Miniproject

Generated by Doxygen 1.10.0

Chapter 1

Class Index

1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 2

Class Documentation

2.1 LinkedList::Iterator Class Reference

Public Member Functions

- **Iterator** (Node ∗ptr)
- int & operator∗ () const

  Dereferences the iterator to access the current value.
- Iterator & operator++ ()

  Pre-increments the iterator.
- Iterator operator++ (int)

  Post-increments the iterator.
- bool operator== (const Iterator &other) const

  Compares two iterators for equality.
- bool operator!= (const Iterator &other) const

  Compares two iterators for inequality.

2.1.1 Member Function Documentation

2.1.1.1 operator"!=()

```
bool LinkedList::Iterator::operator!= (
            const Iterator & other ) const
```

Compares two iterators for inequality.

Parameters

other	The iterator to compare with.

Returns

True if the iterators are not equal, false otherwise.

2.1.1.2 operator*()

```
int & LinkedList::Iterator::operator* ( ) const
```

Dereferences the iterator to access the current value.

Returns

A reference to the current value.

2.1.1.3 operator++() [1/2]

```
LinkedList::Iterator & LinkedList::Iterator::operator++ ( )
```

Pre-increments the iterator.

Returns

A reference to the incremented iterator.

2.1.1.4 operator++() [2/2]

```
LinkedList::Iterator LinkedList::Iterator::operator++ (
            int  )
```

Post-increments the iterator.

Returns

A copy of the iterator before incrementing.

2.1.1.5 operator==()

```
bool LinkedList::Iterator::operator== (
            const Iterator & other ) const
```

Compares two iterators for equality.

Parameters

other	The iterator to compare with.

Returns

True if the iterators are equal, false otherwise.

2.2 LinkedList Class Reference

Classes

- class Iterator

Public Member Functions

- **LinkedList** ()

  Default constructor. Constructs an empty linked list.
- LinkedList (const std::vector< int > &values)

  Constructs a linked list from a vector of integers.
- void append (int value)

  Appends a new value to the end of the linked list.
- void remove (int value)

  Removes the first occurrence of a value from the linked list.
- int pop (int index=-1)

  Removes and returns the value at a specified index.
- void **display** ()

  Displays the linked list contents to the console.
- int search (int value)

  Searches for a value in the linked list.
- int & operator[ ] (int index)

  Accesses the value at a specified index.
- const int & operator[ ] (int index) const

  Accesses the value at a specified index (const).
- Iterator begin ()

  Returns an iterator to the beginning of the linked list.
- Iterator end ()

  Returns an iterator to the end of the linked list.

2.2.1 Constructor & Destructor Documentation

2.2.1.1 LinkedList()

```
LinkedList::LinkedList (
            const std::vector< int > & values )
```

Constructs a linked list from a vector of integers.

Parameters

values	A vector of integers to initialize the linked list with.

2.2.2 Member Function Documentation

2.2.2.1 append()

```
void LinkedList::append (
            int value )
```

Appends a new value to the end of the linked list.

Parameters

value	The value to append.

2.2.2.2 begin()

```
LinkedList::Iterator LinkedList::begin ( )
```

Returns an iterator to the beginning of the linked list.

Returns

An iterator to the first element.

2.2.2.3 end()

```
LinkedList::Iterator LinkedList::end ( )
```

Returns an iterator to the end of the linked list.

Returns

An iterator to one past the last element.

2.2.2.4 operator[]() [1/2]

```
int & LinkedList::operator[] (
            int index )
```

Accesses the value at a specified index.

Parameters

index	The index to access.

Returns

A reference to the value at the index.

Exceptions

std::out_of_range	If the index is out of range.

2.2.2.5 operator[]() [2/2]

```
const int & LinkedList::operator[] (
            int index ) const
```

Accesses the value at a specified index (const).

Parameters

index	The index to access.

Returns

A const reference to the value at the index.

Exceptions

std::out_of_range	If the index is out of range.

2.2.2.6 pop()

```
int LinkedList::pop (
            int index = -1 )
```

Removes and returns the value at a specified index.

Parameters

index	The index of the element to remove. Use -1 to remove the last element.

Returns

The removed value.

Exceptions

std::runtime_error	If the list is empty.
std::out_of_range	If the index is out of range.

2.2.2.7 remove()

```
void LinkedList::remove (
            int value )
```

Removes the first occurrence of a value from the linked list.

Parameters

value	The value to remove.

2.2.2.8 search()

```
int LinkedList::search (
            int value )
```

Searches for a value in the linked list.

Parameters

value	The value to search for.

Returns

The index of the value, or -1 if not found.

2.3 Node Struct Reference

Public Member Functions

- **Node** (int val)

Public Attributes

- int **data**
- std::unique_ptr< Node > **next**

Index