Sorted Lists

CS110C

Max Luttrell, CCSF

sorted list

- in the ADT list, each entry is stored in a position, but the entries are not necessarily in any particular order
  - grocery list
  - list of cities to visit on a tour of Asia
- an **ADT sorted list**'s entries are stored in **sorted order** at all times
  - list of students in CS110C, sorted by name
  - list of Bay to Breakers race finishers, sorted by finish time

list ADT - review

list ADT operations

- is the list empty?
- get the number of entries
- insert an entry at a given position
- remove an entry at a given position
- get the entry at a given position
- set the entry at a given position
- remove all entries from the list

Cañada College
City College of San Francisco
College of San Mateo
De Anza College
Foothill College
Skyline College

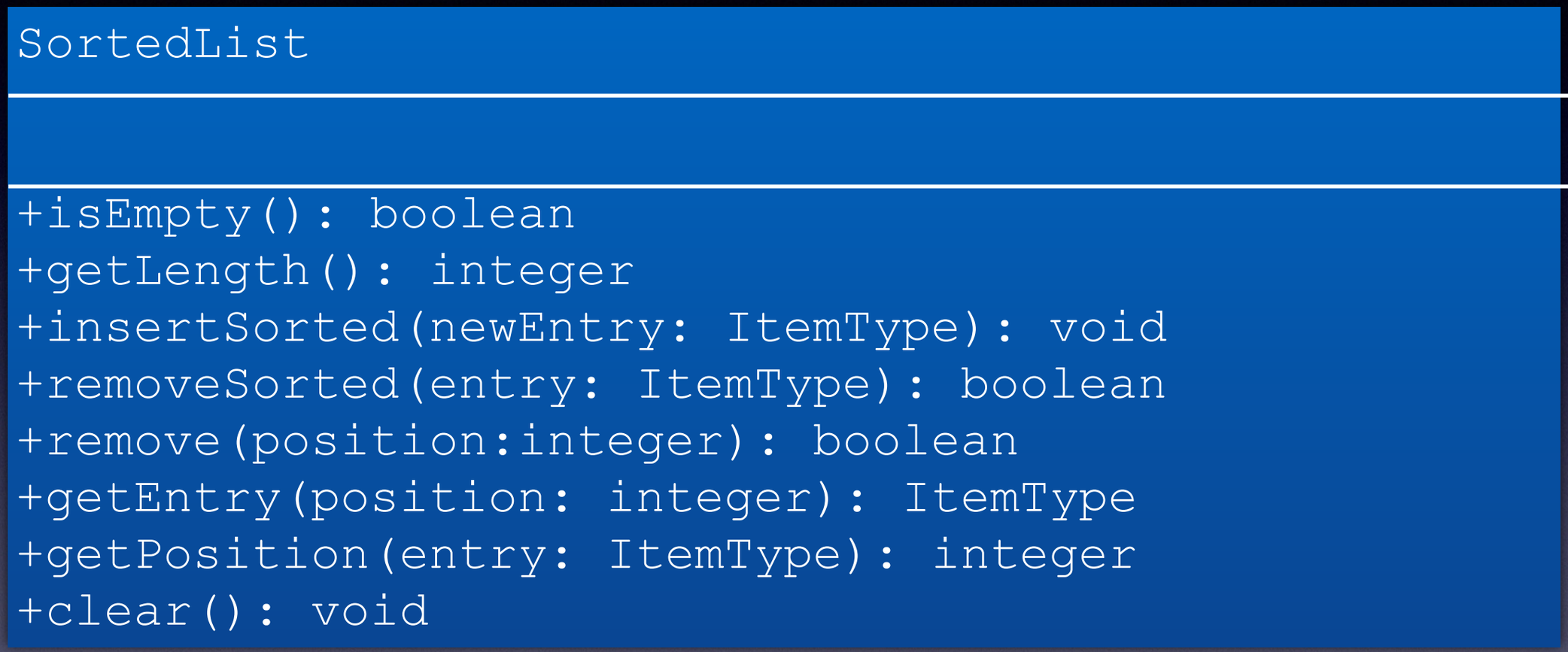
sorted list ADT

- the sorted list ADT is similar to a list ADT, but with some list ADT operations removed, and three new operations added

sorted list ADT operations

- is the list empty?
- get the number of entries
- ~~insert an entry at a given position~~
- **insert an entry at correct position**
- remove an entry at a given position
- **remove a given entry**
- get the entry at a given position
- ~~set the entry at a given position~~
- remove all entries from the list
- **get the position of a given entry**

