CS302 - Data Structures

using C++

Topic: Lists

Kostas Alexis

Specifying the ADT List

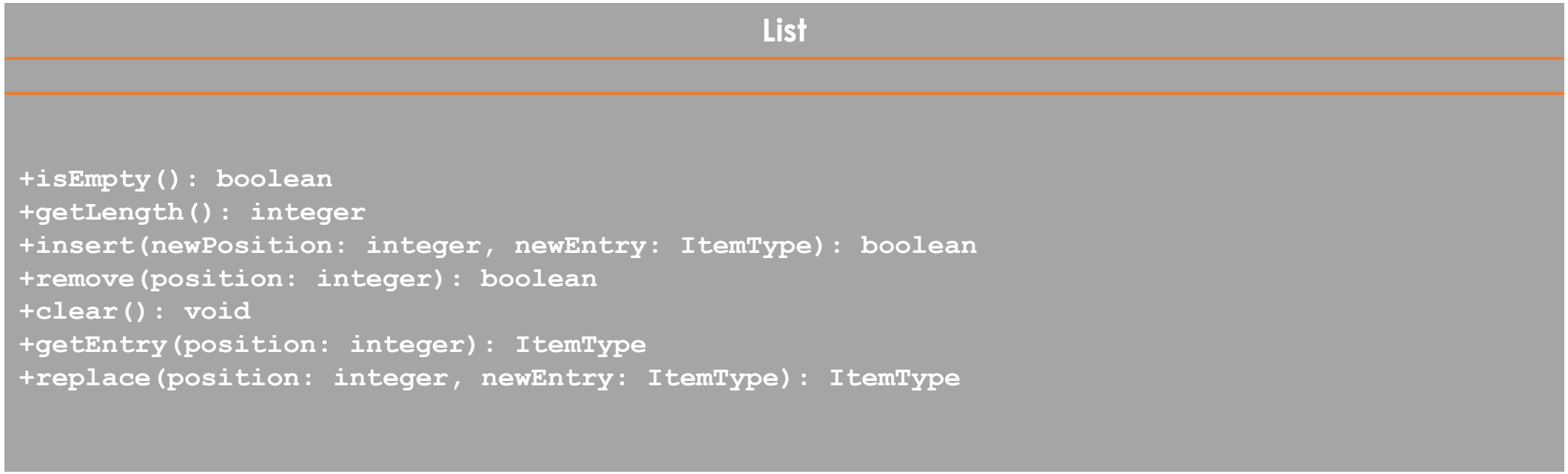
- Things you make lists of
  - Chores
  - Addresses
  - Groceries
- Lists contain items of the same type
- Operations
  - Count items
  - Add, remove items
  - Retrieve

Specifying the ADT List

- Analogous example: a grocery list

Specifying the ADT List

- A UML diagram for the ADT List



Specifying the ADT List

- Definition: ADT List
  - Finite number of objects
  - Not necessarily distinct
  - Same data type
  - Ordered by position as determined by client

Axioms for ADT List

```
1.  (List()).isEmpty = true
2.  (List()).getLength() = 0
3.  aList.getLength() = (aList.insert(i, item)).getLength() – 1
4.  aList.getLength() = (aList.remove(i)).getLength() + 1
5.  (aList.insert(i, item)).isEmpty() = false
6.  (List()).remove(i) = false
7.  (aList.insert(i, item)).remove(i) = true
8.  (aList.insert(i, item)).remove(i) = aList
9.  (List()).getEntry(i) => error
10. (aList.insert(i, item)).getEntry(i) = item
11. aList.getEntry(i) = (aList.insert(i, item)).getEntry(i+1)
12. aList.getEntry(i+1) = (aList.remove(i)).getEntry(i)
13. (List()).replace(i, item) => error
14. (aList.replace(i, item)).getEntry(i) = item
```

Using the List Operations

- Displaying the items on a list

```
// Displays the items on the list aList
displayList(aList)
{
     for (position = 1 through aList.getLength())
     {
          dataItem = aList.getEntry(position)
          Display dataItem
     }
}
```

Using the List Operations

- Replacing an item

```
// Replaces the i-th entry in the list aList with newEntry
// Returns true if the replacement was successful; otherwise return false
replace(aList, I, newEntry)
{
     success = aList.remove()
     if (success)
          success = aList.insert(i, newEntry)
     return success
}
```

Interface Template for ADT List

```
/** ADT list: Link-based implementation
    @file StackInterface.h */

#ifndef LIST_INTERFACE_
#define LIST_INTERFACE_

template<class ItemType>
class ListInterface
{
public:
      virtual bool isEmpty() const = 0;
      virtual int getLength() const = 0;
      virtual bool insert(int newPosition, const ItemType& newEntry) = 0;
      virtual bool remove(int position) = 0;
      virtual void clear() = 0;
      virtual ItemType replace(int position, const ItemType& newEntry) =
      0;
      virtual ~StackInterface() { }

      virtual ~StackInterface() { }
};    // end ListInterface
#endif
```

Thank you