PCS4 Assignments – Week 5

Version: November 2014

Table of Contents

NOTE: The assignments marked with an asterisk are the most important ones.
The others serve as extra practice.

WEEK 5

Assignment 5.1: Linked List of Persons

In this assignment you must write code to implement a linked list of nodes where each node contains the name of a person (a string) together with a reference to the next name. At the start of the program the list contains only one single node, with your name.
Add a method to the form class to display the content of the list. This method must be called every time the list has been changed.

Now we want to allow the user to change the content of the list in several ways. So, for each of the actions mentioned below, you must add a button to the form and implement it:

- One button allows the user to add a new node to the linked list (use a textbox to enter the name for that new node). The list must be sorted at any moment, so the node must be inserted at the right place.

- One button allows the user to delete a name (use that same textbox to enter the name that must be deleted)

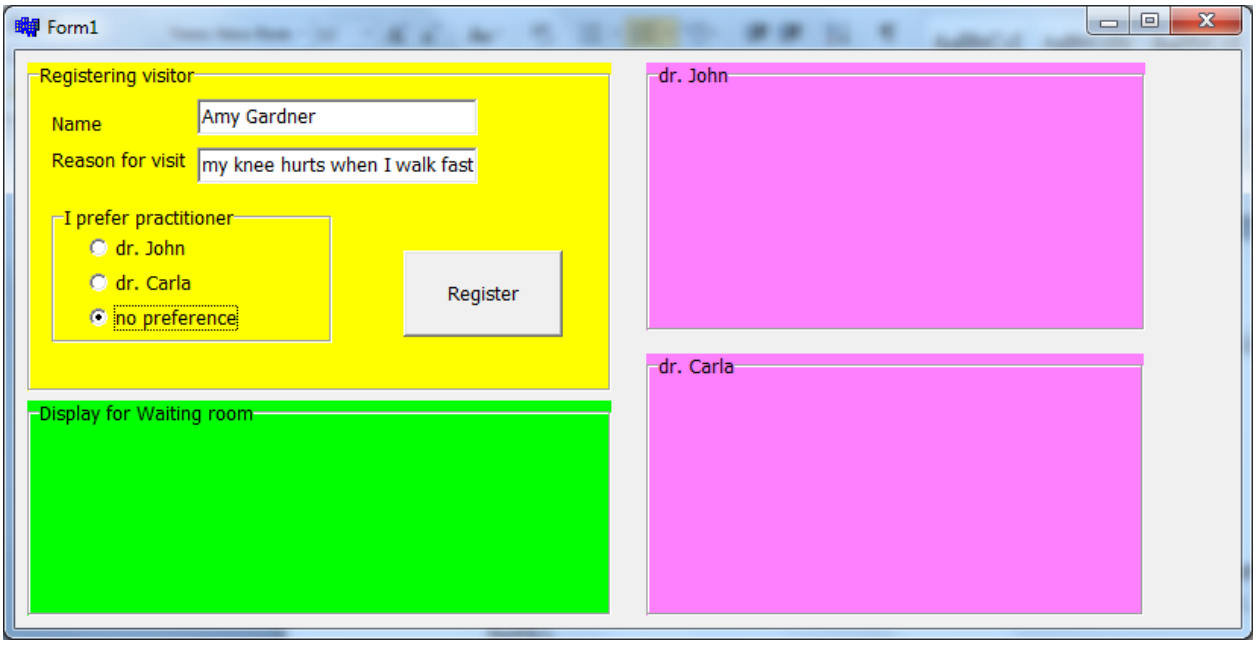
- One button allows the user to determine the number of names contained in the list (show the result on the form).

- One button allows the user to find the name contained in the n-th node, for a given number n. (The list starts with node number 0.) Use textboxes for the input (the value of n) and for the output (the name found).

* Assignment 5.2: A FIFO waiting list

In a health center there are two practitioners: dr. John and dr. Carla.
People, who visit the health center have to register themselves. Registering means they fill in a form and after clicking the Register button, their information is stored in a waiting list. After registering they go to the waiting room, where they should wait for their turn. The waiting list is a fifo-list (fifo stands for: first in, first out).
If one of the practitioners wants to serve the next visitor, he/she can click the button "next patient". He/she will then get information about his/her next patient in the list, the patient will be invited in the practitioner-room and, of course, removed from the waiting list.
In the waiting room all the visitors see information about the first 5 patients in the waiting list.

In fact we need 4 windows: one window for the visitor to register, one for the information in the waiting room, one for dr. John and one for dr. Carla.
For now, we will simulate this situation on one window with 4 regions, like:



The yellow region (a groupbox) is for registering a visitor; the green region is for the waiting room; the regions at the right of the window are respectively for dr. John and dr. Carla (the purple groupboxes).

1. Create a new project and build a user interface, which looks like the above figure.

2. A visitor has a name and a reason for the visit (for instance: "my knee hurts when I walk fast"). Furthermore a visitor can choose if he/she prefers a special practitioner (dr. John or dr. Carla) or not. Add a class to register a visitor.
3. Add a Node-class and a "ToDoList"-class to the project (very easy; just add existing items and change them a bit). A better name for the "ToDoList "-class would be something like "WaitingList". Give this class an appropriate name.

4. Implement the Register-button to add a visitor to the linked list.

5. In the green part (for the waiting room) we want to display the names of the first 5 visitors in the waiting list (if there are at least 5 persons in the waiting list, else . . .). Implement a method for this job.
   For testing purposes: add a test-button to test this method.

6. Now the difficult part: In the purple parts you should place a button for a practitioner asking for the next patient. The next visitor (if there is any) who prefers this practitioner (or both) should be removed from the waiting list. Somewhere in the purple groupbox for this practitioner, the visitor's name and reason for visit is displayed.
   On the display in the waiting room (the green part) should appear a message like "<name visitor>: please go to dr. Carla". And, of course, again the first 5 in the waiting list should be displayed.
   If there is no visitor in the waiting list for this practitioner, you should show a message to this practitioner.

7. Until so far assignment 3. If you have a solution for the former steps, you pass this assignment. For those who cannot stop working on it, here are some ideas for extra functionality.
   7.1. The practitioners have the possibility to see how many visitors are in the waiting list.
   7.2. The practitioners have the possibility to see how many visitors in the waiting list have him/her as the only preferred practitioner; how many prefer the other practitioner and how many prefer both.

Assignment 5.3: Extension of assignment 5.1

Can you change the linked list in assignment 5.1 into a double-linked list?
(In a double-linked list each node has both a forward reference to the next node in the list and a backward reference to the previous node.)