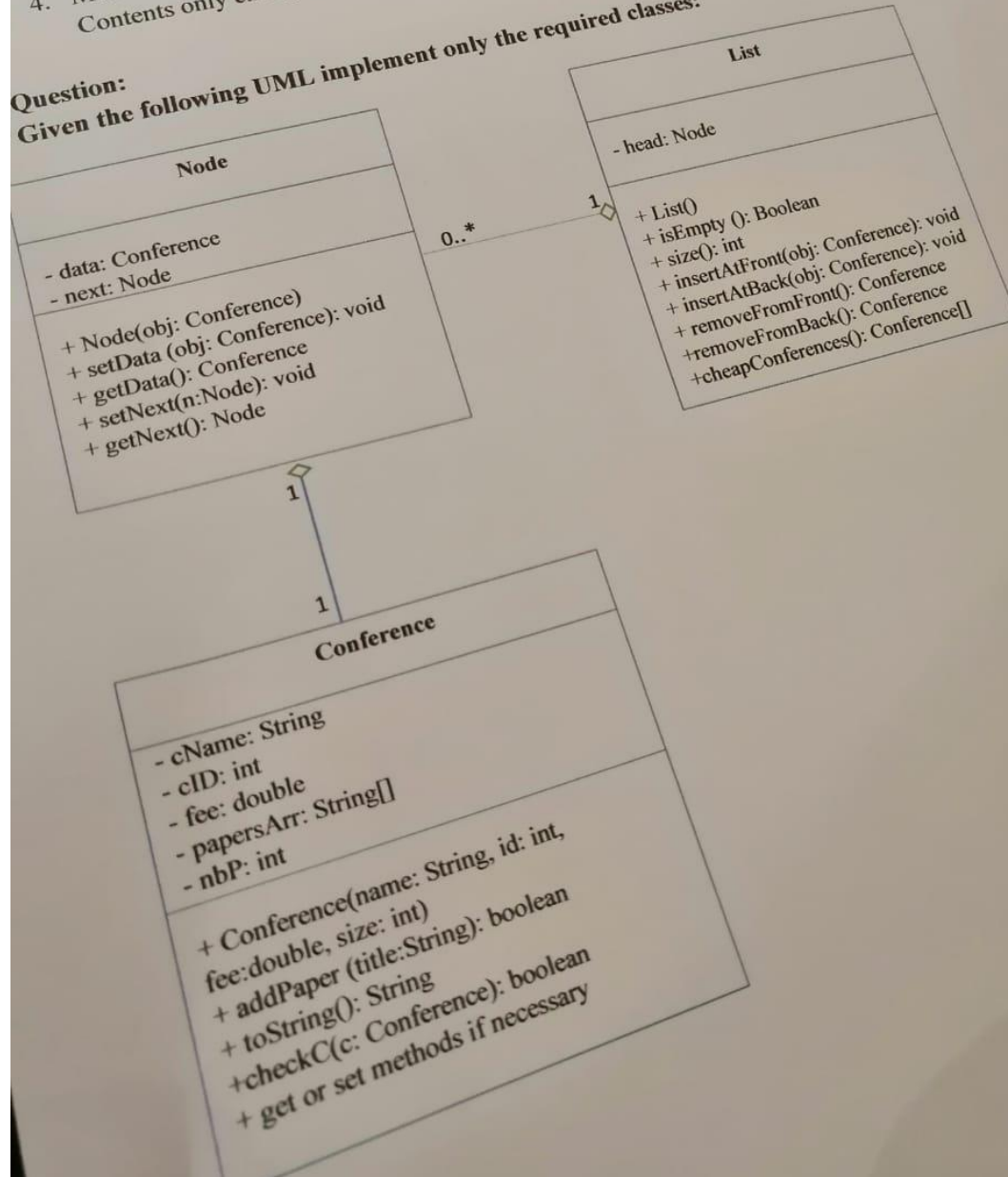


- Instructions:**
- Important:** Create Folder in the desktop with your name and all your work in it.
 - The classes *List* and *Node* have already been implemented for you. You can find them in folder "CSC113 final lab" on the C:\ drive. Copy them in your folder.
 - You **must** make use of existing methods when appropriate.
 - Methods like print, save, check should **NOT** change the contents of lists and/or array. Contents only change when objects are added or removed.

Question:

Given the following UML implement only the required classes:



es List and Node have already been implemented for you, you need to use them properly.
efine a new checked exception of type *InvalidFeeException*.
mplement Class *Conference*:

Attributes

- **cName: String**: The name of the Conference.
- **cID: int**: The id of the Conference.
- **fee: double**: Conference's fee.
- **paperArr: String[]**: An array containing title of papers participating in the Conference.
- **nbP: int**: An integer referring to the actual number of papers participating in the Conference.

Methods

- **Conference(name: String, id: int, fee: double, size: int)**: Constructor to initialize the Conference information.
 - **Hint: size** is the length of the array.
 - An *InvalidFeeException* should be generated and thrown to the calling environment if the fee is less than zero or if it is more than 9999.
- **addPaper(title: String): boolean**: Adds the received paper's title to the first empty location in paperArr. Return true if the paper is added successfully, false otherwise.
- **checkC(c: Conference): boolean**: Compares this Conference to the received Conference. The result is true if and only if *c* is not null and *c* has the same *id* and *name* as the current Conference.
- **toString(): String** Returns the Conference's information (including all papers) in a string.

3. Implement the method *cheapConferences()* in class *List*:

This method returns an array containing the conferences (from the list) which have a fee less than 1000 SR.

e a new Application class named (TestList) with a main method to perform the following tasks:
 Using the class List, create a stack of Conferences named *cList*.
 Hint: In Stack, the order is Last In First Out (LIFO).

- b. Add two conferences to the list. Ask the user to enter their information. If the fee is invalid allow the user to re-enter until a correct value is read. (You need to use the appropriate methods).

- c. Create a conference object name *con3* with the following information and add it to *cList*:

<i>cName</i>	Artificial Intelligence International Conference
<i>cID</i>	52114
<i>fee</i>	3000
<i>paperArr</i>	{"Bioinformatics", "Cognitive systems"}
<i>size</i>	10

- d. Add a new paper named "Ethics in AI" to the *paperArr* in the conference with ID 22112

- e. For each Conference in *cList* do the following:

- Print the Conferences' information.
- Save the Conference in an object file "Conferences.data".

- f. Read the information of new conferences from a text file named "newConf.txt" and add the conferences to *cList*. Each line in the file contains the information of a single conference ordered as follows: name (String) ID (int) fee (double) size (int).

Notes:

- All conferences in the file have an empty papers array.
- Your program should work properly regardless of the number of conferences in the file.

- g. Print the information of the conferences that have a fee less than 1000 SR.
- h. Display a meaningful message indicating if the conference *con3* exists in *cList* or not.

Note: Your main method should handle all possible exceptions by displaying a meaningful message.