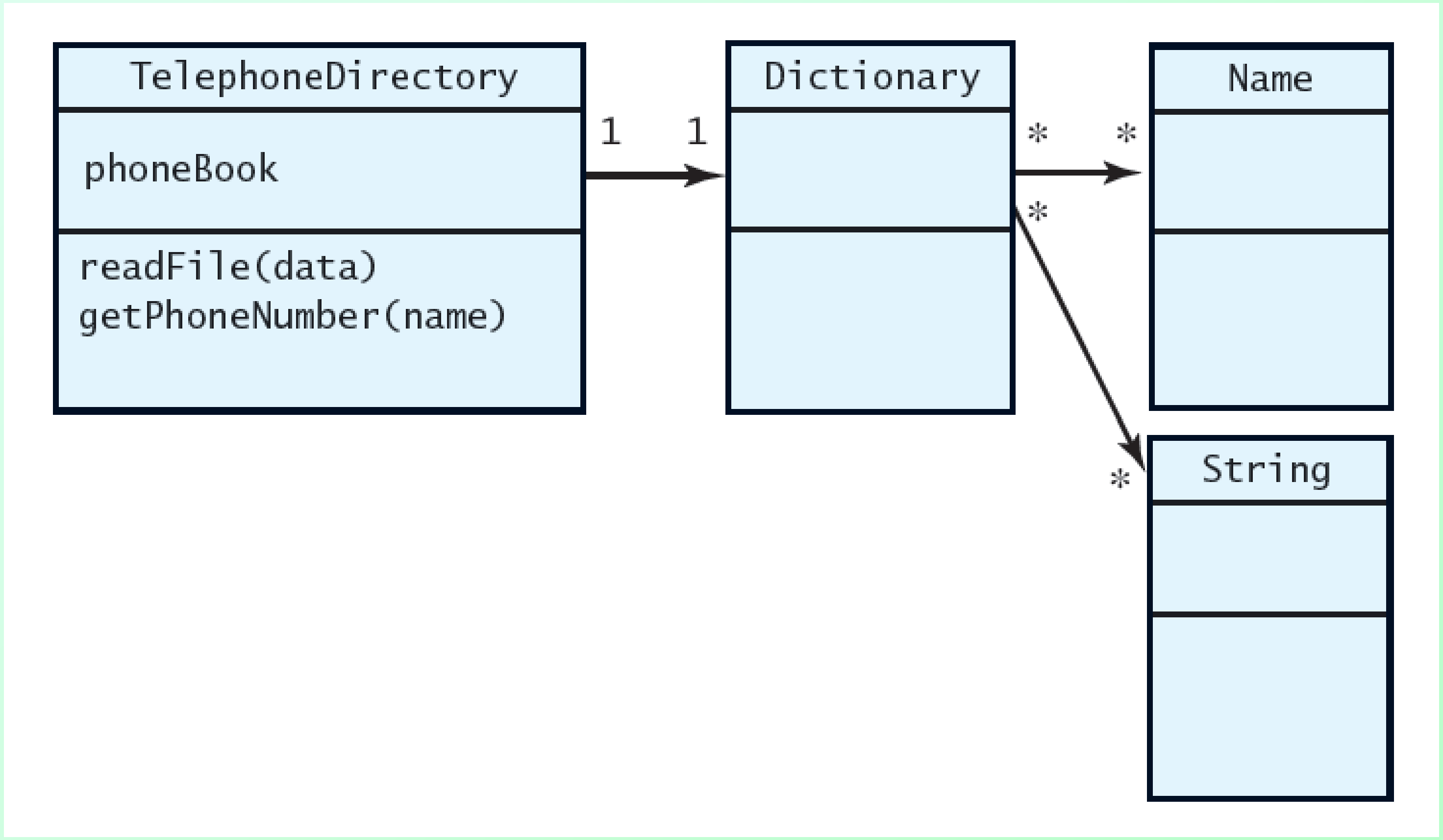


Figure 19-4 A class diagram for a telephone directory

A Directory of Telephone Numbers

- Consider how client will use the class
  Listing 19-2
- Anticipated output
- Assume distinct search keys
  - View class **`TelephoneDirectory`**,
    Listing 19-3

Question 4   Although the statement   directory.readFile(data);
is inside a try block near the beginning of the method main in Listing 19-2, it need not be. Explain its present location, why it can appear outside of a try block, and what you can do to move it.

Question 4   Although the statement   directory.readFile(data);
is inside a try block near the beginning of the method main in Listing 19-2, it need not be. Explain its present location, why it can appear outside of a try block, and what you can do to move it.

The Scanner methods hasNext and next that readFile calls throw only runtime exceptions, which need not be caught. So although the call to readFile can be outside of a try block, it is inside the try block because its argument—the Scanner object data—is local to the try block. By declaring data outside of the try block, you could move the call to readFile after the last catch block.

Question 5   Implement a method for the class TelephoneDirectory that removes an entry from the directory. Given the person's name, the method should return either the person's telephone number or null if the person is not in the directory.

Question 6 Implement a method for the class TelephoneDirectory that changes a person's telephone number. Given the person's name, the method should return either the person's old telephone number or null if the person was not in the directory but has been added to it.

