Introduction to OOA OOD and UML 2022 Spring
07/14

UML Practice on Algorithms (1)

Today's practice is about using UML diagrams to implement two algorithms involved in a "Linked List".

Please submit your answer file "UML_Insert_Your name.png" to "Final Practice 1" and your answer file "UML_Append_Your name.png" to "Final Practice 2" under "Assignments" tab in Manaba +R.

The deadline for "Practice 1" is by **July 14th., 2022 12:10** and the deadline for "Practice 2 is by **July 15th., 2022 12:00.** The maximum points for both "Practice 1" and "Practice 2" are **4p**.

- **Linked List:**

  A Linked List is an ordered collection that contains many objects of the same type. Data in a Linked List is stored in a sequence of containers (nodes). The list holds a reference to the first container (node) and each container (node) has a link to the next one in the sequence. Fig. 1 shows the theory of a Singly Linked list.

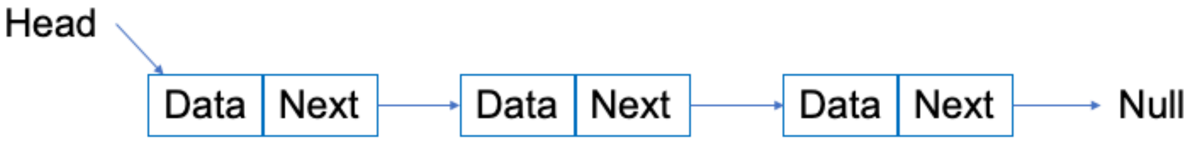

Fig. 1: The illustration of a Linked list

- **Describe a Linked List using a Class Diagram:**

  A Linked List can be represented as a class and a Node as a separate class. The LinkedList class contains a reference to the first node in the list. Fig. 2 shows the UML notations of a LinkedList class and a Node class.

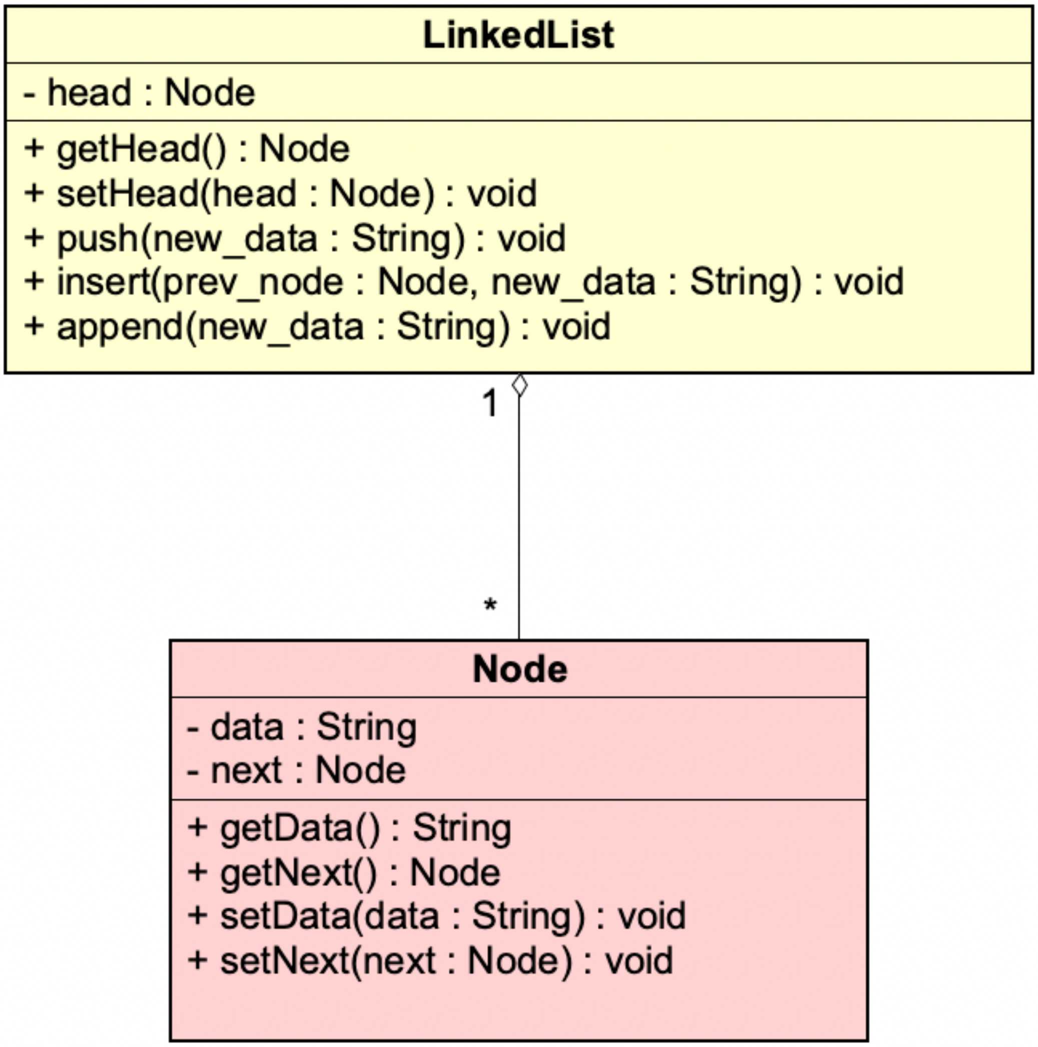
LinkedList
- head : Node
+ getHead() : Node
+ setHead(head : Node) : void
+ push(new_data : String) : void
+ insert(prev_node : Node, new_data : String) : void
+ append(new_data : String) : void
1
*
Node
- data : String
- next : Node
+ getData() : String
+ getNext() : Node
+ setData(data : String) : void
+ setNext(next : Node) : void

