

## Examen de Programación Junio 2020 (Grados en Física y Matemáticas)

Se dispone de un fichero con datos de un estudio clínico de los efectos de un medicamento sobre pacientes covid-19 y se desea hacer parte del software que sirva para su análisis.

El módulo paciente.py dispone de las siguientes constantes enteras definidas, agrupadas en tres grupos en los que los valores numéricos son consecutivos y están definidos en el mismo orden indicado aquí:

- Condiciones de salud o enfermedad previa del paciente: SANO, DIABETES, HIPERTENSION, CARDIOPATIA, PATOLOGIA\_RENAL
- Sexo del paciente: VARON, MUJER
- Estado más grave alcanzado: SIN\_DATOS, LEVE, HOSPITALIZADO, UCI, MUERTE

En el módulo paciente.py se dispone también de la clase Paciente, *ya realizada*, que guarda los datos de un paciente concreto y responde al diagrama de clases que se muestra.

Paciente	EstudioClinico
-nombre: str -edad: int -condicion: int -sexo: int -dosis: float -estado: int	-lista: List[Paciente] -fecha_inicio: str -fecha_fin: str -autor: str
+ __init__(nombre: str, edad: int, condicion: int, sexo: int, dosis: float) {raises ValueError} + establece_nuevo_estado(estado: int) {raises ValueError} + get_nombre(): str + get_edad(): int + get_condicion(): int + get_sexo(): int + get_dosis(): float + get_estado(): int	+ __init__(fecha_inicio: str, fecha_fin: str, autor: str) + lee(nombre_fichero: str) {raises FicheroNoExiste, ValueError} + listado(sexo: str) {raises ValueError} + primer_paciente_uci(sexo: int, condicion: int): Optional[str] {raises ValueError} + num_pacientes_hospitalizados(dosis: float): int {raises ValueError}

La clase dispone de un constructor al que se pasan los valores de los atributos excepto el estado, un método para establecer el estado y métodos observadores, uno para cada atributo. Los atributos de la clase se indican a continuación:

- nombre: el nombre del paciente
- edad: la edad del paciente
- condicion: la condición de salud o enfermedad previa del paciente. Debe ser una de las constantes SANO, DIABETES, HIPERTENSION, CARDIOPATIA o PATOLOGIA\_RENAL
- sexo: el sexo del paciente. Debe ser una de las constantes VARON o MUJER
- dosis: la dosis de medicamento recibida, en mg por día, que puede ser 0.0, 100.0 o 200.0

- estado: el estado más grave alcanzado por el paciente. Debe ser una de las constantes SIN\_DATOS, LEVE, HOSPITALIZADO, UCI o MUERTE

En el módulo estudio\_clinico.py cuyo esqueleto se da ya hecho se pide hacer la clase EstudioClinico que responde al diagrama de clases que se muestra arriba. La clase dispondrá de estos atributos:

- lista: lista de objetos de la clase Paciente, cada uno de ellos con los datos de un paciente sometido al estudio
- fecha\_inicio: la fecha de inicio del estudio
- fecha\_fin: la fecha de finalización del estudio
- autor: el nombre del responsable del estudio

Y dispondrá de diversos métodos para analizar los resultados del estudio:

- *constructor*: crea la lista vacía y da valor al resto de los atributos a partir de los parámetros
- *lee()*: Este método se da ya hecho. Lee los datos del estudio del fichero CSV cuyo nombre se indica, y los mete en la lista. Puede lanzar la excepción FicheroNoExiste, ya definida en el mismo módulo, si el fichero no existe, o la excepción predefinida ValueError si el fichero contiene errores.
- *listado()*: Hace un listado en pantalla de los datos de los pacientes, filtrándolos en su caso por sexo. Si el parámetro sexo vale "\*" se muestran ambos sexos; si vale "VARON" solo los varones o si vale "MUJER" se muestran solo las mujeres. Si no vale ninguno de estos valores se lanzará la excepción predefinida ValueError.

