```java
/*

PUC Minas - Ciencia da Computacao     Nome: SimpleList

Autor: Axell Brendow Batista Moreira  Matricula: 631822

Versao:  1.0                          Data: 24/09/2018

*/

class SimpleList
{
    // e' importante ressaltar que essa nao e' verdadeiramente a primeira celula
    // da fila, a primeira celula fica sempre depois dessa. Caso nao exista uma
    // celula depois, a fila esta' vazia.
    Cell _head;
    Cell _last;

    public SimpleList()
    {
        _head = new Cell();
        _last = _head;
    }

    private Cell getHead()
    {
        return _head;
    }

    private void setHead(Cell first)
    {
        _head = first;
    }

    private Cell getFirst()
    {
        return getHead().getNext();
    }

    private Cell getLast()
    {
        return _last;
    }

    private void setLast(Cell last)
    {
        _last = last;
    }

    public boolean isEmpty()
    {
        return getHead() == getLast();
    }

    public int getNumberOfElements()
    {
        int numberOfElements = 0;
        Cell currentCell = getHead();
        Cell lastCell = getLast();

        while (currentCell != lastCell)
        {
            numberOfElements++;
            currentCell = currentCell.getNext();
        }

        currentCell = null;
        lastCell = null;

        return numberOfElements;
    }

    public Cell getCellOnIndex(int index)
    {
        int numberOfElements = getNumberOfElements();
        Cell cell = null;

        if (index < 0 || index >= numberOfElements)
        {
            System.out.println("[SimpleList]: Indice invalido. (index = " + index +
            ") - funcao getCellOnIndex(int)");
        }

        else if (numberOfElements > 0)
        {
            int currentIndex = 0;
            cell = getFirst();

            while (currentIndex < index)
            {
                currentIndex++;
                cell = cell.getNext();
            }
        }

        return cell;
    }

    public Object removeOnEnd()
    {
        Object removedElement = null;

        if (!isEmpty())
        {
            int numberOfElements = getNumberOfElements();

            Cell last = getLast();
            Cell previous = getCellOnIndex(numberOfElements - 2);

            removedElement = last.getElement();

            last.setNext(null);
            previous.setNext(null);
            setLast(previous);

            previous = null;
            last = null;
        }

        return removedElement;
    }

    public Object removeOnStart()
    {
        Object removedElement = null;

        if (!isEmpty())
        {
            Cell head = getHead();
            Cell first = getFirst();

            removedElement = first.getElement();

            head.setNext(null);

            setHead(first);

            first = null;
            head = null;
        }

        return removedElement;
    }

    public Object remove(int index)
    {
        int numberOfElements = getNumberOfElements();
        Object removedElement = null;

        if (index == numberOfElements - 1)
        {
            removedElement = removeOnEnd();
        }

        else if (index == 0)
        {
            removedElement = removeOnStart();
        }

        else
        {
            Cell previousOfRemovedCell = getCellOnIndex(index - 1);

            if (previousOfRemovedCell != null)
            {
                Cell removedCell = previousOfRemovedCell.getNext();
                removedElement = removedCell.getElement();

                previousOfRemovedCell.setNext(removedCell.getNext());

                removedCell.setNext(null);
                removedCell = null;
            }

            previousOfRemovedCell = null;
        }

        return removedElement;
    }

    public Object removeSecondPosition()
    {
        return remove(1);
    }

    public void addOnEnd(Object element)
    {
        Cell newCell = new Cell(element, null);

        getLast().setNext(newCell);
        setLast(newCell);

        newCell = null;
    }

    public void add(Object element, int index)
    {
        int numberOfElements = getNumberOfElements();

        if (index > numberOfElements || index < 0)
        {
            System.out.println("[SimpleList]: Indice invalido. (index =" + index + ") - funcao add(Object, int)");
        }

        else if (index == numberOfElements)
        {
            addOnEnd(element);
        }

        else
        {
            Cell previousOfCellOnIndex = index == 0 ? getHead() : getCellOnIndex(index - 1);
            Cell cellOnIndex = previousOfCellOnIndex.getNext();
            Cell newCell = new Cell(element, cellOnIndex);

            previousOfCellOnIndex.setNext(newCell);

            newCell = null;
            cellOnIndex = null;
            previousOfCellOnIndex = null;
        }
    }

    public void add(Object element)
    {
        addOnEnd(element);
    }

    public void printSimpleList()
    {
        if (!isEmpty())
        {
            Cell currentCell = getFirst();
            Cell lastCell = getLast();

            System.out.print("[ " + currentCell.getElement());

            while (currentCell != lastCell)
            {
                currentCell = currentCell.getNext();

                System.out.print(", " + currentCell.getElement());
            }

            System.out.println(" ]");
        }
    }
}
```