public boolean troca\_ter\_quarto() {

if( isEmpty() ) {

return false;

}

Node node = firstNode;

Node ter = null;

Node quarto = null;

for ( int i = 0 ; i < 4 ; i++ ) {

if( node == null ) {

return false;

} else {

if ( i == 2 ) {

ter = node;

}

if ( i == 3 ) {

quarto = node;

}

node = node.getNext();

}

}

Object aux = ter.getData();

ter.setData( quarto.getData() );

quarto.setData(aux);

return true;

}

public boolean remove\_impar() throws UnderflowException{

if( isEmpty() ) {

return false;

}

Node node = firstNode;

Node prevNode = null;

while( node != null ) {

if ( (int) node.getData() % 2 == 1) {

if( node.equals(firstNode) ){

removeFromFront();

} else if ( node.equals(lastNode) ){

removeFromBack();

} else {

prevNode.setNext(node.getNext());

}

return true;

}

prevNode = node;

node = node.getNext();

}

return false;

}