El método pone primero un par de líneas de encabezamiento y luego una línea por paciente, con sus datos en columnas. La primera línea del encabezamiento mostrará la fecha de inicio, fecha de fin y nombre del responsable del estudio. La segunda línea del encabezamiento explica los datos de los pacientes que vendrán a continuación. Ejemplo de las primeras líneas de un listado:

```

Estudio clínico desde 1-mar-2020 hasta 31-mar-2020. Por: Doctor B. Pérez
Nombre          Edad          Condicion Sexo  Dosis mg/d          Estado
Paciente 000    94           DIABETES VARON    0.0 HOSPITALIZADO
Paciente 001    79           DIABETES VARON    100.0 LEVE
  
```

- *primer\_paciente\_uci()*: Retorna el nombre del primer paciente de estado UCI cuyo sexo y condicion coinciden con lo indicado en los parámetros. El parámetro sexo debe ser uno de los valores VARON o MUJER definidos en el módulo paciente<sup>1</sup>. El parámetro condicion debe ser uno de los valores enteros SANO, DIABETES, HIPERTENSION, CARDIOPATIA, o PATOLOGIA\_RENAL definidos en el módulo paciente. El método retorna el nombre del primer paciente que cumple los criterios o None si no hay ninguno. Lanza la excepción predefinida ValueError si el sexo o la condicion tienen valores incorrectos.
- *num\_pacientes\_hospitalizados()*: Retorna el número de pacientes cuyo estado es HOSPITALIZADO y que han recibido la dosis indicada por el parámetro dosis, que debe ser 0.0, 100.0 o 200.0 mg/día. El método retorna el número de pacientes que cumplen los criterios señalados. El método lanza la excepción predefinida ValueError si el parámetro dosis no es correcto.

---

1. Recordamos que una forma de usar una constante definida en otro módulo es importar ese módulo y luego usar la notación nombre\_módulo.nombre\_constante

Finalmente, se pide hacer un programa principal, en el mismo módulo estudio\_clinico.py, que haga lo siguiente en base a invocar a los métodos de la clase EstudioClinico:

- a. Crear un objeto de la clase EstudioClinico
- b. Leer los datos del fichero estudio\_h11.csv que se suministra
- c. Hacer un listado completo, para ambos sexos
- d. Mostrar en pantalla el nombre del primer paciente UCI que sea MUJER y con PATOLOGIA\_RENAL (debe salir "Paciente 561")
- e. Mostrar en pantalla el número de pacientes hospitalizados que han recibido una dosis de 200 mg/día (debe salir 73)

Adicionalmente a lo que se pide en los pasos anteriores, se debe programar el siguiente tratamiento de errores:

- Si en el paso b) se lanzase FicheroNoExiste se abandonan los restantes pasos y se escribe el mensaje "El programa no se puede ejecutar"
- Si en los pasos c) o d) se lanzase ValueError se pone un mensaje de error y luego se continúa normalmente con el paso e)
- Si en el paso e) se lanzase ValueError se pone un mensaje de error

*Valoración:* Se valorará el correcto funcionamiento, el estilo de programación que haga el programa fácil de entender, y la eficiencia y tamaño compacto de la solución. Por ello, si no hay tiempo de hacer todo el examen es preferible centrarse en ir dejando cosas cerradas y en funcionamiento que en dejar todo a medias.

La estrategia recomendada sería hacer el constructor, un método y probarlos desde el main(). Luego hacer otro método y añadir instrucciones al main() para probarlo. Y así sucesivamente. Si te atascas con un método no pierdas mucho tiempo y sigue con el siguiente. El método listado() es más largo que los demás, por lo que se recomienda dejarlo para el final.

- constructor: 1 punto
- método listado(): 3 puntos
- método primer\_paciente\_uci(): 2 puntos
- método num\_pacientes\_hospitalizados(): 2 puntos
- programa principal: 1 punto por los pasos a) hasta e) y 1 punto por el tratamiento de errores

*Entrega:* El examen comienza a las 9:00 el viernes 19 de junio y consta de dos entregas. No se admiten entregas retrasadas:

- Entrega 1: entre las 10:20 y las 10:30
- Entrega 2: antes de las 11:00

En ambas entregas hay que entregar el módulo estudio\_clinico.py

Se recomienda leer las instrucciones para el examen final en:  
<https://moodle.unican.es/mod/forum/discuss.php?d=46216>