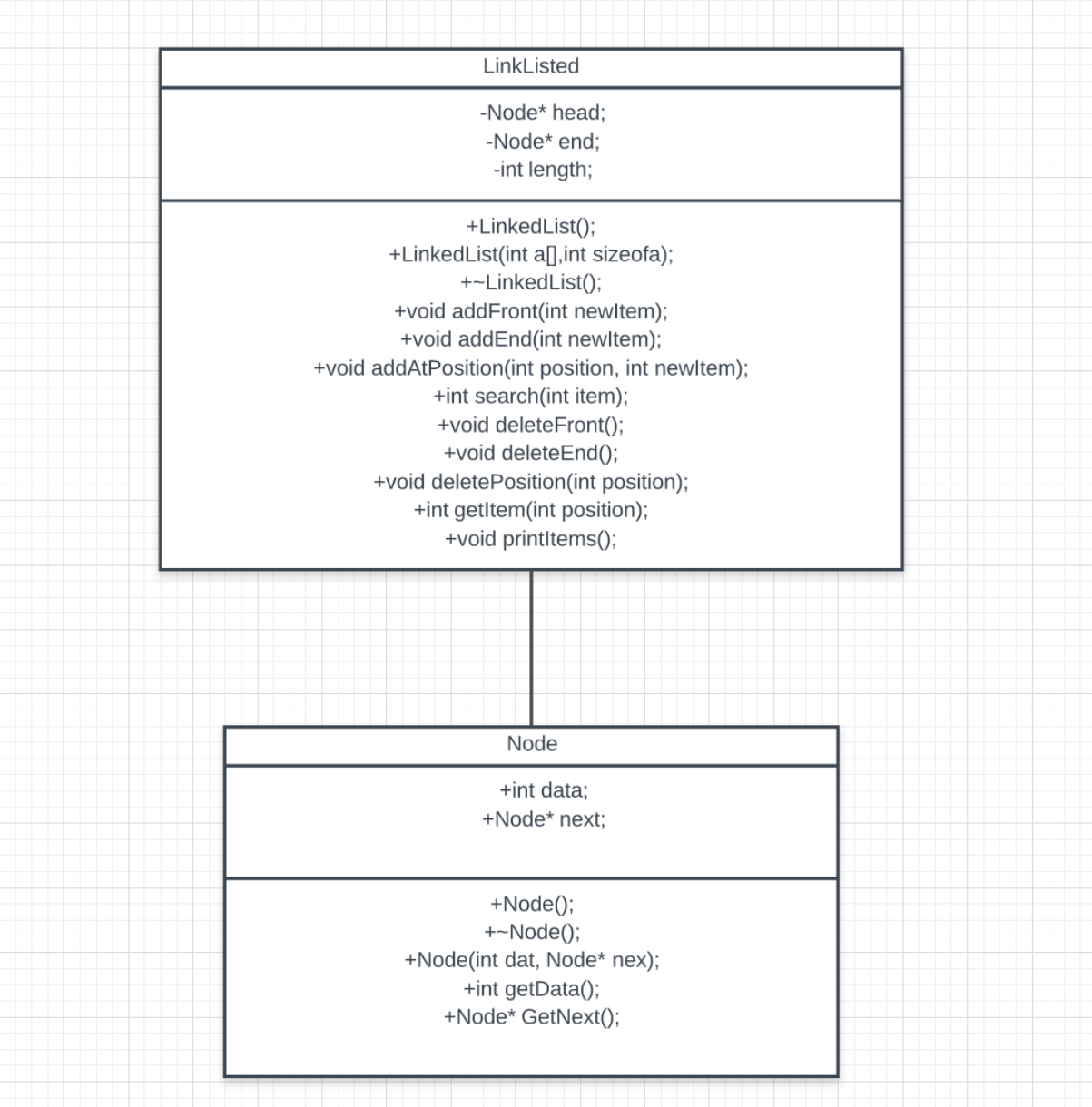
Pra7

Yian Xie
A1702241



Description:

Node class: is used to set data for Node and point to another Node;
+Node(int dat, Node* nex);//set data and point to the next Node
+int getData();//get the data
+Node* GetNext();//get the address for next Node

LinkListed class:

```
LinkedList();
```

//A constructor that takes an array of integers and makes a linked list, containing all the elements of the array, in the same order. As the second parameter, it takes the size of the array.

```
LinkedList(int a[],int sizeofa);
```

//A destructor that manually deletes all the elements that are still in the list.

```
~LinkedList();
```

//The function inserts a new node, containing the newItem, at the beginning of the list.

```
void addFront(int newItem);
```

// The function inserts a new node, containing the newItem, at the end of the list.

```
void addEnd(int newItem);
```

// The function inserts a new node, containing the newItem, such that it is the position-th member of the list.

```
void addAtPosition(int position, int newItem);
```

// The function searched the list for the item, and if found, both prints the position of the of the item (followed by a space) and returns the position of the item in the list (positions start from 1

```
int search(int item);
```

// The function deletes the first element of the list.

```
void deleteFront();
```

// The function deletes the last element of the list.

```
void deleteEnd();
```

// The function deletes the element at the given position of the list. If the position < 1 or it is larger than the size of the list, only print ”outside range”.

```
void deletePosition(int position);
```

// The function both prints the value of the item (followed by a space) and returns the value of the item at the given position of the list

```
int getItem(int position);
```

// The function prints the value of the items of the list from head to tail. In case of an empty list, it does not print anything

```
void printItems();
```

//is used to print out

Test:

Sample input: 5 2 7 11 10 AP 3 99
expected output: 5 2 99 7 11 10
output: 5 2 99 7 11 10

Sample input: 13 4 2 11 22 DP 3 0
expected output: 13 4 11 22
output: 13 4 11 22

Sample input: 45 20 222 1 GI 3 0
expected output: 2 45 20 2 1
output: 222 45 20 2 1