Practical: The Linked List

For this exercise it may help to refer to a diagram of the list(s) to visualise where the nodes are located and how they might change.

1. Copy the files *LinkedListProgram.cs* to your working directory. The files contain slightly more involved details of the TownNode and TownList classes discussed in the lecture.

2. Edit in the content of Main() so that it>

    a) creates an object of a class TownList
    b) calls the list method listAllNames()
    c) adds four towns to the list using method addTownAtEnd(*string*)
    d) calls the list method listAllNames() again

    At each step ensure the program compiles and runs correctly before any further changes. Note the order in which the towns are displayed.

3. The TownList methods addTownAtEnd(*string*) adds the first item to the list and subsequent items to the end of the list. Add a new method called addTownAtStart(*string*) which will add a new element to the *start* of the list regardless of whether it is empty or not. To test, create a second list and then repeatedly call the new method with the same names of towns in the same order as they were added to the first list. If successful on running the program the method listAllNames() should display the order of the towns in reverse order from your original version of Main(), demonstrating that the second list has been built by adding to the start rather than the end of the list as was the case for the first list.

4. Search the web for an animation of a linked list operation.

DEMO TO TUTOR PROGRAM VERSION THAT USES METHOD addTownAtStart()