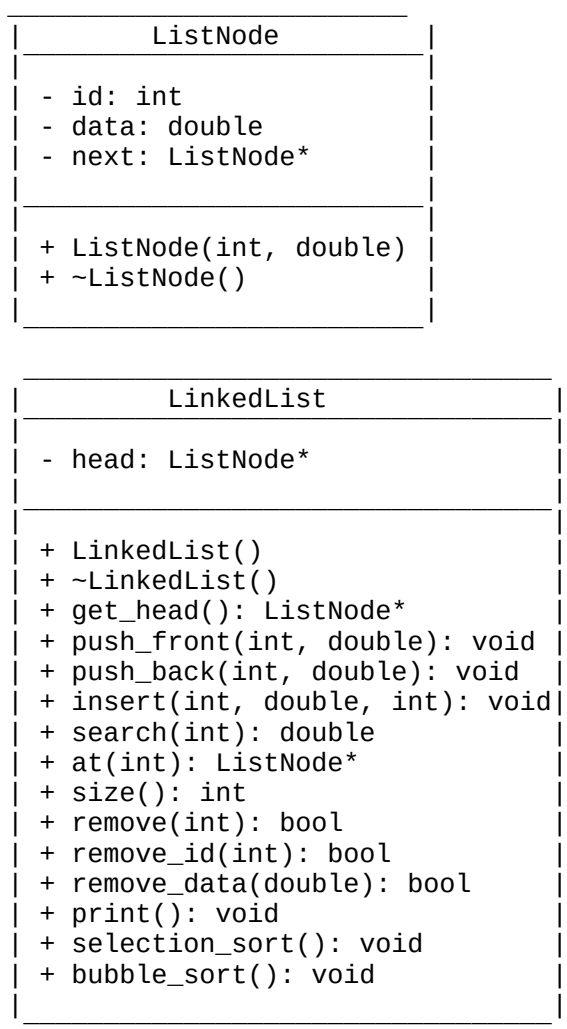
UML diagrams:



Specifcations:

Feature #1: Class ListNode

Purpose: A single Node that stores an integer ID, a double value, and a pointer to the next node.

Assumptions:

- id is an integer and is an identifier for the node.
- data is a double and is the data associated with the node.
- next points to the next node and initializes it to nullptr.

Inputs: int new_id and double new_data

Outputs: A new node object with initialized fields

State Changes:

- The ListNode constructor initializes all member variables.
- The destructor deletes the next node to clean it up.

Cases and Expected Behavior:

- Expected to create a Node
- Also expected to delete the Node

Feature #2: Class LinkedList

Purpose: Creates a list that stores ListNode objects that supports inserting, removing, searching, sorting, and modifying nodes within the linked list.

Assumptions:

- The list is will initially be empty until nodes are added.
- The list will dynamically allocate memory for new nodes.
- The list nodes are linked using pointers and point to the next node.

Inputs:

- int id and double data will be passed to most insert functions.

- int index inserts or remove a node.
- double data and int id for search/removal.

Outputs: Depending on the function
- A node pointer
- A data value
- The result of the boolean
- A printout to the console.

State Changes:
- Adding/removing nodes would change the internal structure like head and their next pointers.
- Sorting reorders nodes in place by swapping values.

Cases and Expected Behavior:
- Constructor: Initialize the list with head = nullptr.
- Destructor: Deletes the entire list.
- push_front(id, data): Adds a new node at the start of the list.
- push_back(id, data): Adds a new node at the end but traverses the list first.
- insert(id, data, index): Adds a node at a certain position.
- search(id): Returns the data of the node with the given id or -1.0 if id is not found.
- at(index): Returns a pointer to the node at the specified index or nullptr.
- size(): Returns the number of nodes in the list.
- remove(index): Deletes the node at a given position. Returns true if successful.
- remove_id(id): Deletes the first node matching the ID. Returns true if found.
- remove_data(data): Deletes the first node matching the data value. Returns true if found.
- print(): Outputs all list contents to the console and the total number of nodes.
- selection_sort(): Sorts the list in ascending order based on id using selection sort.
- bubble_sort(): Sorts the list in ascending order based on id using bubble sort.