

Lecture

Chapter 4

Data Abstraction

Abstract Data Types

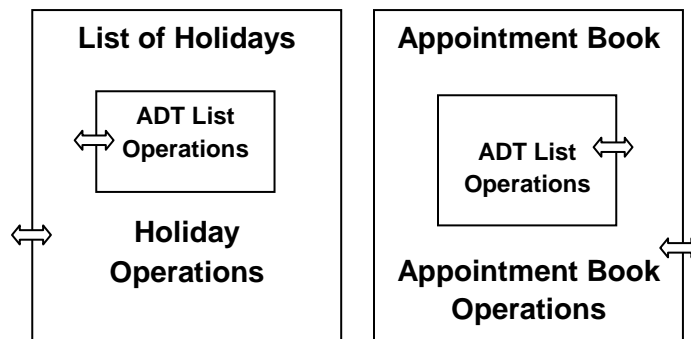
- Write a set of specifications that details the behavior of the module
 - focus on what the module does
 - think about what details will be hidden within the module
 - do not think of how the module will accomplish the task
- Procedural Abstraction
- Information Hiding
 - Identify details to be hidden within the module
 - Make the details inaccessible from outside the module
 - the specifications should include a description of what each operation of the ADT accomplishes, especially access to the restricted information
 - add data
 - remove data
 - queries about the data
 - Data Abstraction
 - Public Information – specify what you can do to a collection of data
 - Private Information
 - ✓ how the data collection is implemented
 - ✓ how the data is manipulated within the module
 - ADT
 - collection of data
 - specify subject matter of the data
 - do not specify how the data is stored
 - set of operations on that data
 - description of the operations must be sufficiently rigorous to completely specify their effect on the data
 - do not specify how they will accomplish their task
 - interface that allows communication with the module, implementation of the contract
- contract

using the interface contract a user requests the ADT operations to manipulate the data in a manner allowed by the contract and pass the results back to the user

Selected ADTs

- ADT List
- Operations -- Prichard 3rd ed. pgs 204
- Pseudocode for the ADT List Operations -- Prichard 3rd ed. pgs 206
- Pseudocode for the ADT Sorted List Operations -- Prichard 3rd ed. pgs 210
 - sorted list (sorted collection) Prichard 3rd ed. pgs 209-210
 - ordered (by position)
 - sorted (by value of the data items)
- Determining a List of Holidays -- Prichard 3rd ed. pgs 211-212
- Specifying an Appointment Book -- Prichard 3rd ed. pgs 212-214

Both the List of Holidays and the Appointment Book can be built on the ADT List



ADT List structure is embedded within the Holiday ADT List structure

ADT List structure is embedded within the Appointment Book ADT List structure

Implementing an ADT List on an Array – Prichard 3rd ed. pgs 226-230

- Each ADT List Operation must be implemented to make use of the available Array Operations. i.e.,
 - add to the end of the list
 - remove an item from the list → shift items to fill the void
 - etc

Axioms for the ADT List – Prichard 3rd ed. pgs 216-217

- Invariants, i.e., composites of the operations which are always TRUE
- Preconditions
- Postconditions

Complete Set of Axioms, Preconditions and Postconditions are all required to completely define the behavior of the operations of an ADT

Implementing an ADT

- **Java Classes – encapsulation**
 - **Subclasses – extend**
 - **Objects**
 - **Constructors**
 - **Garbage collection**
 - **Methods**
 - **Interfaces – implement -- collection of methods without implementation details, i.e., no bodies**
 - ✓ **java.util.Collection**

```
{ public Boolean add(Object o);  
  public Boolean contains(Object o);  
  ...  
}
```
 - ✓ **java.util.Comparable<object>**

```
{ public int compareTo(ObjectTo(object rhs);  
  ...  
}
```