**CSC248 – Fundamentals of Data Structure**

**Academic Session October 2023 – February 2024**

**Lab Assignment 3 – Linked List (Built-In)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Course Outcomes (CO)** | **LO1** | **LO2** | **LO3** |
| CO1 |  |  |  |
| CO2 | √ | √ | √ |
| CO3 |  |  |  |

Given the following Friend and LinkedList ADT

public class Friend

|  |  |
| --- | --- |
| { |  |
|  | private int idno; |
|  | private String name, hpno, email; |
|  |  |
|  | public Friend (int id, String n, String hp, String em) {…} |
|  | public int getId() {…} |
|  | public String getName() {…} |
|  | public String getHP() {…{ |
|  | public String getEmail() {…} |
|  | public void setName(String n) {…} |
|  | public void setHP(String hp) {…} |
|  | public void setEmail(String mail) {…} |
| } |  |

public class LinkedList

|  |  |
| --- | --- |
| { |  |
|  | public LinkedList(); |
|  | public int size(); |
|  | public void add(int index, Object element); |
|  | public Object get(int index); |
|  | public Object set(int index, Object element); |
| } | public Object remove(int index) |

Write a Java program to:

1. Write a complete program of Friend class.

public class Friend {

    private int idno;

    private String name, hpno, email;

    public Friend(int idno, String name, String hpno, String email) {

        this.idno = idno;

        this.name = name;

        this.hpno = hpno;

        this.email = email;

    }

    public int getIdno() {

        return this.idno;

    }

    public void setIdno(int idno) {

        this.idno = idno;

    }

    public String getName() {

        return this.name;

    }

    public void setName(String name) {

        this.name = name;

    }

    public String getHpno() {

        return this.hpno;

    }

    public void setHpno(String hpno) {

        this.hpno = hpno;

    }

    public String getEmail() {

        return this.email;

    }

    public void setEmail(String email) {

        this.email = email;

    }

}

1. Write an application program by implementing a **MENU SELECTION** to do the following tasks.

* 1. Create a linked list named sList then insert some records (identify by the user) into sList. Verify each record to ensure that there is no duplicate record.

* 1. Ask user to enter id no for viewing the record. If the record exists the view it on the screen otherwise display an appropriate message
  2. Remove the record form the sList. If the slist is empty or has only one record then, display an appropriate message.

1. Update record that it’s based on idno given by the user. If the record exists replace its value of hpno and email also given by the user, otherwise display an appropriate message.
2. To print all record from the sList.
3. import java.util.LinkedList;
4. import java.util.Scanner;
5. public class Main {
6. public static void main(String[] args) {
7. Scanner strInput = new Scanner(System.in);
8. Scanner intInput = new Scanner(System.in);
9. LinkedList<Friend> sList = new LinkedList<Friend>();
10. System.out.println(
11. "1. Insert records\n2. View record\n3. Remove the record\n4. Update record\n5. Print all records\n6. Exit\n");
12. System.out.print("Enter your choice: ");
13. int choice = intInput.nextInt();
14. System.out.println();
15. while (choice != 6) {
16. switch (choice) {
17. case 1:
18. System.out.print("Enter the number of friends: ");
19. int numFriends = intInput.nextInt();
20. System.out.println();
21. for (int i = 0; i < numFriends; i++) {
22. System.out.print("Enter the ID number of friend " + (i + 1) + ": ");
23. int idno = intInput.nextInt();
24. System.out.print("Enter the name of friend " + (i + 1) + ": ");
25. String name = strInput.nextLine();
26. System.out.print("Enter the phone number of friend " + (i + 1) + ": ");
27. String hpno = strInput.nextLine();
28. System.out.print("Enter the email of friend " + (i + 1) + ": ");
29. String email = strInput.nextLine();
30. // check if idno already exists
31. if (sList.contains(new Friend(idno, name, hpno, email))) {
32. System.out.println("The data already exists. Please re-enter.");
33. System.out.println();
34. i--;
35. continue;
36. } else {
37. sList.add(new Friend(idno, name, hpno, email));
38. }
39. System.out.println();
40. }
41. break;
42. case 2:
43. System.out.print("Enter the ID number of the friend to view: ");
44. int idno = intInput.nextInt();
45. System.out.println();
46. boolean found = false;
47. for (Friend friend : sList) {
48. if (friend.getIdno() == idno) {
49. System.out.println("ID number: " + friend.getIdno());
50. System.out.println("Name: " + friend.getName());
51. System.out.println("Phone number: " + friend.getHpno());
52. System.out.println("Email: " + friend.getEmail());
53. System.out.println();
54. found = true;
55. break;
56. }
57. }
58. if (!found) {
59. System.out.println("The data does not exist.");
60. System.out.println();
61. }
62. break;
63. case 3:
64. System.out.print("Enter the ID number of the friend to remove: ");
65. idno = intInput.nextInt();
66. System.out.println();
67. found = false;
68. for (Friend friend : sList) {
69. if (friend.getIdno() == idno) {
70. sList.remove(friend);
71. System.out.println("The data has been removed.");
72. System.out.println();
73. found = true;
74. break;
75. }
76. }
77. if (!found) {
78. System.out.println("The data does not exist.");
79. System.out.println();
80. }
81. break;
82. case 4:
83. System.out.print("Enter the ID number of the friend to update: ");
84. idno = intInput.nextInt();
85. System.out.println();
86. found = false;
87. for (Friend friend : sList) {
88. if (friend.getIdno() == idno) {
89. System.out.print("Enter the new phone number of friend " + idno + ": ");
90. String hpno = strInput.nextLine();
91. System.out.print("Enter the new email of friend " + idno + ": ");
92. String email = strInput.nextLine();
93. friend.setHpno(hpno);
94. friend.setEmail(email);
95. System.out.println("The data has been updated.");
96. System.out.println();
97. found = true;
98. break;
99. }
100. }
101. if (!found) {
102. System.out.println("The data does not exist.");
103. System.out.println();
104. }
105. break;
106. case 5:
107. for (Friend friend : sList) {
108. System.out.println("ID number: " + friend.getIdno());
109. System.out.println("Name: " + friend.getName());
110. System.out.println("Phone number: " + friend.getHpno());
111. System.out.println("Email: " + friend.getEmail());
112. System.out.println();
113. }
114. break;
115. default:
116. System.out.println("Invalid choice.");
117. break;
118. }
119. System.out.println(
120. "1. Insert records\n2. View record\n3. Remove the record\n4. Update record\n5. Print all records\n6. Exit\n");
121. System.out.print("Enter your choice: ");
122. choice = intInput.nextInt();
123. System.out.println();
124. }
125. }
126. }