



L-Università ta' Malta
Faculty of Information &
Communication Technology

Assignment Report

Manwel Bugeja

December 27, 2019

Contents

0.1	Task 1	2
0.1.1	The Design	2
0.1.2	Testing	3
0.2	Task 2	4
0.3	Task 3	4
0.4	Appendix	4

0.1 Task 1

0.1.1 The Design

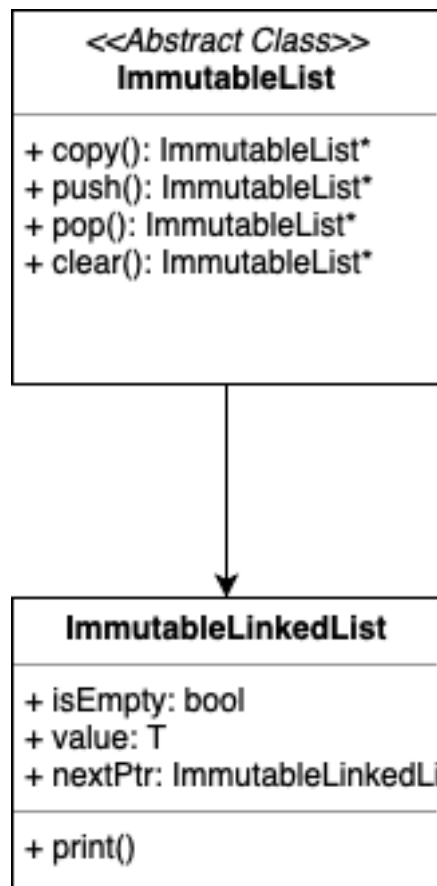


Figure 1: Task 1 UML Diagram

In this section a pure abstract class was created called 'ImmutableList' containing the required functions. Another class was created called 'ImmutableLinkedList' which inherited from the first class. This was done to implement the methods of the abstract class for a structure called a linked list. The class has 2 constructors and 3 variables.

The constructors

The key to this design is the fact that there are actually two constructors but one of them is inaccessible by the user. The hidden constructor has the ability to create nodes with custom values for the variables isHead,

value and nextPtr whose use is explained later. On the other hand the public constructor is only able to set the value of a head node. The public constructor is implemented using the private, passing isHead as true and nextPtr as nullptr and the value containing what is passed by the user.

isHead

This variable is a boolean type which indicates if the current node is the head of a list or not. This is needed to distinguish the head from other nodes of the list. isHead enables the possibility to have an empty list as the value head node is completely ignored by the api.

value

This variable contains the value of the node which is the same type as that indicated at the creation of the immutable list.

nextPtr

This variable contains a pointer towards the next node of the immutable list.

0.1.2 Testing

Initial testing

To test the immutability of a list, a list was created with some initial values. Then a new list was created using the old list and the push method. A third list was created using the first one and the push method with a different value passed as a parameter. All three lists were printed as should.

Concurrent testing

The list was also tested concurrently. A function was created that receives a list and a value to append. Two threads were created and the function was run in each thread with a different value passed. After the threads were joined, the list that was passed was printed to show that it was unchanged.

0.2 Task 2



Figure 2: Task 2 UML Diagram

0.3 Task 3

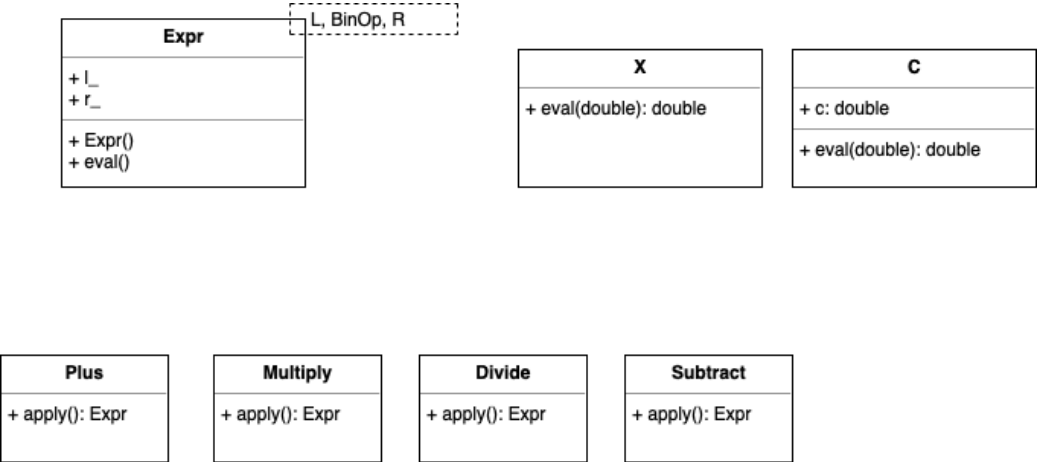


Figure 3: Task 3 UML Diagram

0.4 Appendix