

Un consultorio médico

Implementar un TAD *Consultorio* que simule el comportamiento de un consultorio médico simplificado. La especificación hará uso de los tipos *Medico* y *Paciente* que se representan con un *string*, y *Fecha* que es una clase con tres datos: día, hora y minuto. Las operaciones del consultorio son:

nuevoMedico(m): da de alta un nuevo médico en el consultorio. Si el médico ya estaba en el consultorio éste no se modifica.

pideConsulta(p,m,f): un paciente (*p*) pide consulta con un médico (*m*) para una fecha (*f*). Si el médico no está dado de alta se lanzará una excepción con el mensaje *Medico no existente*. Si la fecha ya está reservada, se lanzará una excepción con el mensaje *Fecha ocupada*. Un paciente puede tener varias citas con el mismo médico, siempre que sean en distinto momento.

siguientePaciente(m): consulta el paciente al que le toca el turno de ser atendido por un médico (*m*). Suponemos que el siguiente paciente es el que tiene una fecha menor. Si el médico no está dado de alta se lanzará una excepción con el mensaje *Medico no existente*. Si el médico no tiene pacientes asignados se lanzará una excepción con mensaje *No hay pacientes*.

atiendeConsulta(m): elimina el siguiente paciente de un médico (*m*). Suponemos que el siguiente paciente es el que tiene una fecha menor. Si el médico no está dado de alta se lanzará una excepción con el mensaje *Medico no existente*. Si el médico no tiene pacientes asignados se lanzará una excepción con mensaje *No hay pacientes*.

listaPacientes(m,f), devuelve la lista de pacientes de un cierto médico (*m*) que tienen cita el día (*f*). Se supone que el día es un número entero correcto. Si el médico no está dado de alta se lanzará una excepción con mensaje *Médico no existente*. Si el médico no tiene pacientes ese día, la lista de retorno será vacía.

Requisitos de implementación.

Implementar un tipo de datos *Fecha* con tres campos *día*, *hora* y *minuto*, con métodos para acceder a los datos, para modificar los datos, sobrecargar el operador de comparación menor y si se quiere los operadores de entrada y salida de datos. No es necesario realizar más métodos para resolver el problema.

Los tipos *medico* y *paciente* pueden ser simplemente un *string*.

Los métodos del TAD *Consultorio* no deben mostrar nada por pantalla. El manejo de la entrada y salida de datos se realizará en funciones externas al TAD.

Entrada

La entrada consta de una serie de casos de prueba. Cada caso comienza con una línea en la que se indica el número de operaciones que se llevará a cabo en el consultorio. En las líneas siguientes se muestran las operaciones, una por cada línea. Las operaciones posibles son:

nuevoMedico, seguido del nombre del médico.

pideConsulta, seguido del nombre de un médico; del nombre de un paciente; y de una fecha expresada con tres números enteros: el primero indica el día de la cita, el segundo la hora y el tercero el minuto.

siguientePaciente, seguido del nombre de un médico

atiendeConsulta, seguido del nombre de un médico

listaPacientes, seguido del nombre de un médico y de un valor entero correspondiente al día del que se quiere sacar el listado.

Los nombres de los médicos y pacientes no tienen blancos y todas las fechas pueden considerarse correctas.

Salida

Para cada caso de prueba se escribirán los datos que se piden. Cada caso termina con una línea con seis guiones. Las operaciones que generan datos de salida son: *siguientePaciente* que escribe dos líneas; en la primera debe poner *Siguiente paciente doctor* seguido del nombre del doctor que atiende y en la segunda el nombre del paciente. La operación *listaPacientes* escribe como cabecera *Doctor* seguido del nombre del doctor, seguido de *día*, seguido del día del que se pide el listado. En las líneas siguientes se escribe el nombre seguido de la hora y del minuto de la cita de todos los pacientes de ese día con ese doctor, ordenados por hora de consulta. La hora se expresa con dos puntos entre el valor de la hora y el de los minutos. Cada operación termina con una línea con tres guiones.

Si alguna operación produce una excepción se mostrará el mensaje de la excepción como resultado de la operación, seguido de una línea con tres guiones para cerrar la operación.

Entrada de ejemplo

```
12
nuevoMedico Hernandez
nuevoMedico Alvarez
pideConsulta Ana Alvarez 16 12 30
pideConsulta Antonio Alvarez 16 12 10
pideConsulta Alvaro Alvarez 17 10 00
pideConsulta Alba Alvarez 17 10 15
pideConsulta Anacleto Alvarez 17 10 55
listaPacientes Alvarez 16
siguientePaciente Alvarez
atiendeConsulta Alvarez
listaPacientes Alvarez 16
listaPacientes Alvarez 17
6
nuevoMedico Hernandez
pideConsulta Alvarez Ana 16 12 30
pideConsulta Helena Hernandez 17 10 00
atiendeConsulta Hernandez
listaPacientes Hernandez 17
atiendeConsulta Hernandez
```

Salida de ejemplo

```
Doctor Alvarez dia 16
Antonio 12:10
Ana 12:30
---
Siguiente Paciente doctor Alvarez
Antonio
---
Doctor Alvarez dia 16
Ana 12:30
---
Doctor Alvarez dia 17
Alvaro 10:00
Alba 10:15
Anacleto 10:55
---
-----
Medico no existente
---
Doctor Hernandez dia 17

---
No hay pacientes
---
-----
```

Autor: Isabel Pita.