

READ ME file for assignment 1

In my code, I have constructed classes **Hostel**, **Course**, **Department** which implements interface **Entity_**. then I have constructed class **Student** implementing interface **Student_**, **LinkedList<T>** implementing **LinkedList_<T>**, **Position<T>** implementing **Position_<T>**, **ListIterator<T>** implementing **java.util.iterator<T>**, **Nodelterator<T>** implementing **java.util.iterator<T>**, **CourseGrade** implementing **CourseGrade_** and so on. I have made some other classes which according to me is necessary for implementing the code efficiently. Then I have written the code for each method in each class with the return type of that very method. After that I have implemented **public class Assignment1** which have 3 functions – **private functions** **getData** and **answerQueries** and a **public static void main(String args[])**.

If I shed some light on the difference between **ListIterator** and **Nodelterator**, then ListIterator iterates over the value of the nodes of the list and returns the value of the nodes(Here position term is used in place of the Node) but the Nodelterator iterates over the Nodes/Positions of the list and returns the corresponding Node/Position. Also, I have just changed the name in the interface from iterator to ListIterator and nothing else. Both are even the same things and changing names does not create any difference and the return type is still the same. I have done the same for the Nodelterator also.

A. In the class Position<T> ,

1. I have defined a variable of type T data and an object of type Position_<T> named next. Then I have created a constructor in it which takes two arguments(T value and Position_<T> node). Then I assigned value to data and node to next.
2. I have written and tested a function called the value of T type that returns data and a function after of Position_<T> type which returns next as a result.

B. In the class LinkedList<T> ,

1. I have defined two variables of type Position_<T> named head and tail.
2. Then I have written and tested a function called add, which takes as input a T type data, in this function, I have created an object of type Position_<T> and initialized it with the argument data and tail. Thus a node is created with that value and its next pointer is tail. In this, after creating an object, I have assigned the head and tail the new object. For the first element, it is null. The second has first as next and so on. And this function returns the new object created.
3. I have written and tested a function called count of type int, which starts counting from the head of the list by head.after() and finally returns the number.
4. I have written and tested a function called positions of type Nodelterator<T> which takes the head of the list as an input and returns an iterator over it, which actually provides us the nodes/positions of the list.

C. In the class ListIterator<T> ,

1. I have created an object of type Position_<T> cursor and written a constructor which takes a list as an input and assigns cursor the head of the list.
2. I have written and tested a function called hasNext, which returns false if the cursor is null otherwise returns true.

3. I have written and created a function called next, which returns the value of the cursor and shifts the cursor to the next position of the list.

D. In the class NodeIterator<T>

1. All the functions in this class are the same as ListIterator except the next, which returns the node instead of the data of the node.

E. In the class bubbleSort,

1. I have written and created a function called sort, which takes an array of strings as an input and it compares each element of the array with all its succeeding elements and if two strings are not lexicographically arranged, then it swaps the two strings. It compares two strings by the well-defined function "compareTo". Thus after both the loops get completed, I get an array which is lexicographically sorted.

F. In the class Student,

1. I have written and created a function called name which returns the name of the student.
2. I have written and created a function called entryNo, which returns the entry number of the student.
3. I have written and created a function called hostel, which returns the hostel of the student.
4. I have written and created a function called department, which returns the department of the student.
5. I have written and created a function called completed credits, in which I have used the course list of that student which I have created in getData while reading the file. I have called an ListIterator over the course list and taken the letter grade in each course and if the grade is not I, then I have added 3 to an integer created in this function. After whole while loop, the value of the integer gave me the completed credits and I returned it in the form of a string.
6. I have written and created a function called cgpa, which again iterates over the course list and takes every grade which is not "I" and then called a function gradepoint, which returned the integer value for each letter grade when the grade is in enum format. So I have added all the grade points and multiplied it by 3 and then divided it by the completed credits and returned it in the form of a string with two digits after the decimal.
7. I have written and created a function called courseList of type ListIterator which returns an iterator over course list of the concerned student.

