Proyecto de laboratorio - Java

Estructura general del proyecto

a) El proyecto debe contener tres carpetas: src, data y doc.
b) La carpeta src debe incluir el código fuente, organizado en paquetes, la carpeta data el fichero de trabajo que más adelante se le proporcionará y, por último, la carpeta doc donde se guardarán la documentación del proyecto.
c) El proyecto debe contener un fichero README.md donde se describirán los datos y los tipos implementados.

Entrega 1

1) Diseño moderno[1]

Cree un paquete de nombre fp.clinico y programe los siguientes tipos como records.

Persona

- Propiedades:
  - nombre, de tipo String.
  - apellidos, de tipo String.
  - dni, de tipo String.
  - fecha de nacimiento, de tipo LocalDate.
  - edad, de tipo Integer. (Derivada a partir de la fecha de nacimiento).
- Restricciones:
  - La fecha de nacimiento debe ser anterior a la fecha actual.
  - El dni debe ser una cadena con ocho dígitos y seguidos de una letra.
- Representación como cadena: por defecto asociado al record.
- Criterio de igualdad: por defecto asociado al record.
- Orden natural: por dni.
- Comentarios: añada los siguientes métodos de factoría:
  - Método static of:
    - recibe nombre, apellidos, dni y fecha de nacimiento y devuelve una persona.
  - Método static parse:
    - Recibe una cadena con un formato específico y devuelve una persona. Ejemplo de cadena: “Juan, García Rodríguez, 12755078Z”.
  - Incluya un método main para comprobar el correcto funcionamiento del método parse. *public static void main(String[] args){ … }*

[1] Siguiendo las pautas de diseño vistas en clase de teoría con Miguel Toro.

Paciente

- Propiedades:
  - persona, de tipo Persona.
  - código de ingreso, de tipo String.
  - fecha y hora de ingreso, de tipo LocalDateTime.
  - fecha de ingreso, de tipo LocalDate. (Derivada a partir de la fecha y hora de ingreso)
  - hora de ingreso, de tipo String. (Derivada a partir de la fecha y hora de ingreso). Ejemplo de cadena: "15:03". Es decir, la hora y los minutos tienen que tener dos dígitos. No valdría escribir "15:3".
- Restricciones:
  - La fecha y hora de ingreso debe ser anterior o igual a la fecha actual.
- Representación como cadena: por defecto asociado al record.
- Criterio de igualdad: por defecto asociado al record.
- Comentarios: añada los siguientes métodos de factoría:
  - Método static of:
    - recibe nombre, apellidos, dni, fecha de nacimiento, código y fecha y hora de ingreso y devuelve un paciente.
  - Método static of:
    - recibe un objeto persona, un código y una fecha y hora de ingreso y devuelve un paciente.

PacienteEstudio

- Propiedades:
  - id, de tipo String.
  - genero, de tipo String.
  - edad, de tipo Double.
  - hipertensión, de tipo Boolean.
  - enfermedad del corazón, de tipo Boolean.
  - tipo de residencia, enumerado TipoResidencia, cuyos valores son rural o urbana.
  - nivel medio de glucosa, de tipo Double.
  - factor de riesgo, de tipo Boolean. (Derivada, si tiene hipertensión y más de 40 años se considerará que tiene factor de riesgo).
- Restricciones:
  - La edad tiene que ser mayor o igual que cero y menor o igual que 130.
  - El nivel medio de glucosa tiene que ser mayor o igual que cero.
- Representación como cadena: informa del id y la edad del paciente.
- Criterio de igualdad: por defecto asociado al record.
- Criterio de orden: según la edad y el id.
- Comentarios: añada los siguientes métodos de factoría:
  - Método static of:
    - recibe valores para cada propiedad básica y devuelve un objeto del tipo.
  - Método static parse:
    - recibe una cadena con un formato especificado y y devuelve un objeto del tipo. Ejemplo de cadena: "6306;Male;80;false;false;URBANA;83.84"

- Incluya un método main para comprobar el correcto funcionamiento del método parse. *public static void main(String[] args){ … }*

Cree un paquete de nombre fp.vacunas y programe el siguiente tipo como un record.

Vacunacion

- Propiedades:
  - fecha, de tipo LocalDate.
  - comunidad, de tipo String.
  - pfizer, de tipo Integer.
  - moderna, de tipo Integer.
  - astrazeneca, de tipo Integer.
  - janssen de tipo Integer.
  - número de personas, de tipo Integer.
  - número total, de tipo Integer. (Derivada, siendo la suma de dosis de Pfizer, moderna, astrazeneca y janssen).
- Restricciones:
  - La fecha de debe ser posterior al 01/02/2021.
- Representación como cadena: por defecto asociado al record.
- Criterio de igualdad: por defecto asociado al record.
- Orden natural: por comunidad y en caso de igualdad por fecha.
- Comentarios: añada los siguientes métodos de factoría:
  - Método static of:
    - recibe valores para cada propiedad básica y devuelve un objeto del tipo.
  - Método static parse:
    - recibe una cadena con un formato específico y devuelve un objeto del tipo. Ejemplo de cadena: “04/01/2021;Andalucía;140295;0;0;0;0”.
  - Incluya un método main para comprobar el correcto funcionamiento del método parse. *public static void main(String[] args){ … }*

2) Diseño clásico[2]

Cree un paquete de nombre fp.farmaceutico y programe el siguiente tipo como una clase.

Medicamento

- Propiedades:
  - nombre del medicamento, de tipo String, observable.
  - tipo de medicamento, enumerado de tipo TipoMedicamento, observable. Los valores del enumerado son anatómico, químico y terapéutico.
  - código de la enfermedad, de tipo String, observable.

[2] Los records fueron introducidos en la versión 14 de Java – el JDK14 – que salió a la luz el 17 de marzo de 2020. Llamaremos *diseño clásico* al diseño que se solía hacer hasta ese momento. En general, no se insistía en el uso de tipos inmutables para los tipos simples – que teniendo records se programan con mucha comodidad – ni, por otro lado, se utilizaban una serie de pautas o patrones software a la hora de programar los tipos.

  - farmacéutica, de tipo String, observable.
  - puntación, de tipo Double, observable.
  - índice somático, de tipo Integer, observable.
  - fecha de catálogo, de tipo LocalDate, **observable y modificable**.
  - tratar enfermedad, de tipo Boolean. (Derivada, siendo cierta si el código de la enfermedad coincide con un parámetro de tipo cadena que reciben como argumento la propiedad).
- Restricciones:
  - La puntación tiene que ser mayor estricta que cero.
  - El índice somático tiene que ser mayor o igual que 1000.
  - La fecha de catálogo tiene que ser posterior al 01/01/2015.
- Representación como cadena: según el nombre del medicamento y de la farmacéutica.
- Criterio de igualdad: por nombre del medicamento y farmacéutica.
- Orden natural: por nombre del medicamento y en caso de igualdad por la farmacéutica.
- Comentarios:
  - Programe una **clase de nombre FactoriaMedicamentos** que incluya, de momento, un método static de nombre **parseaMedicamente**, que recibe una cadena con un formato específico y devuelve un objeto de tipo Medicamento.
    Ejemplo:
    "efavirenz,Anatomico,Y212XXA,Actavis Mid Atlantic LLC,90.0,1848,04/12/2019". (Siendo la información del nombre del medicamento, el tipo de medicamento, el codigo de la enfermedad, la farmaceutica, la puntuacion, el índice somatico, y la fecha de catalogo)
  - Implemente una clase de nombre TestFactoriaMedicamentos en un paquete de nombre fp.farmaceutico.test y compruebe el correcto funcionamiento del método anterior.