8. Linked Lists :: Implementation :: DList Class :: add() Pseudocode

Here is the pseudocode for the *add*() method:

```
Method add(In: pIndex; In: pData) Returns Nothing Throws IndexOutOfBoundsException
  -- Check for an invalid index.
  If pIndex is out of bounds Then Throw IndexOutOfBoundsException
  -- Check for appending.
  If pIndex = getSize() Then
    newNode ← create new Node storing pData. Make the mPrev reference of newNode
              refer to the tail node. Make the mNext reference null.
    If the list is empty Then
      Make the mHead reference refer to newNode
    Else
      Change the mNext reference of the tail node to refer to newNode
    End If
    Change the mTail reference to refer to newNode
```

8. Linked Lists :: Implementation :: DList Class :: add() Pseudocode

```
  -- Otherwise, we are not appending.
  Else
    node ← getNodeAt(pIndex)
    newNode ← create new Node storing pData. Make the mPrev reference of newNode
              refer to the Node preceding node. Make the mNext reference of newNode
              refer to node.
    -- Check to see if we are prepending. If we were, node would not have a
    -- preceding Node.
    If $pIndex \neq 0$ Then
      Change the mNext reference of the Node preceding node refer to newNode
    End If
    Change the mPrev reference of node refer to newNode
    -- If we are prepending we have to make mHead refer to newNode.
    If $pIndex = 0$ Then
      Change the mHead reference to refer to newNode
    End If
  End If
  Increment mSize
End Method add
```