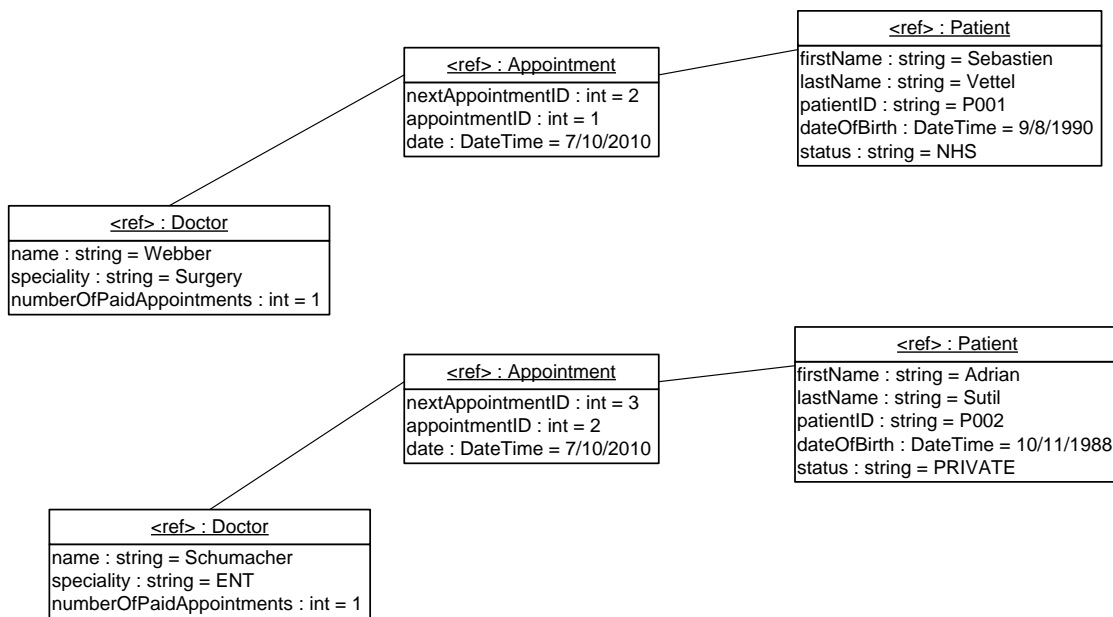


## LAB 3: Recording Appointments (continued) – answers to questions

### Questions on task 1:

- Draw an object diagram which shows the final state and associations of the objects involved in steps 2 to 5 above.



**the object references are declared in the GUI code - don't worry about the reference names for now**

- What can you say about the association between Appointment and Doctor? Can you say the same about Appointment and Patient? Why?

Can be changed – mutable

Appointment/Patient – could be mutable as the relevant property in Appointment is read/write – no way to change it in this application, though

### Questions on task 3:

- Question: You could write Note as a class rather than a struct, and you would notice no significant difference when you run the program. There would be a difference, though - what would be different about what happens in the computer's memory when a method is called with a parameter of type Note? When might the difference between using structs and classes have a real effect on a program?

Struct as method parameter or local variable is stored entirely on the stack, whereas for classes only an object reference is stored on the stack. Passing many large structs as parameters might fill up stack memory and cause "out-of-memory" error