Classes with Generics

DEBORAH A. TRYTTEN

CS 1324

Data Structures

- Ways of organizing data for efficient storage and retrieval
- ArrayList is our first data structure
  - Resizable array (special case of oversize array)
- Part of Java Collections Framework (JCF)
  - Library of Java data structures
  - API will look at bit odd at first
    - Uses advanced programming techniques
- There is a companion class of class methods
  - Collections

Why Do This Now?

- We can have more fun
  - With this class we can store large amounts of data
  - We can write programs more like the ones you really use
- Continue to improve OOP vocabulary
  - Class versus object
  - Class methods versus instance (object) methods
  - Parameters versus arguments
- Have a really amazing example of a mutable class

What is an ArrayList?

- A way of storing multiple objects in a sequence
- Objects are zero indexed
- No gaps permitted in sequence
- All of the objects must be the same type
  - Homogeneous
- Familiar properties hint at what's hidden inside

Construct an ArrayList

- Construct an ArrayList that stores String objects
- Read API
- What is <E>?
  - A variable that stands in for the type of data stored
  - We'll use String and wrapper classes
- Declare a reference

```
ArrayList<String> list;
```

- Construct an object

```
list = new ArrayList<String>();
```

- Draw memory diagram
  - list contains 10 null references
  - list does not contain any objects initially

Instant Quiz Question 1

- Which line of code below would construct an ArrayList that can store rainfall in inches?

a) ArrayList<Double> list = new ArrayList();

b) ArrayList<double> list = new ArrayList<double>();

c) ArrayList<double> list = new ArrayList();

d) ArrayList<Double> list = new ArrayList<Double>();

e) ArrayList<Double> list;

Capacity and Size

- ArrayList objects are given an initial capacity of 10 by default constructor
  - This is the number of objects that can be stored without resizing
- There is another constructor that lets you control the initial capacity
  - Use it if you know capacity in advance
- An ArrayList object initially has a size of 0
  - No objects are stored in it
- As objects are added the ArrayList object increases its size
- Capacity is automatically increased as necessary behind the scenes (encapsulated)

Instant Quiz Question 2

- Suppose that we created an ArrayList<Integer> object and have added three Integer objects (1, 3, and 5)
- Which of the following is true?
- a) Size is 3 and capacity is 10
- b) Size is 3 and capacity is 3
- c) Size is 10 and capacity is 3
- d) Size is 10 and capacity is 10

Shopping List: Add items

- Look for method in API
  - Accessor or mutator?
- Add items at end of list
- Add items at start of list
- Show memory diagram
- Where are we allowed to put items?
  - Read API for add(int index, E Element)

Shopping List: Display Items

- Use toString()
- How to access individual elements?
- How many elements are in the list?
- What control structure should we use?
- Write code to display all elements in this format
  - 1. First item
  - 2. Second item
  - …

Shopping List: Delete Items

- By item number
  - How to avoid leaving gaps
- By name
- All items
  - Two ways

Collections Class

- Similarity to Arrays
- Contains only static methods
  - Some really crazy syntax
    - Use common sense
- Arraylist objects are passed by sharing
  - Exactly like arrays
- If new ArrayList is constructed inside method it must be returned

Collections Rules

- ArrayList can be arguments for any parameter that asks for
  - Collection
  - List
- Any method that returns a Collection or List may be stored in an ArrayList
- Methods that need to do < only work on objects that have compareTo() methods
  - String, Integer, Double, Character, other wrappers
  - What would it mean to sort Point objects?

Lots of Great Methods

sort()

copy()

disjoint()

fill()

frequency()

max()

min()

reverse()

- Print the shoppingList in sorted order

Instant Quiz Question 3

- Suppose the Collections class had a method that performed linear search for a target value. What would the signature be?

a) boolean contains()
b) boolean contains(E target)
c) boolean contains(ArrayList<E> list, E target)
d) boolean contains (ArrayList<E> list)
e) void contains (ArrayList<E> list, E target, boolean isInThere)

Using Wrapper Classes

- Suppose that we’re storing examination scores
  - ArrayList<Integer> scores;
- Add the scores 90 to 99 to the ArrayList object
  - Add 93 four times
- Play with some Collections methods
  - Find the maximum
  - Find how many times 93 occurred
  - Shuffle the list

Binary Search

- Same as in Arrays class
- Requires sorted data
  - If not sorted the results are undefined
- Analogy to dictionary
- Algorithm same as for arrays
- Very, very fast when compared to linear search

Instant Quiz Question 4

- Suppose we execute the following code

```
ArrayList<Integer> list = new ArrayList<Integer>();
for (int i=15; i>0; --i)
    list.add(new Integer(i));  // read carefully
int index = Collections.binarySearch(list,
                new Integer(2));
```

Which of the following is true:

a) The index will be positive

b) The index will be zero

c) The index will be negative

d) The index will be meaningless

Autoboxing

- Java knows the relationship between primitive data types and wrapper classes
  - If you insert a primitive data type instead of an object Java constructs the object automatically
- Rework the example from the last slide with int instead of Integer
  - Remember that objects are being constructed
- Similarity to hidden String constructor

ArrayList and Mutable Objects

- ArrayLists and mutable objects can cause trouble
- Create ArrayList of StringBuilder objects
  - Sort it
  - Save the reference to the second object
  - Change it
  - What happens to the object in the ArrayList?
- Show memory diagram
- Try sorting and binary search
- Why would JCF work this way?
- Note Java strongly prefers immutable objects
  - Avoids subtle debugging problems