Lab 6 Lists and Iterators

Data Structures & Algorithms

Objectives:

After performing this lab, the students should be able to

- implement a data structure and its associated operations using a positional list ADT

Activities

Implement a `CardHand` class that supports a person arranging a group of cards in his or her hand. The simulator should represent the sequence of cards using a single positional list ADT so that cards of the same suit are kept together. Implement this strategy by means of four "fingers" into the hand, one for each of the suits of hearts, clubs, spades, and diamonds, so that adding a new card to the person's hand or playing a correct card from the hand can be done in constant time. The class should support the following methods:

- `addCard(r, s)`: Add a new card with rank `r` and suit `s` to the hand.
- `play(s)`: Remove and return a card of suit `s` from the player's hand; if there is no card of suit `s`, then remove and return an arbitrary card from the hand.
- `iterator( )`: Return an iterator for all cards currently in the hand.
- `suitIterator(s)`: Return an iterator for all cards of suit `s` that are currently in the hand.

Hint: Keep all cards in a single list, and four positions to mark the beginning of the respective suits.

Source code provided:

- **Do not modify**
  - `Position.java` (as provided in the text)
  - `PositionalList.java` (as provided in the text)
  - `LinkedPositionalList.java` (as provided in the text)
  - Card.java: Represents a card with a suit and rank; declares suit and rank as `enum` type

- **Modify the following file**
    - `CardHand.java`: Represents a hand of cards as a linked positional list; supports the methods listed above