/** Interface for the ADT list 
Listing 8-1
@file ListInterface.h */

#ifndef _LIST_INTERFACE
#define _LIST_INTERFACE

template<class ItemType>
class ListInterface
{
public:
    /** Sees whether this list is empty. 
     * @return True if the list is empty; otherwise returns false. */
    virtual bool isEmpty() const = 0;

    /** Gets the current number of entries in this list. 
     * @return The integer number of entries currently in the list. */
    virtual int getLength() const = 0;

    /** Inserts an entry into this list at a given position. 
     * @pre None. 
     * @post If 1 <= position <= getLength() + 1 and the insertion is 
     * successful, newEntry is at the given position in the list, 
     * other entries are renumbered accordingly, and the returned 
     * value is true. 
     * @param newPosition The list position at which to insert newEntry. 
     * @param newEntry The entry to insert into the list. 
     * @throw PrecondViolatedExcep if insertion cannot be performed. */
    virtual void insert(int newPosition, const ItemType& newEntry) throw (PrecondViolatedExcep) = 0;

    /** Removes the entry at a given position from this list. 
     * @pre None. 
     * @post If 1 <= position <= getLength() and the removal is successful, 
     * the entry at the given position in the list is removed, other 
     * items are renumbered accordingly, and the returned value is true. 
     * @param position The list position of the entry to remove. 
     * @throw PrecondViolatedExcep if removal cannot be performed. */
    virtual void remove(int position) throw (PrecondViolatedExcep) = 0;

    /** Removes all entries from this list. 
     * @post List contains no entries and the count of items is 0. */
    virtual void clear() = 0;

    /** Gets the entry at the given position in this list. 
     * @pre 1 <= position <= getLength(). 
     * @post The desired entry has been returned. 
     * @param position The list position of the desired entry. 
     * @throw PrecondViolatedExcep if no such entry exists. */
    virtual ItemType getEntry(int position) const throw (PrecondViolatedExcep) = 0;
};
/** Replaces the entry at the given position in this list. 
   @pre 1 <= position <= getLength(). 
   @post The entry at the given position is newEntry. 
   @param position The list position of the entry to replace. 
   @param newEntry The replacement entry. 
   @throw PrecondViolatedExcep if no such entry exists. */
virtual void setEntry(int position, const ItemType& newEntry) throw 
     (PrecondViolatedExcep) = 0;
}; // end ListInterface
#endif