

Iterators

9 April 2008

CMPT166

Dr. Sean Ho

Trinity Western University

Quiz 7

- Create an **ArrayList** of Strings, fill it with your choice of Strings, and print the list. [4]
- What is a **design pattern**? [4]
- **Name** and **describe** two design patterns. [4]
- What is a Java **Collection**? What can you do with one? [4]
- What is the relationship among **List**, **AbstractList**, and **ArrayList**? [4]

Quiz 7: answers

- Create an `ArrayList` of Strings, fill it with your choice of Strings, and print the list. [4]

- ◆ `ArrayList<String> myApples
= new ArrayList<String>();`
- ◆ `myApples.add("Fuji");`
- ◆ `myApples.add("Red Delicious");`
- ◆ `myApples.add("Golden Delicious");`
- ◆ `myApples.add("Gala");`
- ◆ `for (String apple : myApples)
System.out.println(apple);`

Quiz 7: answers

- What is a **design pattern**? [4]
 - Named **abstraction** from a recurring concrete **form**, expressing the **essence** of a proven general **solution** technique
- **Name** and **describe** two design patterns. [4]
 - **Creational**: factory method, abstract factory, prototype, singleton
 - **Structural**: facade, adapter, bridge, proxy, decorator, flyweight
 - **Behavioural**: observer, mediator, chain of responsibility, iterator, memento, command

Quiz 7: answers

- What is a Java **Collection**? What can you do with one? [4]
 - An object that **holds** other objects
 - Check **size**, if it **contains** an object, **iterate**
- What is the relationship among **List**, **AbstractList**, and **ArrayList**? [4]
 - List: **interface** (sub-interface of Collection)
 - AbstractList: **abstract** superclass impl List
 - ArrayList: **concrete** subclass of AbstractList

Collections

- A **collection** is an object that holds other objects
 - Java **interface**: `Collection<T>` (java.util)
 - must implement: `isEmpty()`, `size()`, `contains()`, `containsAll()`, `toArray()`, `iterator()`, etc.
- `toArray()`: convert to a regular **array** of objects
- `containsAll(Collection<?> c)`: check if **subset**
 - **wildcard**: `ArrayList<?>`: type can be anything
 - ◆ `ArrayList<? extends String>`: type must subclass `String`
- **Optional** (empty bodies): `add()`, `remove()`

Iterators

- An **iterator** is an object that steps through each element in a collection
 - it is not the **collection** object itself
- **Iterator<T> interface** (also in java.util):
 - **public T next()**
 - ◆ Get **next object** in the collection, or throw **NoSuchElementException**
 - **public boolean hasNext()**
 - **Optional: public void remove()**
 - ◆ **Remove** the object last returned by next()