Bags

Chapter 1

Copyright ©2012 by Pearson Education, Inc. All rights reserved

Reading Quiz

1. Which of the following is not a characteristic of a Bag object?
   a. A finite collection of objects
   b. Items arranged in a ring
   c. Items in no particular order
   d. May contain duplicate items

Copyright ©2012 by Pearson Education, Inc. All rights reserved

Contents

Copyright ©2012 by Pearson Education, Inc. All rights reserved

Objectives

- Describe the concept of abstract data type (ADT)
- Describe ADT bag
- Use ADT bag in Java program

Copyright ©2012 by Pearson Education, Inc. All rights reserved

Definition: Bag

- A finite collection of objects
- In no particular order
- May contain duplicate items

Copyright ©2012 by Pearson Education, Inc. All rights reserved

What's in the Bag?

```
Bag<String> aBag = new Bag<String>();


aBag.add("peas");
aBag.add("carrots");
aBag.add("tofu");
aBag.add("celery");
aBag.remove("tofu");
```

Copyright ©2012 by Pearson Education, Inc. All rights reserved

Behaviors

- Determine how many objects in bag
  - Full?
  - Empty?
- Add, remove objects
- Count duplicates
- Test for specific object
- View all objects

Copyright ©2012 by Pearson Education, Inc. All rights reserved

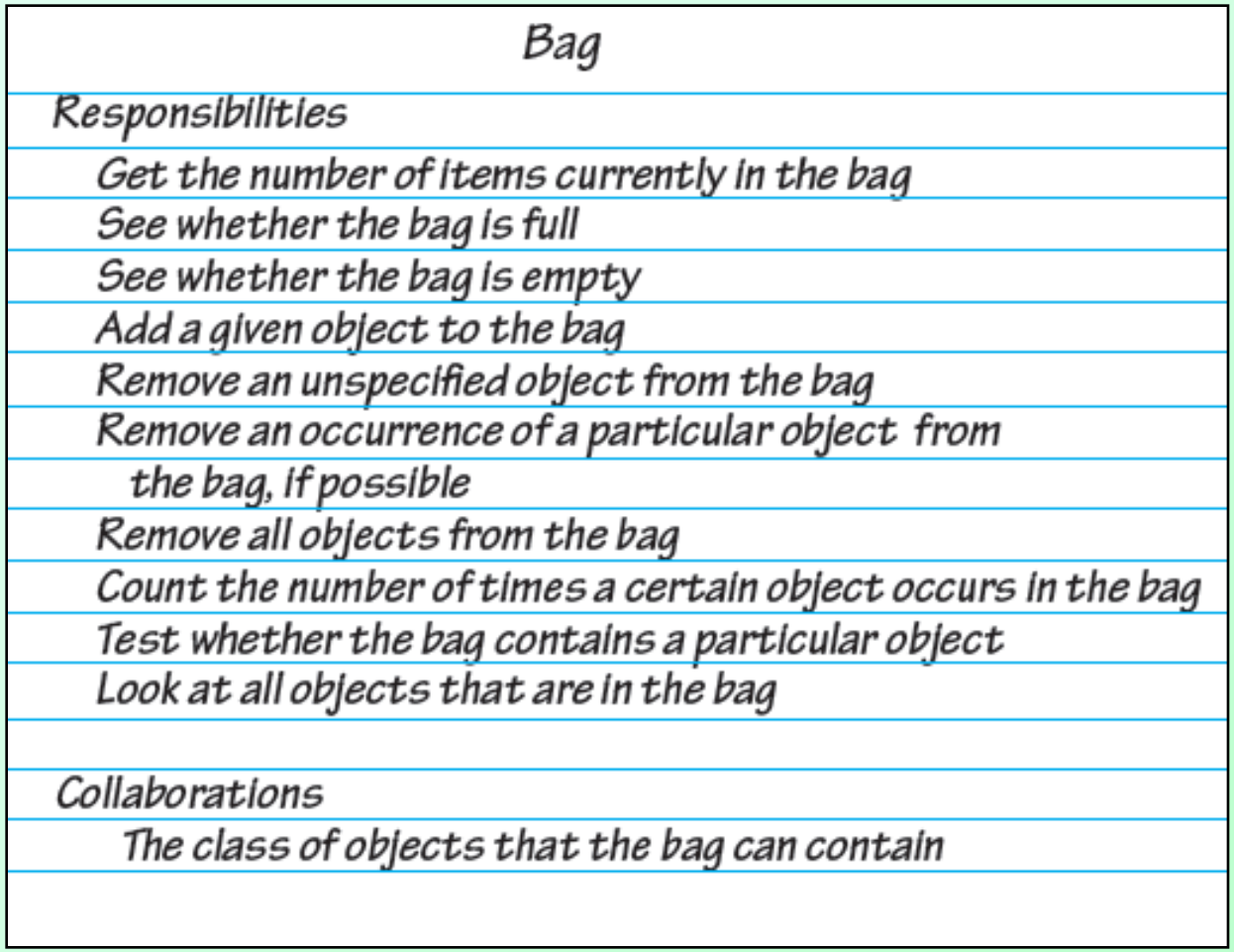

Figure 1-1 A CRC card for a class **`Bag`**

Copyright ©2012 by Pearson Education, Inc. All rights reserved

Specifying a Bag

- Describe data
- Specify methods for bag's behaviors
  - Name methods
  - Choose parameters
  - Decide return types
  - Write comments

Copyright ©2012 by Pearson Education, Inc. All rights reserved

Design Decisions

- What should the method **`add`** do when it cannot add a new entry?
  - Nothing?
  - Leave bag unchanged, signal client of condition?

Copyright ©2012 by Pearson Education, Inc. All rights reserved

Design Decisions

- What should happen when an unusual condition occurs?
  - Assume invalid never happens?
  - Ignore invalid event?
  - Guess at client's intention?
  - Return flag value?
  - Return boolean value – success/failure?
  - Throw exception?