Fig. 2: A UML class diagram notation for a LinkedList and a Node

- **Describe the “push()” method using an Activity Diagram:**

  The “push()” method in the LinkedList class is for inserting a new node at the front of the Linked List. The last two steps of the algorithm for the “push()” method is described in the Fig. 3 and it has following four steps:

1. Allocate a node object (the object name is “new_node”)
2. Initialize the “data” of the “new_node” with the “new data”.
3. Assign the value of “head” as the reference of the “new_node”.
4. Refer the “head” to the object “new_node”.

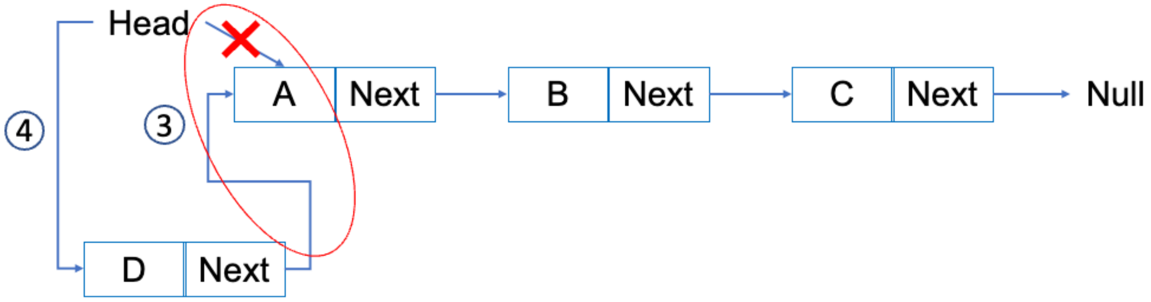

Fig. 3: An illustration of the method “push()”

Fig. 4 shows an activity diagram for describing the four steps of the “push()” based on Fig. 3.

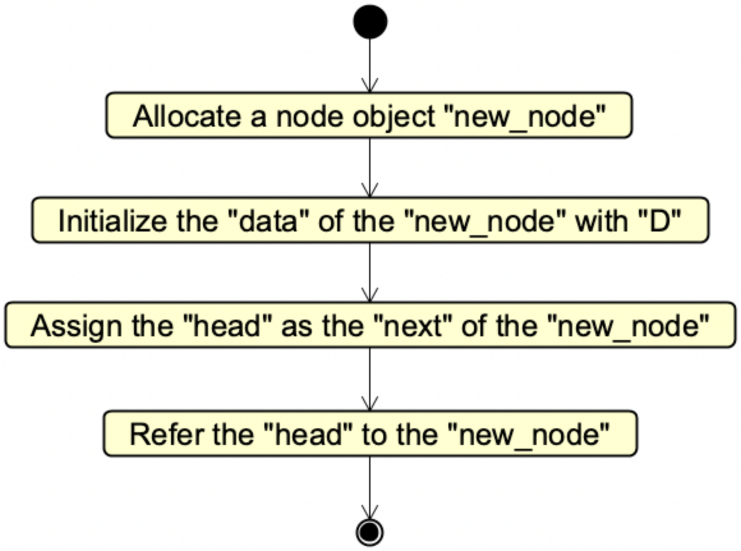
Allocate a node object "new_node"
Initialize the "data" of the "new_node" with "D"
Assign the "head" as the "next" of the "new_node"
Refer the "head" to the "new_node"

Fig. 4: The activity diagram for “push()” in Fig. 3

- **Practice 1:** The “insert()” method of the “LinkedList” class is for inserting a new node after a given node in the Linked List. The last two steps of the algorithm for the “insert()” method is described in the Fig. 5 and it has following four steps:
  1. Allocate a node object (the object name is “new_node”)
  2. Initialize the “data” of the “new_node” with the “new data”.
  3. Assign the reference of the given “previous node” as the reference of the “new_node”.
  4. Refer the reference of the given “previous node” to the object “new_node”.

