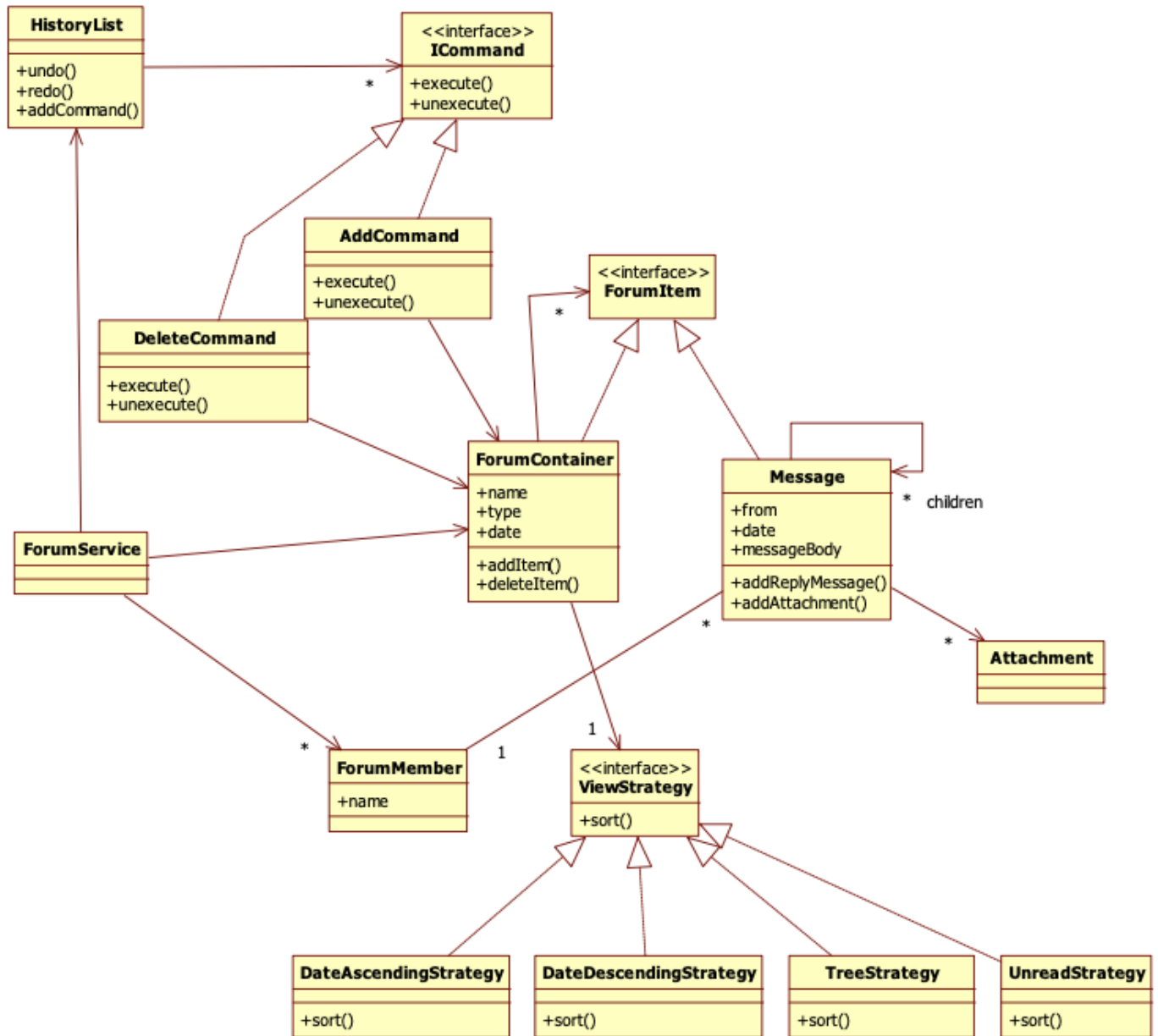**Question 1 [50 points] {60 minutes}**

In this DE course we used the message forum functionality of Sakai. In the Sakai forums section you are able to create new forums, delete forums and view forums. For every forum you can create, delete and view topics. For every topic you can create, delete and view threads. For every thread you can create, delete and view messages. You can reply to messages and you can add multiple attachments to messages.

We need to design a general Message-Forum framework. This framework should contain the following functionality

- In the Sakai forums we have a fixed structure where we have multiple forums that contain multiple topics that contain multiple threads that contain multiple messages. The message forum framework should support more flexibility. One application migh support only 1 forum with multiple threads, and another application might support multiple forums that contain multiple sub-forums, that contain multiple threads, etc.
- We should be able to undo and/or redo all add or delete actions on forums, topics, threads, sub-forums, etc.
- In the Sakai forums, we are also able to sort the content of a thread in 4 different ways sort as a tree structure, sort by date ascending, sort by date descending and sort by unread (show the unread messages first). The message forum framework should support this functionality for forums, topics, threads, sub-forums, etc. Applications build with this framework should be able to add their own sorting functionality
- It should be possible to see the tree structure of messages, which starts with the first message in a thread followed by the reply messages for this message, and these reply messages can contain other reply messages, etc.

Draw a **class diagram** of the design of the message forum framework. **Make sure you add all necessary UML elements (attributes, methods, multiplicity, etc) to communicate the important parts of your design. In your design you don't have to worry about database access.**

Command pattern, Composite pattern and Strategy pattern

Question 2 [50 points] {60 minutes}

We need to design a rental framework which can be used to write rental applications such as a DVD rental application or a car rental application. The framework has the following requirements:

- We should be able to add new rentals, delete rentals and search rentals.

- For every rental we need to store the name, phone, email, street, city and zip code of the customer. We also need to keep track of the start date of the rental, the maximum duration of the rental and the end date of the rental.
- The framework should also support the functionality reserve products that you can rent. For every reservation you make, you need to keep track of the date you reserved it.
- The framework should also allow for categorizing the products that you can rent. So you can have for example a category "new releases" for a DVD rental application, and this category contains sub-categories like "drama" of "romance"
- The framework should also support different algorithms to compute the rental fee.
- The framework should support functionality for handling the events when customers don't return their products on time. Every rental has a start date and a maximum duration. If the maximum duration has passed, the rental application should be able to handle this by sending an email, sending an SMS, writing to a log file, etc. The framework should have the functionality that an event is generated when the maximum duration has passed, and the application class should handle the desired action for the particular application.
- Rental shops often have more than one copy of the same product. A DVD rental shop might have 4 copies of one DVD title. A car rental might have 4 Ford Mustang cars. The rental framework should support this.

Now we want to use this rental framework to implement a DVD rental application. This DVD rental application has the following additional requirements:

- If the maximum duration of the DVD rental has passed, the rental application should send an email.
- We have 2 different formulas for computing the rental fee, one for new releases and one for all the others (not new releases)
- The DVD rental application should support international addresses for customers, so for every customer, we should also store the country (besides the street, city and zip).