Question 5 Implement a method for the class TelephoneDirectory that removes an entry from the directory. Given the person's name, the method should return either the person's telephone number or null if the person is not in the directory.

```
public  String remove(Name personName)
{     return  phoneBook.remove(personName);
}
```

Question 6 Implement a method for the class TelephoneDirectory that changes a person's telephone number. Given the person's name, the method should return either the person's old telephone number or null if the person was not in the directory but has been added to it.

```
public  String changePhoneNumber(Name personName, String newPhoneNumber)
{     return  phoneBook.add(personName, newPhoneNumber);
}
```

The Frequency of Words

- A class to count each occurrence of a word as a document is read
  - Will read the input text from a file
  - Results will be displayed as output
- Client program, Listing 19-4
  - Output for "*row, row, row, your boat*"
- The class **`FrequencyCounter`**, Listing 19-5

Question 7 The method readFile does not call contains to see whether a word is already in the dictionary, but instead calls getValue . Why did we do this?

```
public void  readFile(Scanner data)
{    data.useDelimiter("\\W+");
     while (data.hasNext())
     {          String nextWord = data.next();
                nextWord = nextWord.toLowerCase();
                Integer frequency = wordTable.getValue(nextWord);
                if (frequency == null)
                {          // add new word to table
                           wordTable.add(nextWord, new  Integer(1));
                }
                else       // increment count of existing word; replace wordTable entry
                {          frequency++;
                           wordTable.add(nextWord, frequency);
                }
   }
    data.close();
}
```

Question 7 The method readFile does not call contains to see whether a word is already in the dictionary, but instead calls getValue . Why did we do this?

```
public void  readFile(Scanner data)
{    data.useDelimiter("\\W+");
     while (data.hasNext())
     {          String nextWord = data.next();
                nextWord = nextWord.toLowerCase();
                Integer frequency = wordTable.getValue(nextWord);
                if (frequency == null)
                {               // add new word to table
                                wordTable.add(nextWord, new  Integer(1));
                }
                else            // increment count of existing word; replace wordTable entry
                {               frequency++;
                                wordTable.add(nextWord, frequency);
                }
   }
    data.close();
}
```

We called getValue instead of contains to simplify the logic. If we called contains and found that the current word was already in the dictionary, we would need to call getValue to get its frequency. But we can use the result of getValue to see whether the word is in the dictionary.

A Concordance of Words

- Read text from a file
  - List each word
  - Indicate line numbers in which word occurs
- Class **`Concordance`**, Listing 19-6
  - Note similarities to class **`FrequencyCounter`**

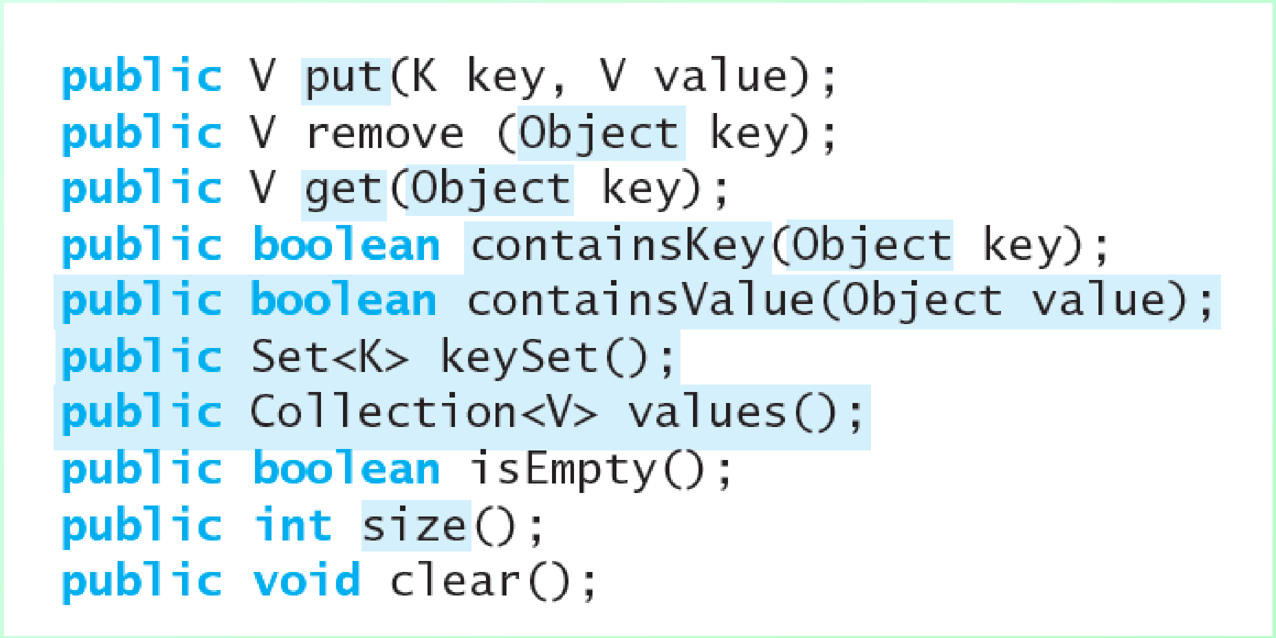
Question 9   Write a method getLineNumbers for the class Concordance that returns a list of the numbers of the lines that contain a given word.

Question 9   Write a method getLineNumbers for the class Concordance that returns a list of the numbers of the lines that contain a given word.

```
public  ListWithIteratorInterface<Integer>   getLineNumbers(String word)
{    return  wordTable.getValue(word);
}
```

Java Class Library: The Interface **`Map`**

- Method headers
  - Similar to our **`Dictionary`** class
  - Differences highlighted



```
public V put(K key, V value);
public V remove (Object key);
public V get(Object key);
public boolean containsKey(Object key);
public boolean containsValue(Object value);
public Set<K> keySet();
public Collection<V> values();
public boolean isEmpty();
public int size();
public void clear();
```

End

Chapter 19