Copyright ©2012 by Pearson Education, Inc. All rights reserved

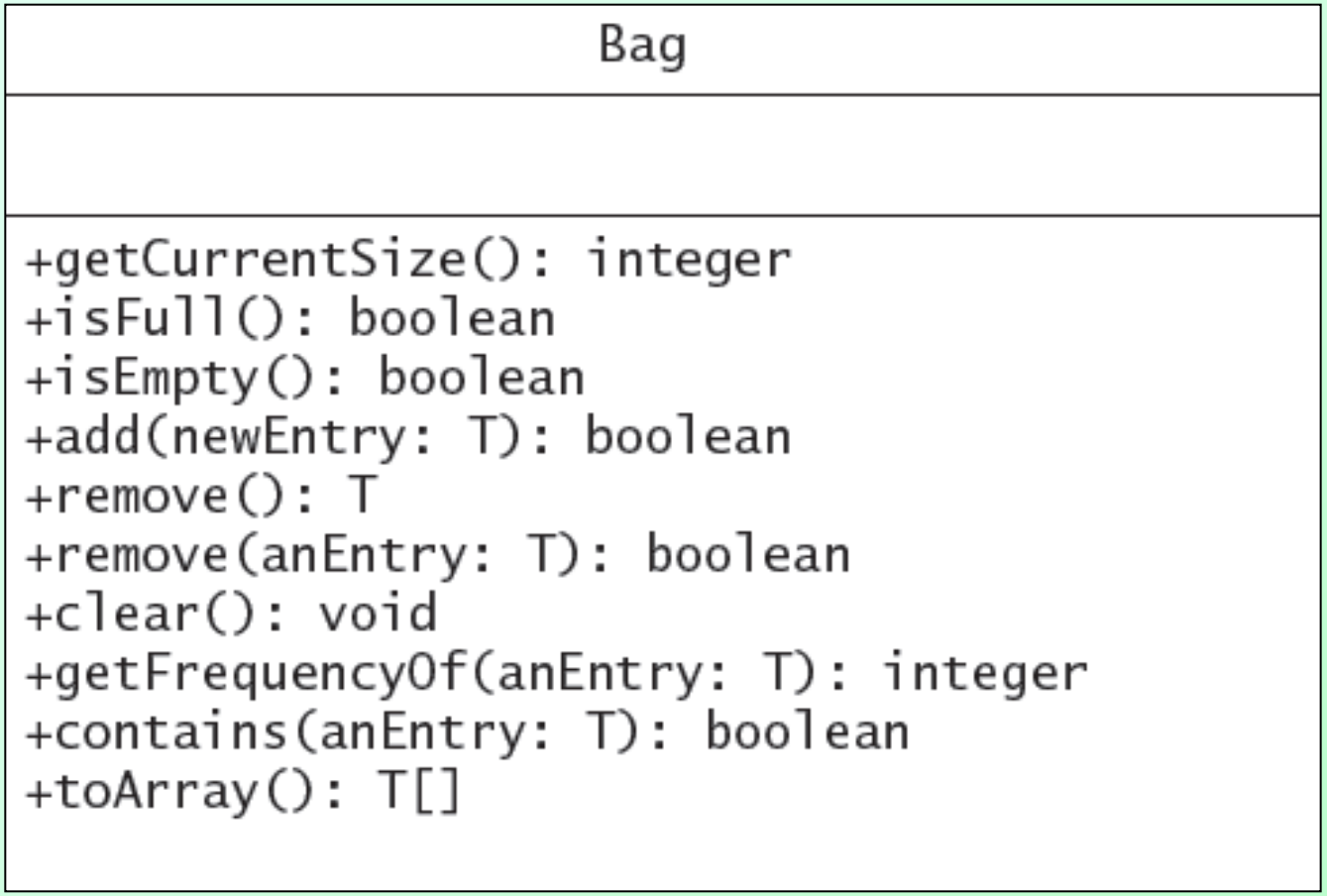

Figure 1-2 UML notation for the class **`Bag`**

Copyright ©2012 by Pearson Education, Inc. All rights reserved

Checkpoint 1

0) Write the Java signature for the add method based on the previous UML diagram

Question 1 Suppose `aBag` represents an empty bag that has a finite capacity. Write some pseudocode statements to add user-supplied strings to the bag until it becomes full.

Question 3 Is it legal to have two versions of `remove`, one that has no parameter and one that has a parameter, in the same class? Explain.

Question 4 Given the full bag `aBag` that you created in Question 1, write some pseudocode statements that remove and display all of the strings in the bag.

Copyright ©2012 by Pearson Education, Inc. All rights reserved

Checkpoint 1

0) Write the Java signature for the add method based on the previous UML diagram **boolean add(T newEntry);**

Question 1 Suppose `aBag` represents an empty bag that has a finite capacity. Write some pseudocode statements to add user-supplied strings to the bag until it becomes full.

```
1. // aBag is empty
   do
   {
      entry = next string read from user
      aBag.add(entry)
   } while (!aBag.isFull())
   // aBag is full
```

Question 3 Is it legal to have two versions of `remove`, one that has no parameter and one that has a parameter, in the same class? Explain.

Yes. The two methods have different signatures. They are overloaded methods.