sorted list ADT UML



new methods

- `insertSorted(newEntry: ItemType): void`
  - insert newEntry into the sorted list in its proper position so that the list remains sorted.
- `removeSorted(entry: ItemType): boolean`
  - remove the first (or only) occurrence of entry from the sorted list
  - return true if entry was located and removed, false if not.
- `getPosition(entry: ItemType): integer`
  - return the first (or only) position where entry occurs in the sorted list. If entry is not in the list, return the position where it would occur, but as a negative integer.

```
template<class ItemType>
class SortedListInterface
{
public:
   /** Inserts an entry into this sorted list in its proper order
       so that the list remains sorted.
   virtual void insertSorted(const ItemType& newEntry) = 0;

   /** Removes the first or only occurrence of the given entry from this
       sorted list.
   @return  True if removal is successful, or false if not. */
   virtual bool removeSorted(const ItemType& anEntry) = 0;

   /** Gets the position of the first or only occurrence of the given
       entry in this sorted list. In case the entry is not in the list,
       determines where it should be if it were added to the list.
   @return  Either the position of the given entry, if it occurs in the
       sorted list, or the position where the entry would occur, but as a
       negative integer. */
   virtual int getPosition(const ItemType& anEntry) const = 0;

   // The following methods are the same as those given in ListInterface
   virtual bool isEmpty() const = 0;
   virtual int getLength() const = 0;
   virtual bool remove(int position) = 0;
   virtual void clear() = 0;
   virtual ItemType getEntry(int position) const = 0;
}; // end SortedListInterface
```

sorted list ADT example

```
SortedListInterface<string>* nameListPtr = new SortedList<string>();

nameListPtr->insertSorted("Jamie");
nameListPtr->insertSorted("Brenda");
nameListPtr->insertSorted("Mary");

nameListPtr->getPosition("Mary");
nameListPtr->getPosition("Tang");
nameListPtr->getEntry(2);

nameListPtr->remove(2);
nameLIstPtr->remove("Brenda");
```

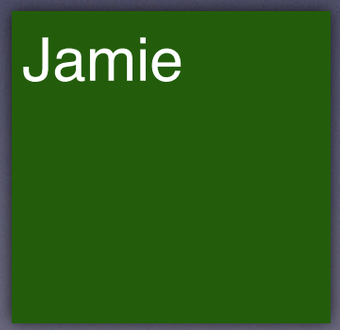
sorted list ADT example

```
SortedListInterface<string>* nameListPtr = new SortedList<string>();

nameListPtr->insertSorted("Jamie");
nameListPtr->insertSorted("Brenda");
nameListPtr->insertSorted("Mary");

nameListPtr->getPosition("Mary");
nameListPtr->getPosition("Tang");
nameListPtr->getEntry(2);

nameListPtr->remove(2);
nameLIstPtr->remove("Brenda");
```



sorted list ADT example

```
SortedListInterface<string>* nameListPtr = new SortedList<string>();

nameListPtr->insertSorted("Jamie");
nameListPtr->insertSorted("Brenda");
nameListPtr->insertSorted("Mary");

nameListPtr->getPosition("Mary");
nameListPtr->getPosition("Tang");
nameListPtr->getEntry(2);

nameListPtr->remove(2);
nameLIstPtr->remove("Brenda");
```

Brenda
Jamie

sorted list ADT example

```
SortedListInterface<string>* nameListPtr = new SortedList<string>();

nameListPtr->insertSorted("Jamie");
nameListPtr->insertSorted("Brenda");
nameListPtr->insertSorted("Mary");

nameListPtr->getPosition("Mary");
nameListPtr->getPosition("Tang");
nameListPtr->getEntry(2);

nameListPtr->remove(2);
nameLIstPtr->remove("Brenda");
```

Brenda
Jamie
Mary

sorted list ADT example

```
SortedListInterface<string>* nameListPtr = new SortedList<string>();

nameListPtr->insertSorted("Jamie");
nameListPtr->insertSorted("Brenda");
nameListPtr->insertSorted("Mary");

nameListPtr->getPosition("Mary");
nameListPtr->getPosition("Tang");
nameListPtr->getEntry(2);

nameListPtr->remove(2);
nameLIstPtr->remove("Brenda");
```

