Message class - deleting a message from the messages file

We keep using the following messages. Now we implement the method that deletes a message given the message's starting record number.

	data (RECORD_DATA_LEN bytes)	**next** (pointer to next record: 4 byte int)
0	Dear Susan, how are you today? I am going to try to write you more frequently ab	2
1	Hi Johnny, do you want to go to the movies tonight? I have free passes. Suzie	-1
2	out the material that we had left open last time. It seems that the entries that	3
3	you had included in the invoices contain some calculation errors. For example, t	4
4	he first item has a price of $39.75 and there were 5 items ordered. The resultin	6
5	To everyone at the sales department: This is a reminder that our monthly luncheo	8
6	g price should be $39.75 x 5 = $198.75 but you have it showing as $201.25, and t	7
7	he tax is calculated based on this figure of $201.25. Could you please review th	9
8	n will take place at Favourite Pizza, 275 Pepperoni Road. See you all there.	-1
9	e entire worksheet and email me a new copy. I truly appreciate it. Sincerely Pet	10
10	er, Inventory Manager.	-1

Message class

Implement the following method in the Message class

```
public void deleteFromMessagesFile(int recordNumber)
```

that deletes an entire message from the messages file.

The parameter **recordNumber** is the first record of the message.

The method does not access the messages file directly but indirectly through the methods of the Record class.

Pseudo-code is below

```
// pseudo-code to delete a message

public void deleteFromMessagesFile(int recordNumber) {
     declare and instantiate an object record of the Record class

     while we have not reached an end of message
          delete the record from the messages file
          let recordNumber = next record
     }
}
```

EmailServer

For all tests be sure to:

1) Use the file _messages.dat in the PickUp folder. This file contains the table above (page 1)
2) Use the following template in the main program

```
main() {
     declare and instantiate an object of the Message class, for example message
     instantiate the availableList to an empty list

     open the messages file (use FileIO.openMessagesFile() method; do not open directly)

     if successful
          < ...
            write code here using the different tests in the table below
          ... >

          print the available list starting from the head

          close the messages file
     end if
}
```

3) check the expected results as outlined in the table below

Tests

test	conditions	code being tested	expected results
1	1 message that fits in a single record; for example, delete message at record position 1	running through loop only once	DELETED marker is now in position 0 of record 1; availableList contains record number 1
2	1 message that uses two records; for example, delete message at record position 5	running through two iterations of the loop	DELETED marker is now in position 0 of records 5 and 8; availableList contains record numbers 5 and 8
3	1 message that uses more than two records; for example, delete message at record position 0	same as above but more than two records are deleted; redundant check	DELETED marker is now in position 0 of records 0, 2, 3, 4, 6, 7, 9, 10; availableList contains record numbers 0, 2, 3, 4, 6, 7, 9, 10
4	make two separate calls to message.deleteMessage() to ensure that subsequent calls are working properly; for example, delete the message at position 0, then delete the message at position 5	subsequent calls so that availableList is not empty when the second call to deleteMessage() is made	DELETED markers are in the correct records; the availableList contains 0, 2, 3, 4, 6, 7, 9, 10, 5, 8