Activity 17-3: The Adapter Patterns

Why?

The Adapter patterns are widely used and should be in every software engineering designer's pattern toolbox.

Learning Objectives

- Understand the Adapter patterns and how to use them in design
- Know how to select between the Adapter patterns

Success Criteria

- Be able to explain the structure and behavior of the Class and Object Adapter patterns
- Be able to explain the criteria used to choose between the Class and Object Adapter patterns
- Be able to use the Adapter patterns in mid-level design and implementation

Resources

ISED section 17.3

Vocabulary

Wrapper, adapter, adaptee, thread-safe

Plan

1. Review *ISED* section 17.3 individually.
2. Answer the Key Questions individually, and then evaluate the answers as a team.
3. Do the Exercises as a team, and check your answers with the instructor.
4. Do the Problems and Assessment as a team.
5. Turn in the Problems and Assessment as a team deliverable.

Key Questions

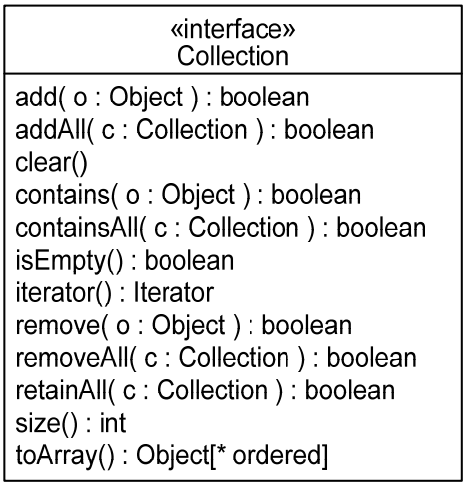
1. Explain the difference between the Class and Object Adapter patterns.
2. What does it mean for a class to be thread-safe?

Exercises

1. How can the Adapter patterns be used to make a class thread-safe?
2. What considerations help make a decision about which of the Adapter patterns to use in a particular situation?

Problems (Deliverable)

One of the operations in the `java.util.Collections` class is `static Collection unmodifiableCollection( Collection c )` (actually this operation uses generics, but we ignore these for now). This operation accepts an arbitrary `Collection` as a parameter and returns a `Collection` that has the same elements but cannot be modified. Calling any `Collection` operation that attempts to modify the new `Collection` results in an `UnsupportedOperationExeception`. The `Collection` interface is pictured below.



1. How might the `unmodifiableCollection()` operation be implemented?
2. You can use an Adapter pattern to solve this design problem. Which Adapter pattern is most appropriate and why?
3. Suppose that you had an operation for making unmodifiable versions of a class (rather than an interface). Which Adapter pattern would you use in that case and why?
4. Make a class diagram detailing your design for implementing the `unmodifiableCollection()` operation using the Adapter pattern you have chosen. Indicate with stereotypes which class is the adapter and which the adaptee.
5. Write some Java code in accord with your design. In particular, write code for the `unmodifiableCollection()` operation, the adapter class constructor, and the `isEmpty()` and `add()` operations in the adapter class.

Assessment (Deliverable)

1. Did this activity help you achieve the learning objectives?
2. How could the instructor improve this activity?