Brenda
Jamie
Mary

sorted list ADT example

```
SortedListInterface<string>* nameListPtr = new SortedList<string>();

nameListPtr->insertSorted("Jamie");
nameListPtr->insertSorted("Brenda");
nameListPtr->insertSorted("Mary");

nameListPtr->getPosition("Mary"); // 3
nameListPtr->getPosition("Tang");
nameListPtr->getEntry(2);

nameListPtr->remove(2);
nameLIstPtr->remove("Brenda");
```

Brenda
Jamie
Mary

sorted list ADT example

```
SortedListInterface<string>* nameListPtr = new SortedList<string>();

nameListPtr->insertSorted("Jamie");
nameListPtr->insertSorted("Brenda");
nameListPtr->insertSorted("Mary");

nameListPtr->getPosition("Mary"); // 3
nameListPtr->getPosition("Tang");
nameListPtr->getEntry(2);

nameListPtr->remove(2);
nameLIstPtr->remove("Brenda");
```

Brenda
Jamie
Mary

sorted list ADT example

```
SortedListInterface<string>* nameListPtr = new SortedList<string>();

nameListPtr->insertSorted("Jamie");
nameListPtr->insertSorted("Brenda");
nameListPtr->insertSorted("Mary");

nameListPtr->getPosition("Mary"); // 3
nameListPtr->getPosition("Tang"); // -4
nameListPtr->getEntry(2);

nameListPtr->remove(2);
nameLIstPtr->remove("Brenda");
```

Brenda
Jamie
Mary

sorted list ADT example

```
SortedListInterface<string>* nameListPtr = new SortedList<string>();

nameListPtr->insertSorted("Jamie");
nameListPtr->insertSorted("Brenda");
nameListPtr->insertSorted("Mary");

nameListPtr->getPosition("Mary"); // 3
nameListPtr->getPosition("Tang"); // -4
nameListPtr->getEntry(2);

nameListPtr->remove(2);
nameLIstPtr->remove("Brenda");
```

Brenda
Jamie
Mary

sorted list ADT example

```
SortedListInterface<string>* nameListPtr = new SortedList<string>();

nameListPtr->insertSorted("Jamie");
nameListPtr->insertSorted("Brenda");
nameListPtr->insertSorted("Mary");

nameListPtr->getPosition("Mary"); // 3
nameListPtr->getPosition("Tang"); // -4
nameListPtr->getEntry(2);         // Jamie

nameListPtr->remove(2);
nameLIstPtr->remove("Brenda");
```

Brenda
Jamie
Mary

sorted list ADT example

```
SortedListInterface<string>* nameListPtr = new SortedList<string>();

nameListPtr->insertSorted("Jamie");
nameListPtr->insertSorted("Brenda");
nameListPtr->insertSorted("Mary");

nameListPtr->getPosition("Mary"); // 3
nameListPtr->getPosition("Tang"); // -4
nameListPtr->getEntry(2);         // Jamie

nameListPtr->remove(2);
nameLIstPtr->remove("Brenda");
```

Brenda
Jamie
Mary

sorted list ADT example

```
SortedListInterface<string>* nameListPtr = new SortedList<string>();

nameListPtr->insertSorted("Jamie");
nameListPtr->insertSorted("Brenda");
nameListPtr->insertSorted("Mary");

nameListPtr->getPosition("Mary"); // 3
nameListPtr->getPosition("Tang"); // -4
nameListPtr->getEntry(2);         // Jamie

nameListPtr->remove(2);
nameLIstPtr->remove("Brenda");
```

Brenda
Mary

sorted list ADT example

```
SortedListInterface<string>* nameListPtr = new SortedList<string>();

nameListPtr->insertSorted("Jamie");
nameListPtr->insertSorted("Brenda");
nameListPtr->insertSorted("Mary");

nameListPtr->getPosition("Mary"); // 3
nameListPtr->getPosition("Tang"); // -4
nameListPtr->getEntry(2);         // Jamie

nameListPtr->remove(2);
nameLIstPtr->remove("Brenda");
```

Brenda
Mary

sorted list ADT example

```
SortedListInterface<string>* nameListPtr = new SortedList<string>();

nameListPtr->insertSorted("Jamie");
nameListPtr->insertSorted("Brenda");
nameListPtr->insertSorted("Mary");

nameListPtr->getPosition("Mary"); // 3
nameListPtr->getPosition("Tang"); // -4
nameListPtr->getEntry(2);         // Jamie

nameListPtr->remove(2);
nameLIstPtr->remove("Brenda");
```