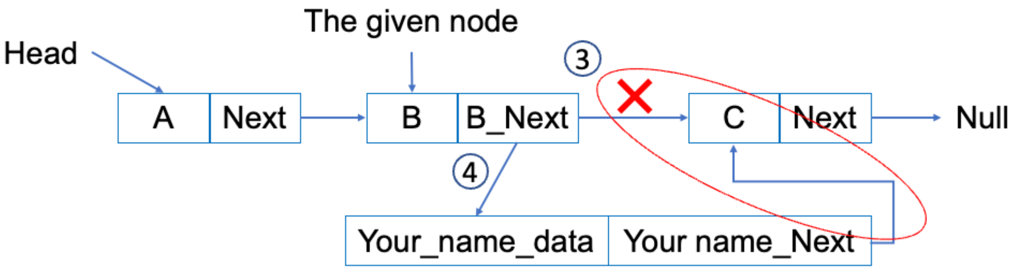

Fig. 5: An illustration of the method “insert()”

Requirements for Practice 1:

1. Please make an activity diagram for describing the “insert()” method based on Fig. 5.
2. The “new node” should be described in your activity diagram as follows: for the name of the “new node”: your name; for the data of the “new node”: your name_data; for the reference of the “new node”: “your name_next”
3. The given “previous node” should be described in your activity diagram as follows: for the name of the “previous node”: “B”; for the data of the “previous node”: “B”, for the reference of the “previous node” “B_Next”.

- **Practice 2:** The “append()” method of the “LinkedList” class is for inserting a new node at the end of the linked List. The last three steps of the algorithm for the “append()” method is described in the Fig. 6 and it has following five steps:
  1. Allocate a node object (the object name is “new_node”)
  2. Initialize the “data” of the “new_node” with the “new data”.
  3. Assign “null” as the reference of the “new_node”.
  4. Traverse the Linked List until finding the “last node”. The “last node” means the reference of the node refers to “null”.
  5. Refer the reference of the “last node” to the object “new_node”.

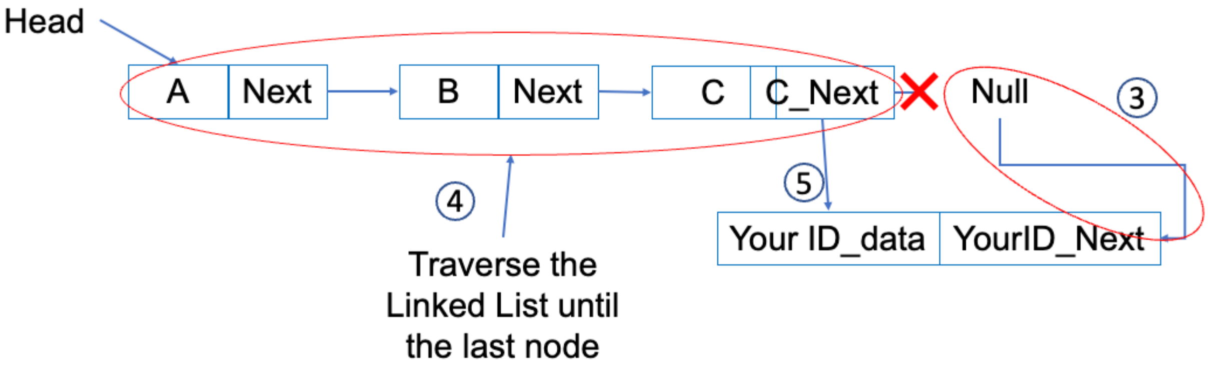

Fig. 6: An illustration of the method “append()”

Requirements for Practice 2:

1. Please make an activity diagram for describing the “append()” method based on Fig. 6.
2. The “new node” should be described in your activity diagram as follows: for the name of the “new node”: your ID; for the data of the “new node”: your ID_data; for the reference of the “new node”: “your ID_next”.
3. The step “4” has been described in the given “astah” file “LinkedList.asta”.