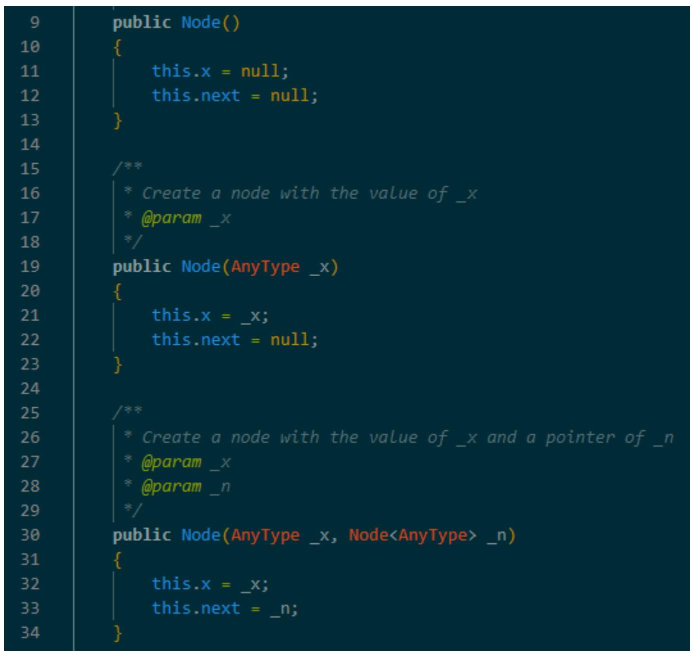
Dr. Barker’s CS 2420
Report by Weston Shakespear
Summer 2023

