

Read this first!!

When doing the exercises, it is advised that you make notes! These notes will be useful for the assignment you have to write later in the course.

Note: These assignments will be part of a mandatory hand-in.

Exercise 6.1 Struct/linked List Exercises

In this exercise you must design a general *linked list* as an Abstract Data Type (ADT) (“Class” in Java).

Design (on paper) the structure of your linked list. Describe how an element is added and removed. Describe for all cases: empty list, with one element, with more elements etc. – This is to understand the problem!!

List Examples

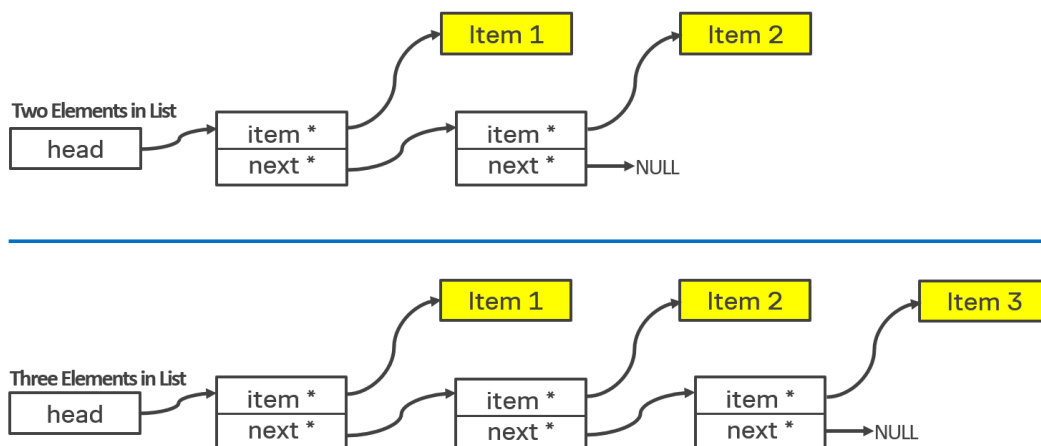
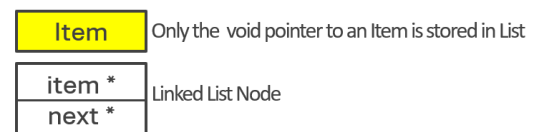


Figure 1 Example of different Linked List states

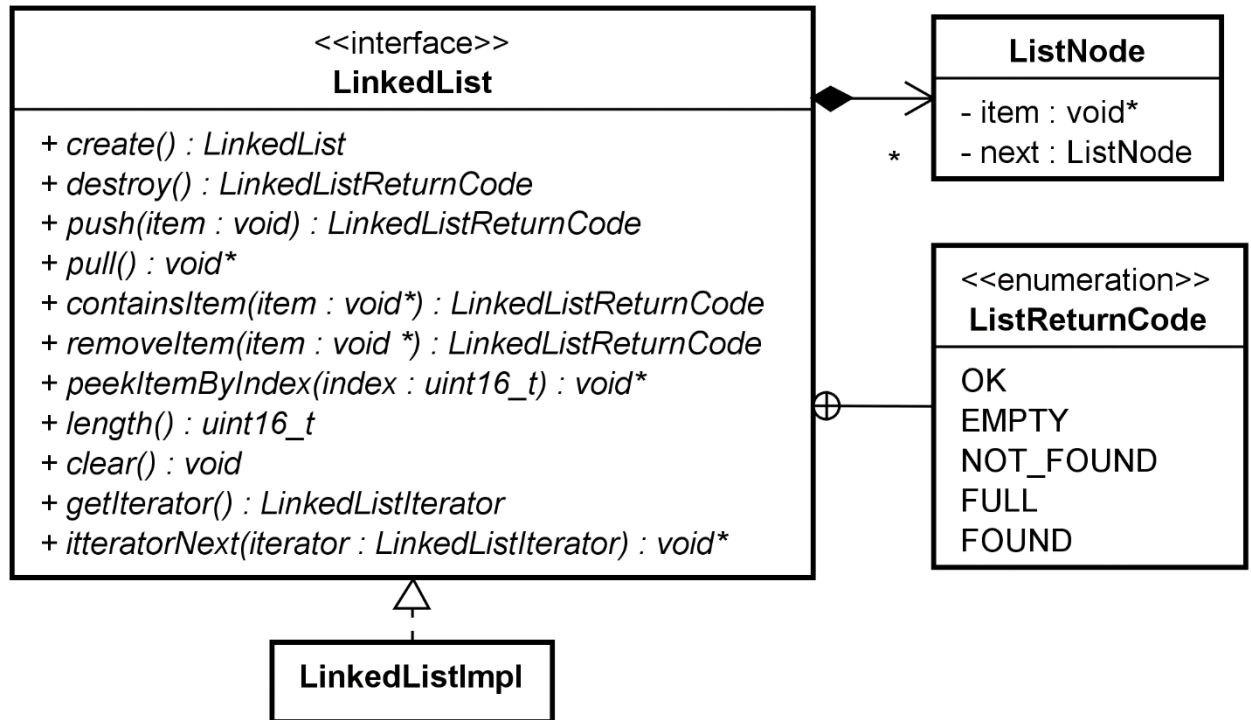
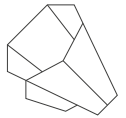


Figure 2 Preliminary Design for Exercise 6.1

Implement a linked list in two files `LinkedList.h` and `LinkedList.c`.

