CSE1720

Week 03, Lecture 05

Winter 2014 ◆ Tuesday, Jan 21, 2014

YORK UNIVERSITÉ UNIVERSITY

Objectives for this class meeting

1. Complete Discussion about sec 8.1.1-8.1.4 "Aggregation"
   - focus on aggregations that are collections

YORK UNIVERSITÉ UNIVERSITY

Let's Revisit Our Example from L04

- Suppose our elements are the colours of the rainbow
- We will use the class `java.awt.Color` to encapsulate each colour

```
Color red = new Color(255, 0, 0);
Color orange = new Color(255, 165, 0);
Color yellow = new Color(255, 255, 0);
Color green = new Color(0, 255, 0);
Color blue = new Color(0, 0, 255);
Color purple = new Color(128, 0, 128);
```

3

YORK UNIVERSITÉ UNIVERSITY

Using an collection…

- Refer to code example `L04Ex02`

```
ArrayList<Color> theRainbow1 = new ArrayList<Color>();
```

- can I add more elements to an `ArrayList` collection?

4

YORK UNIVERSITÉ UNIVERSITY

Alias, Shallow Copy, Deep Copy

- Let's draw a memory diagram of an alias, and then do the same for each of a shallow copy and deep copy
- See code example `L04Ex03_alias`, `L04Ex04_shallow`, `L04Ex05_deep`

YORK UNIVERSITÉ UNIVERSITY

What does "traverse" mean?

A traversal can be thought of as a trip that visits each element once and only once.

JBA, p.318

What this means:

- No element can be missed
- No element can be visited more than once.

YORK UNIVERSITÉ UNIVERSITY

What does "traverse" **NOT** mean?

That the elements will be visited in **any particular order**

- Even if you traverse a given collection several times, you should not assume that the elements will be visited in the same order.
- There is **no order** defined over the elements (even if the elements are things that you may commonly think of as having a "natural" order, such as numbers)

7

What does "traverse" **NOT** mean?

Traverse doesn't mean you get to do **"partial trips"**

- e.g., visit "every other element" or "the first half of the elements" or any other trip that is anything other than the complete traversal of all the elements
- A collection simply is not defined to provide this.
- This is just a variant of trying to **impose a particular order** on the elements of the collection.

8

Iterator-Based Traversal 8.2.4

If `collection` is a variable that refers to a collection object, then the following enhanced for loop will be provided:

```
for (ElementType e : collection) {
     // visit element e
}
```

OK, but what is `ElementType`?

…a collection is an aggregate, which means, by definition *a class that has as one of its features an attribute that is non-primitive.*
What is this non-primitive type?
Don't know – let's just call it `ElementType` for the time being…

Iterator-Based Traversal 8.2.4

Another version of iterator-based traversal is…

```
while (collection.hasNext()) {
     ElementType e = collection.next();
}
```

It is equivalent to the enhanced for loop version…

(see the API of the `Iterable` interface)

Indexed Traversal 8.2.3

A sneaky bit of material that has the potential to confuse…

we just emphasized that traverse does not mean visiting the elements in any particular order… but…

Sometimes a collection may, *in addition to its requisite methods*, also support **an indexing scheme for its elements,** such as via this method:

```
public ElementType get(int index)
```


11

Indexed Traversal 8.2.3

If there is an indexing scheme, then we can implement full and even partial traversals…

```
for (int i = 0; i < collection.size(); i++) {
     ElementType e = collection.get(i);
}
```

Since collection is a collection, it must have a size() method

12

Where can I get a collection?

- Some classes encapsulate collections
  - E.g., the `Portfolio` class implements a collection of `CreditCard` elements; use the static method `getRandom()` to get a randomly-populated collection.
  - A `Portfolio` object is a collection because of the services it offers (can add, remove `CreditCard` elements, can iterate over it)
- Other classes offer services, start out as empty collections
  - E.g., the `ArrayList<E>` class implements a collection of elements of type `E`
  - When invoking the constructor, the client must specify the type `E` of the elements.

Where can I get me a collection?

What if I want to create and populate my own collection:

1. use a constructor to create an empty collection
2. add the elements one by one

A design tradeoff

When a new, empty collection is created, a block of run-time memory gets allocated for this object.

How large should this block be?

- If the block is too small:
  - then the size will be quickly exceed. When this happens, a whole new empty collection will need to be created, using a larger block and all of the elements copied over.
- If the block is too large:
  - a significant amount of memory will sit empty and cannot be used for anything else

The extreme form of the **small block** version is to start with a block so small that is it not big enough to hold even a single element. Then the amount of memory that is used grows as the collection grows.

YORK UNIVERSITÉ UNIVERSITY

Questions about Collections

- What is a collection?
  What is an aggregate with variable multiplicity?
  How are these questions related?
- RQ8.19 What does variable multiplicity mean?
  How is aggregation depicted in UML, both with fixed and with variable multiplicity?
- RQ8.20 If a collection is statically allocated, then what should be passed to its constructor?
- RQ8.21 Can you add an element to a collection even if it is already in it?

YORK UNIVERSITÉ UNIVERSITY

Questions about Collections

- RQ8.22 What happens if you attempt to add an element to a full, statically-allocated collection?
- RQ8.23 What is a traversal?
- RQ8.24 How do you determine the number of elements in a collection if it supports indexed traversals?
- RQ8.25 How do you determine the number of elements in a collection if it supports iterator-based traversals?
- RQ8.26 (a) Explain how a traversal can be used to perform a search. (b) Why are traversal-based searches called exhaustive?

YORK UNIVERSITÉ UNIVERSITY