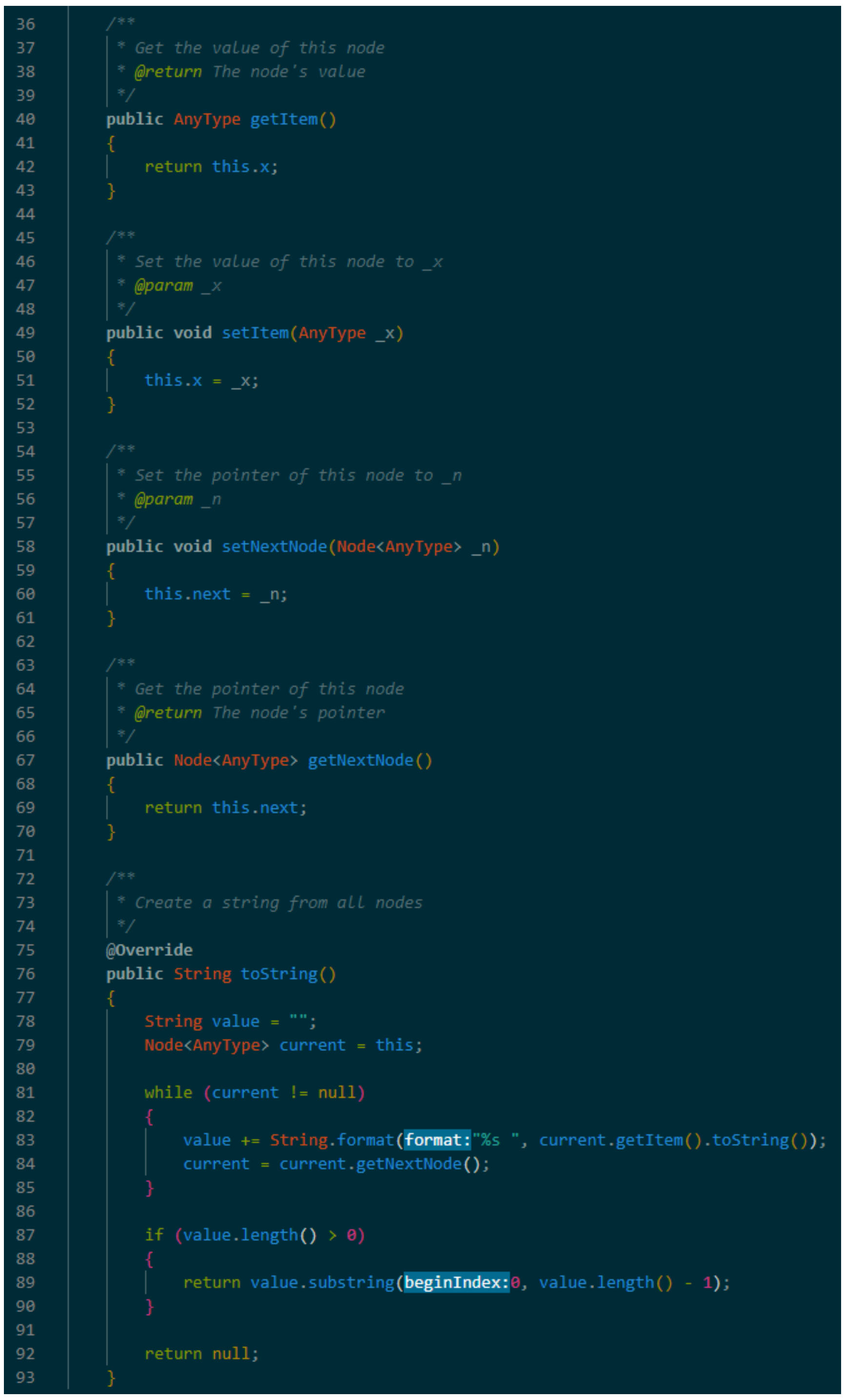
Chapter 17: Linked List Lab

Node Constructors:



```java
public Node()
{
    this.x = null;
    this.next = null;
}

/**
 * Create a node with the value of _x
 * @param _x
 */
public Node(AnyType _x)
{
    this.x = _x;
    this.next = null;
}

/**
 * Create a node with the value of _x and a pointer of _n
 * @param _x
 * @param _n
 */
public Node(AnyType _x, Node<AnyType> _n)
{
    this.x = _x;
    this.next = _n;
}
```