Use Test Driven Development (TDD).

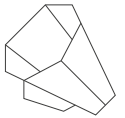
The items to be stored in the list is given as a void pointer (`void *`) that points to the element. In this way, you can implement a generic linked list that will be able to hold any kind of elements.

The linked list must at least have the contents shown in the class diagram above – **except** for the `getIterator` and `iteratorNext` functions.

The documentation for the wanted linked list can be found here:

<https://github.com/ihavn/ESW1-LinkedList>

Challenge: If you have time for it and want to challenge yourself you can implement the two last functions `getIterator` and `iteratorNext`.



Exercise 6.2 Use of your linked list

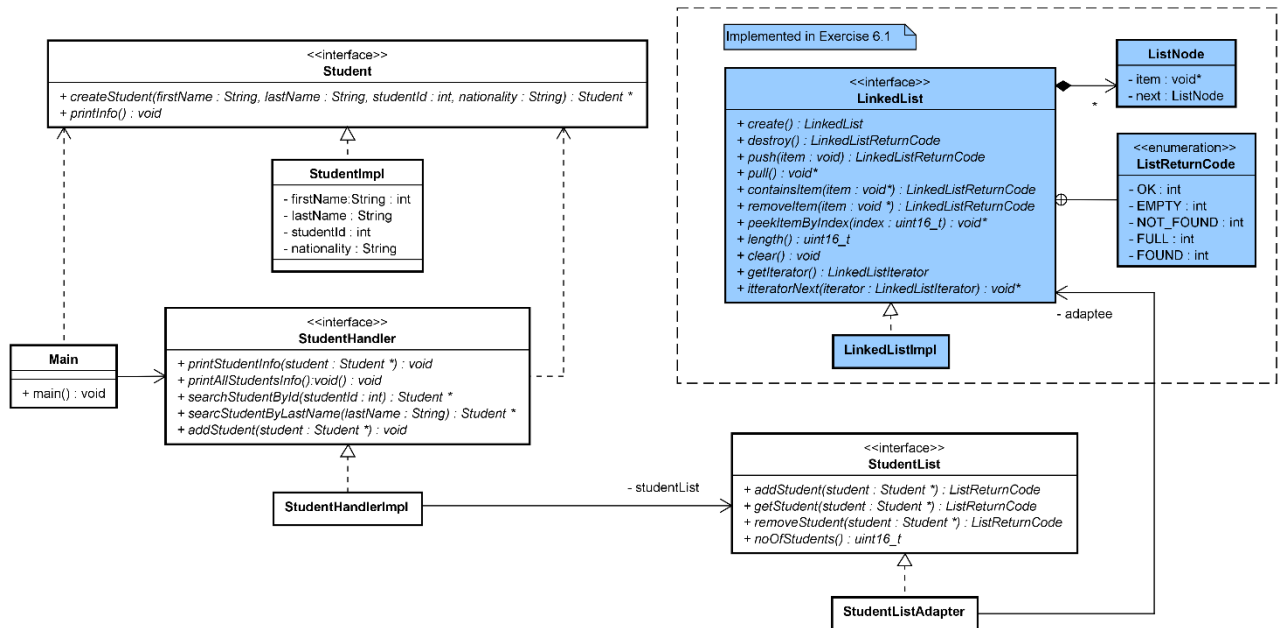


Figure 3 Preliminary Design for Exercise 6.2

To try out your linked list you must do the following:

1. Implement an abstract data type (ADT) in a program that holds information about a student. Try to use typedefs to make your code more easily readable. The information about a student is:
 - First Name
 - Last Name
 - VIA Student ID
 - Nationality
2. Use your ADT linked list to hold the students (these will be the items in your linked list) As it can be seen in Figure 3, I suggest the use of the *Adapter Design Pattern* to make a *StudentListAdapter* for the *LinkedList* (Adaptee). The adapter will then take care of all the casting to and from *student* to *void**.

The following is included in Figure 3 Preliminary Design for Exercise 6.2:

3. Make a function that can print the information about one student. The function must take a pointer to a student struct as argument
4. Build a main program that creates three students add them to the linked list
5. Get the students one by one from the linked list and print them out
6. **[Optional]** Implement a function that searches a linked list for a given VIA Student ID.
7. **[Optional]** Implement a function that searches a linked list for a given student by last name.