January 2023

Data Structurs CSC212

Project Phase 1

Table of Contents

Students:

1- Ahmed Abdulaziz Alkhulayfi , 442102783

2- Ali Abdullah Alswayed, 442101708

3- Badr Ahmed Alhomaidhi, 442106007

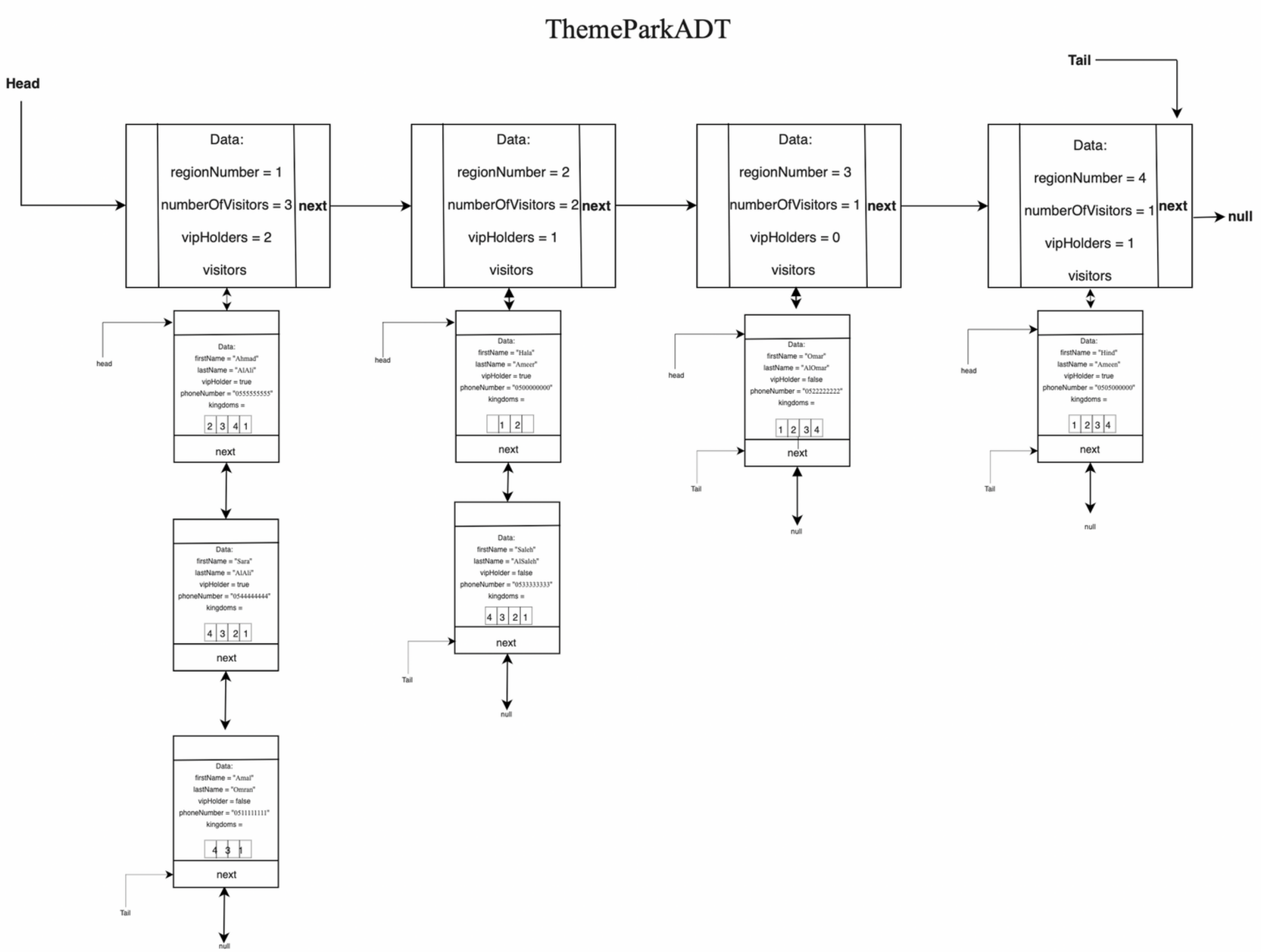
4- Nasser Khalid Alqudairy, 441101432

5- Malik Khalid Alorain, 442101540

Introduction

- Our ADT is basically a Linked List of regions inside every region another Double-Linked List of visitors sorted based on if the visitor is a VIP holder or not. If the visitor is a VIP holder, it will be added at the beginning, otherwise it will be added at the end.
- and each visitor has a number of attributes like :
  - FirstName
  - LastName
  - vipHolder
  - PhoneNumber
  - Kingdoms order

Graphical Representation



Specification

1-Method FindByLastName(String lastName)

Requires: List regions not empty.

Input: lastName.

Results: returns a list of all the visitor they have last name equals to lastName.

Output: list of all the visitors.

2-Method NumRegions()

Requires: List regions not empty.

Input: none.

Results: display the total number of regions visitors are coming from.

Output: numberOfRegions.

3-Method RegionVisitors()

Requires: List regions not empty.

Input: none.

Results: goes through the list of regions, and displays the region number and it's number of visitors.

Output: none.

4-Method NumberOfVIP(int regionNumber)

Requires: Valid region number.

Input: regionNumber.

Results: displays the number of visitors holding VIP passes from a specific region.

Output: none.

5-Method VIPLocation()

Requires: List regions not empty.

Input: none.

Results: display the current location of all visitors holding VIP passes.

Output: none.

6-Method VIPPhoneNumber(int phoneNumber1, int phoneNumber2)

Requires: none.

Inputs: phoneNumber1, phoneNumber2

Results: the method first search at the beginning of the visitors list for the two visitor then, check first if the two visitor VIPholder and from the same region:

- if true it continue and check the current kingdom if they in the same kingdom it set flag true.
- if false it display message indicating that and set flag false.

Output: flag.

7-Method SameRegion(int phoneNumber1, int phoneNumber2)

Requires: valid phone numbers.
Inputs: phoneNumber1, phoneNumber2.

Results: the method first search at the end of the list for the two visitor then check first if the two visitor not VIPholder and from the same region:

- if true it contiue and check the order of kingdoms and set flag true, if they have the same kingdoms order.
- if false it display message indicating that and set flag false.

Output: flag.

8-Method readFileAndStore(String fileName)

Requires: existing file with valid fileName.

Input: fileName.

Results: This method will open the file as input, read the text in the file and analyze it by putting each visitor in a new node dependent on his region, each input line creating a new node. First two variables put it in Name, third for Region, 4th if it’s 1 put VipHolder equals true and will be added at the beginning else it will be added at the and, 5th for phone number, and the rest for kingdoms in the same order, after it's done it will call the method sortRegions().

Output: none.

9-Method sortRegions()

Requires: List regions not empty.

Input: none.

Results: sorts the list based on the number of visitors.

Output: none.

10-Method canExit(Visitor v , String orderOfExit)

Requires: none.

Input: v, orderOfExit.

Results: this method will be called when the visitor tries to exit the park, and it checks the order of kingdoms to be in reverse.

Output: boolean flag.

Time Complexity

Operation	Big O
FindByLastName	O(n)
NumRegions	O(1)
RegionVisitors	O(r)
NumberOfVIP	O(v)
VIPLocation	O(v)
VIPPhoneNumber	O(v)
SameRegion	O(g)
readFileAndStore	O(n)
sortRegions	O(nlog n)
canExit	O(1)