Question 4 Given the full bag `aBag` that you created in Question 1, write some pseudocode statements that remove and display all of the strings in the bag.

```
// aBag is full
while (!aBag.isEmpty())
{
   entry = aBag.remove()
   Display entry
}
// aBag is empty
```

Copyright ©2012 by Pearson Education, Inc. All rights reserved

Interface

- Write Java headers
- Organize into interface
- Note items in bag are of same type
  - Generic type **<T>**
- View Listing 1-1

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

Copyright ©2012 by Pearson Education, Inc. All rights reserved

Using ADT `Bag`

- Implementation done from specifications
  - User needs know what ADT does, not how
- Type of object in bag specified by program using the ADT
- Example of `Bag` for online shopping
  Listing 1-2 See OnlineShoppingApp in download

Copyright ©2012 by Pearson Education, Inc. All rights reserved

Checkpoint 2

Question 5 Given the full bag `aBag` that you created in Question 1, write some pseudocode statements to find the number of times, if any, that the string `"Hello"` occurs in `aBag`.

Question 6 Given the full bag `aBag` that you created in Question 1, write some Java statements that display all of the strings in `aBag`. Do not alter the contents of `aBag`.

Copyright ©2012 by Pearson Education, Inc. All rights reserved

Checkpoint 2

Question 5 Given the full bag `aBag` that you created in Question 1, write some pseudocode statements to find the number of times, if any, that the string `"Hello"` occurs in `aBag`.

```
Display "The string Hello occurs in aBag " + aBag.getFrequencyOf("Hello") + " times."
```

Question 6 Given the full bag `aBag` that you created in Question 1, write some Java statements that display all of the strings in `aBag`. Do not alter the contents of `aBag`.

```
String[] contents = aBag.toArray();
for (int index = 0; index < contents.length; index++)
   System.out.print(contents[index] + " ");
System.out.println();
```

Copyright ©2012 by Pearson Education, Inc. All rights reserved

Using ADT **`Bag`**

- Example of **`Bag`** for class of piggy banks
  Listing 1-3
- Demonstration of class **`PiggyBank`**
  Listing 1-4

Copyright ©2012 by Pearson Education, Inc. All rights reserved

Figure 1-3 A Vending Machine

Copyright ©2012 by Pearson Education, Inc. All rights reserved

Vending Machine Like An ADT

- Perform only available tasks
- User must understand the tasks
- Cannot access inside of mechanism
- Usable without knowing inside implementation
- New inside implementation unknown to users

Copyright ©2012 by Pearson Education, Inc. All rights reserved

Checkpoint 3

Question 8 Consider the program in Listing 1-4. After creating the instance myBank of the class PiggyBank, suppose that we add several unknown coins to myBank. Write some code that will remove coins from the bank until either you remove a penny or the bank becomes empty.

Copyright ©2012 by Pearson Education, Inc. All rights reserved

Checkpoint 3

Question 8 Consider the program in Listing 1-4. After creating the instance myBank of the class PiggyBank, suppose that we add several unknown coins to myBank. Write some code that will remove coins from the bank until either you remove a penny or the bank becomes empty.

```
boolean lookingForPenny = true;
while (!myBank.isEmpty() && lookingForPenny)
{
   Coin removedCoin = myBank.remove();
   System.out.println("Removed a " + removedCoin.getCoinName() + ".");
   if (removedCoin.getCoinName() == CoinName.PENNY)
// if (removedCoin.getValue() == 1) // ALTERNATE
   {
      System.out.println("Found a penny. All done!");
      lookingForPenny = false; // penny is found
   }
} // end while

if (lookingForPenny)
   System.out.println("No penny was found. Sorry!");
```

Copyright ©2012 by Pearson Education, Inc. All rights reserved

Java Class Library

- The interface **Set** (**import java.util.Set**)
  - **a Bag that contains only unique entries**
  - **no duplicates**
  - **works with Object not generic type T**

```
public boolean add(Object newEntry)
public boolean remove(Object anEntry)
public void clear()
public boolean contains(Object anEntry)
public boolean isEmpty()
public int size()
public Object[] toArray()
```

Copyright ©2012 by Pearson Education, Inc. All rights reserved

End

Chapter 1

Copyright ©2012 by Pearson Education, Inc. All rights reserved