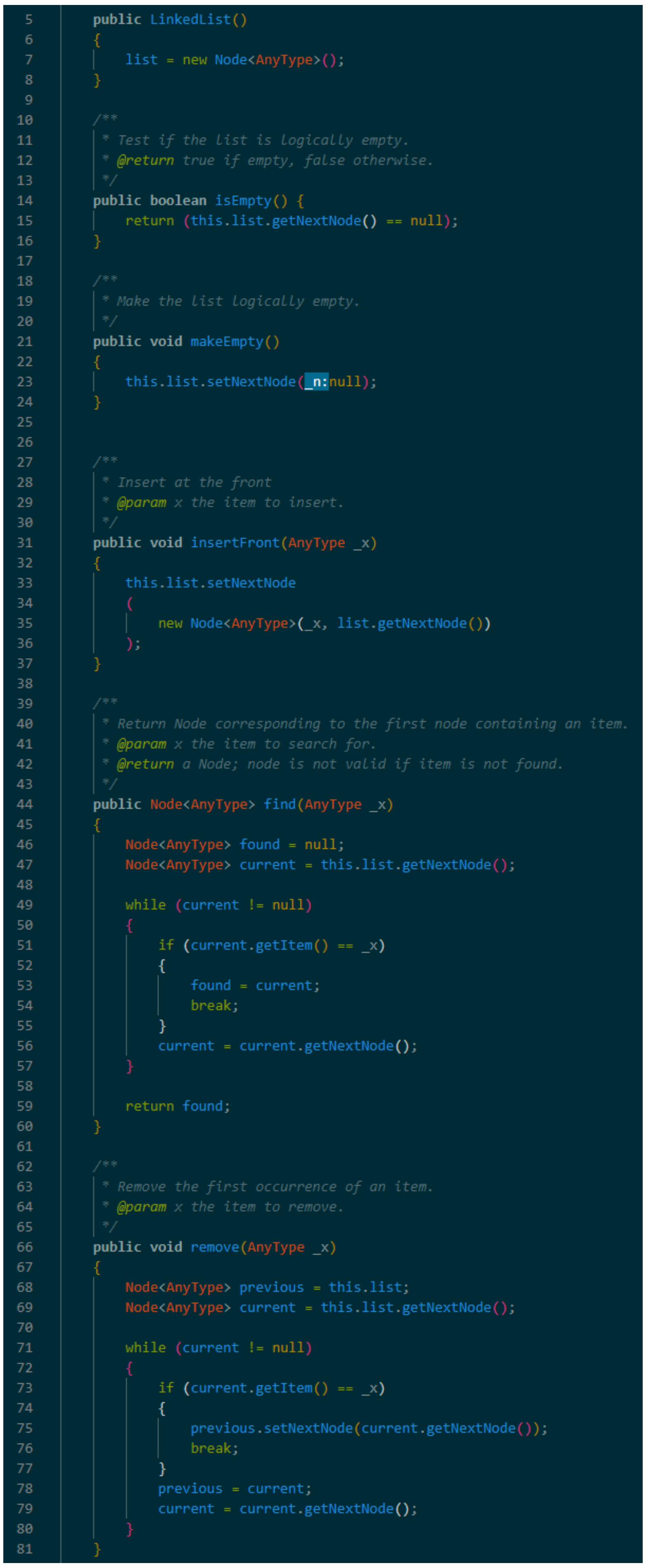
Node Methods:



```java
/**
 * Get the value of this node
 * @return The node's value
 */
public AnyType getItem()
{
    return this.x;
}

/**
 * Set the value of this node to _x
 * @param _x
 */
public void setItem(AnyType _x)
{
    this.x = _x;
}

/**
 * Set the pointer of this node to _n
 * @param _n
 */
public void setNextNode(Node<AnyType> _n)
{
    this.next = _n;
}

/**
 * Get the pointer of this node
 * @return The node's pointer
 */
public Node<AnyType> getNextNode()
{
    return this.next;
}

/**
 * Create a string from all nodes
 */
@Override
public String toString()
{
    String value = "";
    Node<AnyType> current = this;

    while (current != null)
    {
        value += String.format(format:"%s ", current.getItem().toString());
        current = current.getNextNode();
    }

    if (value.length() > 0)
    {
        return value.substring(beginIndex:0, value.length() - 1);
    }

    return null;
}
```

