PRÁCTICA

Enunciado general

Con el objeto de aportar nuestro granito de arena en la lucha contra la pandemia, la Agencia Europea del Medicamento (EMA) nos ha encargado el desrrollo de un software para la gestión de la vacunas contra el COVID-19.

Para ello, crearemos tres clases y una interface en un único paquete llamado gal.teis.vacunas:

- Interface **IAutorizable**
- Clase abstracta **VacunaAutorizacion**.
- Clase (*POJO*) **Vacuna**
- Clase **VacAlmacen**
- Clase **Aplicación** (método main())

Interfaz IAutorizable

La interfaz **IAutorizable** se usará para autorizar o no los objetos creados a partir de **Vacuna** y tendrá dos métodos:

- **boolean autorizar()** que hace que una vacuna esté autorizada para su uso.
- **boolean rechazar()** que hace que una vacuna no esté autorizada para su uso.

El valor devuelto de los dos métodos indica si la operación se ha realizado o no.

Clase abstracta VacunaAutorizacion

Esta clase debe implementar la interface **IAutorizable**.

Tendrá los siguientes atributos de tipo **boolean** que indican si ha superado o no cada una de las fases necesarias para su aprobación.

- **fase1Superada**
- **fase2Superada**
- **fase3Superada**

Para saber si la información de las distintas fases ya es la definitiva se deberá utilizar el atributo *byte* **fasesCompletadas**, que tendrá el valor correspondiente a las fases de las que se ha almacenado su resultado. Cuando el resultado de las tres fases hayan sido introducidas, este atributo tendrá el valor **3**.

Los métodos para modificar el valor de **fase1Superada**, **fase2Superada** y **fase3Superada** son los que modificarán el valor del atributo **fasesCompletadas**. De esta forma se interpretará correctamente el valor *false* por defecto que tienen los atributos *boolean*, que no significa que la fase no ha sido superada, sino que aun no se ha introducido el resultado de la fase.

La implementación de los métodos de la interfaz **IAutorizable** deberán analizar los atributos **fase…** y devolver *true* o *false* en función del método a implementar (autorizada(), rechazada()). Para que una vacuna sea autorizada deberá verificar que todas las fases de investigación se han completado con éxito, analizando el valor de

fasesCompletadas, **fase1Superada**, **fase2Superada** y **fase3Superada**, en cualquier otro caso, las vacunas no serán autorizadas.

La decisión de rechazar una vacuna es una decisión de la EMA, más allá de que una vacuna haya superado todas las fases (que podría ser).

La clase también tendrá los siguientes atriburos:

- *LocalDate* **fechaResultado** que recogerá la fecha en que la vacuna ha sido rechazada o autorizada.
- *boolean* **autorizada** y *boolean* **rechazada** que contendrá el resultado de la operación **autorizar()** y **rechazar()** respectivamente. Es importante que existan los dos atributos ya que si los dos valores son *false*, significa que aún no se an completado las fases de investigación.

Además de los métodos de la interfaz y los de modificación de **faseiSuperada** debemos tener métodos que:

- Devuelva el valor de **fechaResultado.**
- Devuelva el resultado de la investigación de la **útima** fase que se ha implementado con la vacuna.

Cuestiones a tener en cuenta:

- Deben existir métodos para ver si una vacuna ha sido autorizada o no, pero la autorización solo se debe producir con los métodos implementados de **IAutorizable** .
- El valor de **fechaResultado** solo se podrá modificar desde el método **autorizar()** y **rechazar()** y se corresponderá con la fecha actual del sistema en el momento de llevar a cabo la operación.
- El valor de **autorizada** solo podrá ser modificado desde **autorizar().**
- El valor de **rechazada** solo podrá ser modificado desde **rechazar ().**

Si una vacuna ha sido rechazada no puede ser posteriormente autorizada, de la misma forma, si ha sido rechazada no podrá ser autorizada.

Clase Vacuna

La clase **Vacuna** deberá almacenar: el **código**, el **nombre**, el **principio activo**, **farmacéutica**, el **precio recomendado**.

El comportamiento de la clase vendrá determinado por:

- Un **constructor** que inicializará los datos a los valores que se indiquen.
- Dos **métodos públicos**, uno para acceder al atributo del precio y otro para darle valor. Hacer lo mismo con el resto de los atributos, si procede.
- El **código** tendrá el siguiente formato:
  - Comenzará por la letra V seguida de una vocal en mayúsculas.
  - A continuación, tres o cuatro letras minúsculas.
  - Finaliza, o con dos números del 4 al 7, o bien con el número 8.

- Método **toString()** que mostrará los datos de una vacuna de la siguiente manera (siempre que esté autorizada por la EMA).

```
  Datos de la vacuna:
     Código          VAedf45
     Nombre          COVID-19 vacuna Astrazeneca
     P. activo       Adenovirus de chimpacé
     Farmaceutica    AstraZeneca
     Precio          2.9 €
```

  En el caso de que la vacuna no esté autorizada por la EMA mostrará

```
Datos de la vacuna:
  Código          ZSgtf23
  Nombre          COVID-19 vacuna Sputnik V
  P. activo       Adenovirus Ad26 y Ad5
```

- Método **equals()** e **hashCode()** que determinarán que dos vacunas son iguales si sus códigos los son también.

Clase VacAlmacen

La clase **VacAlmacen** deberá gestionar una colección de vacunas y tener los métodos necesarios para ejecutar las distintas operaciones del menú que se mustra en la clase **Aplicación**.

Clase Aplicación

Realizará las operaciones que se corresponden con el menú siguiente:

```
1.Listar todas las vacunas y mostrar todos sus datos
2.Buscar vacuna.
3.Agregar vacuna.
4.Eliminar vacuna.
5.Introducir resultado de las fases de la vacuna.
6.Autorizar/Rechazar vacuna.
7.Ver vacunas autorizadas.
8.Ver vacunas rechazadas.
9.Ver vacunas pendientes de autorizar/rechazar.
10.Ver la última fase investigada de cada vacuna almacenada.
```

El punto 5 debe dar la posibilidad de introducir las distintas fases de investigación. Esta operación se puede gestionar de distintas formas.