G. In the class Hostel or Department,

1. I have written and created a function called name, which returns the name of the entity.
2. I have written and created a function called studentsList, which returns the iterator over the list of students belonging to that entity.

H. In the class Course,

1. I have written and created a function called name, which returns the name of the entity.

3. I have written and created a function called `studentsList`, which returns the iterator over the list of students belonging to that entity. How it stores the list of students is explained in the function `getData`.

I. In the class `CourseGrade`,

1. I have written and created a function called `EntryNo`, which returns the entry number of the student related to that course.
2. I have written and tested a function called `coursenum`, which returns the course number of that course.
3. I have written and tested a function called `grade`, which returns the grade of that particular student with the given entry number in that vary course.
4. I have written and tested a function called `coursetitle`, which returns the full Course name of that course.

Explanation of implementation of `getData()` :-

1. In `getData`, I have to read student file and courses file . For this I have passed both these files as an argument in this function. In the function, I have created linked lists for storing the data of students, courses, hostels, departments, and course details.
2. In my code, firstly, with the help of a buffered reader, I have read the second file in the way, i.e., line by line. For this, I have created a linked list named CoursesDetails. I have created an object of class CourseGrade and passed all the words of the line as an argument in this object and added this object in the linked list CoursesDetails.
3. Then at the same time, I have created an object of class Course which takes the name of course from the line as an argument and added this object in the linked list allCourses if it does not exist in the list already.
4. Then, I have read the first file and created an object of type Student(class) and passed all the details of that line and the linked list CourseDetails as an argument in that object and added that object in a linked list named allStudents. I have done this for all the lines. when this is done, then in the constructor of the Student class, I have iterated over the Coursedetails list and compares the entry number of the student with the entry number of all the objects in that list, and whichever objects satisfy this, I have added that object in the linked list named studentsWithCourse. So now I can easily call an iterator over this list in the Student class.
5. At the same time, I have created objects of Hostel and Department and passed the name of the hostel and the department in them respectively. Also for each of the object of hostel and department, I have created a linked list which contains all the students in that entity. So finally I have linked lists which contains details of all the students, another contains all the hostels and all the departments which contain lists of students in that particular entity.
6. Then I have created a linked list which contains the list of all the students enrolled in that course. For that I have called an iterator over course list and for every object of it, I have called an iterator over allStudents list and compare the entry no of the two

objects and if they equal, then I have added that Student object in the students' list of that course.

Explanation of implementation of answerQueries() :-

1. In this function, I have read the third file which is a query file. For every query, I have first checked the first word and then checked its type, i.e., “INFO”, “SHARE” or “COURSETITLE”. Then accordingly, I have written my code.
2. If I will tell in details, then for “INFO” term, I have called an iterator over the list allStudents and checked the entry number of all the students with the given entry number in the line containing “INFO” and from the matching object, I have stored its name, entry number, cgpa, hostel and department in a string A. Then I called its course list and stored all its course with the grade in them combined in an array and sorted it in lexicographical order and finally added all these details to another string B. Then I have added both strings A and B. Then I have added that string in a linked list named **megaList**.
3. If the first word of the line is “SHARE”, then I have checked the third word whether it is hostel or department or course and then taken the list of all the students in that entity and stored them except the one given in the query in a linked list. Then I have taken all the elements from the list and put them in an array and sorted the array and then added the elements from the array to a string and added that string in the megaList.
4. If the first word is “COURSETITLE”, then I have called an iterator over the list CourseDetails and checked the course number of each with the given course number in the line and added the satisfying course name in a list. Then I have done the same procedure for it as I have done for the other two types of lines.
5. Finally, I have iterated over the megaList and printed all the elements. My linked list adds a new element at the head of the list so, automatically all the elements are added in reverse order. So all I have to do is just print all the elements directly.
6. Then I have written my main function with the argument as (String [] args) and called both the private functions in it with the arguments args[0] and args[1] in first function and args[2] in second function.