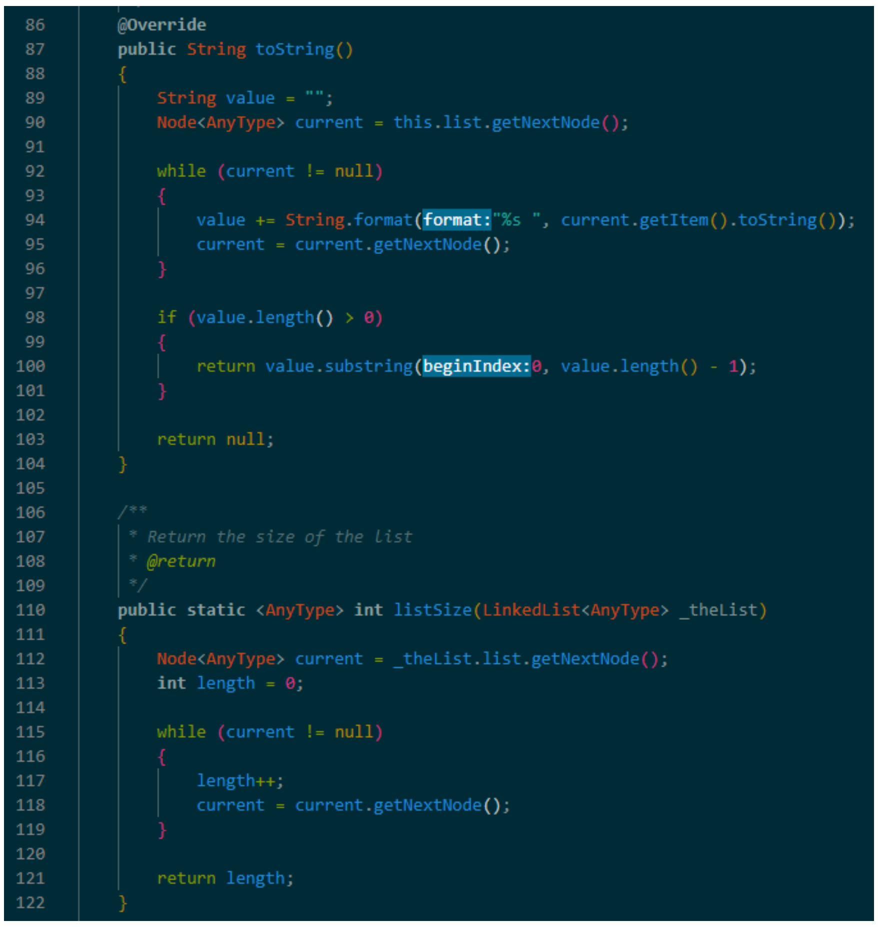
Chapter 17: Linked List Lab

Linked List Constructor and Methods:



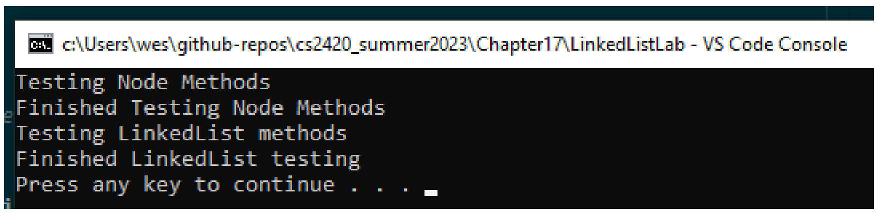
```java
    public LinkedList()
    {
        list = new Node<AnyType>();
    }

    /**
     * Test if the list is logically empty.
     * @return true if empty, false otherwise.
     */
    public boolean isEmpty() {
        return (this.list.getNextNode() == null);
    }

    /**
     * Make the list logically empty.
     */
    public void makeEmpty()
    {
        this.list.setNextNode(_n:null);
    }


    /**
     * Insert at the front
     * @param x the item to insert.
     */
    public void insertFront(AnyType _x)
    {
        this.list.setNextNode
        (
            new Node<AnyType>(_x, list.getNextNode())
        );
    }

    /**
     * Return Node corresponding to the first node containing an item.
     * @param x the item to search for.
     * @return a Node; node is not valid if item is not found.
     */
    public Node<AnyType> find(AnyType _x)
    {
        Node<AnyType> found = null;
        Node<AnyType> current = this.list.getNextNode();

        while (current != null)
        {
            if (current.getItem() == _x)
            {
                found = current;
                break;
            }
            current = current.getNextNode();
        }

        return found;
    }

    /**
     * Remove the first occurrence of an item.
     * @param x the item to remove.
     */
    public void remove(AnyType _x)
    {
        Node<AnyType> previous = this.list;
        Node<AnyType> current = this.list.getNextNode();

        while (current != null)
        {
            if (current.getItem() == _x)
            {
                previous.setNextNode(current.getNextNode());
                break;
            }
            previous = current;
            current = current.getNextNode();
        }
    }
```

Chapter 17: Linked List Lab



```java
    @Override
    public String toString()
    {
        String value = "";
        Node<AnyType> current = this.list.getNextNode();

        while (current != null)
        {
            value += String.format(format:"%s ", current.getItem().toString());
            current = current.getNextNode();
        }

        if (value.length() > 0)
        {
            return value.substring(beginIndex:0, value.length() - 1);
        }

        return null;
    }

    /**
     * Return the size of the list
     * @return
     */
    public static <AnyType> int listSize(LinkedList<AnyType> _theList)
    {
        Node<AnyType> current = _theList.list.getNextNode();
        int length = 0;

        while (current != null)
        {
            length++;
            current = current.getNextNode();
        }

        return length;
    }
```

Testing Output:



```
c:\Users\wes\github-repos\cs2420_summer2023\Chapter17\LinkedListLab - VS Code Console
Testing Node Methods
Finished Testing Node Methods
Testing LinkedList methods
Finished LinkedList testing
Press any key to continue . . .
```