

ACTIVIDAD DE RECUPERACIÓN

MÓDULO DE PROGRAMACIÓN

La actividad de recuperación de POO consistirá en la revisión y corrección de la práctica de **Vacunas**. Esta actividad se realizará en el horario habitual de clases combinado con trabajo en casa.

Las tareas de recuperación podrán tener un ejercicio que se realice en clase y que puntúe para la nota final. La puntuación se indicará cada día y será proporcional a la dificultad que implique.

Las tareas que cada alumno/a tendrá que tener realizadas para cada día de recuperación se indican en la siguiente tabla:

Miércoles, 2 de junio	
Planificación del trabajo	
Viernes, 4 de junio	
1. Revisión en grupo del trabajo realizado. 2. Realización de una tarea sobre el trabajo realizado.	POO <ul style="list-style-type: none"> • Agregar vacuna. • Buscar vacuna. • Eliminar vacuna. • Modificar vacuna. • Listar todas las vacunas y mostrar sus datos. • Crear una opción para cargar una lista de 5 vacunas de forma automática. Swing/Ficheros <ul style="list-style-type: none"> • Guardar todas las vacunas en un fichero a partir de la lista donde se almacenan todas las vacunas. Separar la información de cada vacuna con un salto de línea. • Mostrar las vacunas en la interfaz gráfica. • Añadir una entrada de menú para ver el número de vacunas.
Lunes, 7 de junio	
1. Revisión en grupo del trabajo realizado. 2. Realización de una tarea sobre el trabajo realizado.	POO <ul style="list-style-type: none"> • Introducir el resultado de las fases de la vacuna. • Autorizar/Rechazar vacuna. • Ver vacunas autorizadas. • Ver vacunas rechazadas. • Ver vacunas pendientes de autorizar/rechazar. • Ver la última fase investigada de cada vacuna. Swing/Ficheros <ul style="list-style-type: none"> • Guardar las vacunas en un fichero, poniendo delante de cada vacuna su estado: <ul style="list-style-type: none"> ○ A (autorizada) ○ N (no autorizada) ○ P (pendiente de autorización)

	<ul style="list-style-type: none"> Añadir una entrada de menú para ver el número de vacunas en cada estado cuando el fichero de vacunas está abierto.
Martes, 8 de junio	
1. Revisión en grupo del trabajo realizado. 2. Realización de una tarea sobre el trabajo realizado.	<p>POO</p> <ul style="list-style-type: none"> Añadir una clase (<i>Pais</i>) que represente a los países a los que se le asigna la vacuna. Con código y nombre de país como atributos. Añadir una clase (<i>ConjuntoPaises</i>) que contenga una lista de países de tipo <i>hashset</i> y métodos para realizar cualquier operación con la lista (agregar, modificar, eliminar, buscar y listar). Crear una opción para almacenar 5 países en la lista de forma automática. <p>Swing/Ficheros</p> <ul style="list-style-type: none"> Crear una entrada en el menú llamada <i>Dividir</i> que, a partir del fichero de vacunas abierto, genere tres ficheros: uno con las vacunas autorizadas, otro con las vacunas no autorizadas y un tercero con las vacunas pendientes de autorización. Los ficheros generados deben tener un nombre significativo.
Miércoles, 9 de junio	
1. Revisión en grupo del trabajo realizado. 2. Realización de una tarea sobre el trabajo realizado.	<p>POO</p> <ul style="list-style-type: none"> Incorporar la parte de asignación de vacunas del examen. <p>Swing/Ficheros</p> <ul style="list-style-type: none"> Crear un fichero con la información de cada país, tipo de vacuna y nº de vacuna asignada.
Viernes, 10 de junio	
1. Revisión en grupo del trabajo realizado. 2. Realización de una tarea sobre el trabajo realizado.	
Miércoles, 16 de junio las 16:00 examen sobre la práctica	

DIRECTRICES PARA UNA BUENA PROGRAMACIÓN

Evitarán:

- Realizar entradas o salidas por teclado/pantalla en clases diferentes a la clase con *main()*.
- Utilizar `System.print()`

Hacer:

- El proyecto debe tener por lo menos debe tener los siguientes paquetes:
 - **aplicación**: con la clase Java *main*.
 - **vacunas**: con todas las clases/Interfaces. que tengan que ver con las vacunas.
 - **país**: con todas las clases/Interfaces. que tengan que ver con las vacunas.
 - **vacunas.asignacion**: con todas las clases que tienen que ver con la asignación de vacunas a los países.
 - **excepciones**: si hay excepciones propias.
- Todas las clases deben tener implementados los métodos *toString()*, *equals()* y *hashCode()*. Se pueden excluir de estas indicaciones las clase abstractas si así lo permite el programa.
- Controlar excepciones en las entradas de pantalla y trabajo con ficheros.
- Tener el código con formato/indentación adecuado.
- Utilizar la librería para el control de entrada de datos con *Scanner*.
- Mostrar mensajes en el caso de que la operación no se pueda realizar (listas vacías, elementos no encontrados, añadir elementos repetidos cuando no se puede, etc.).

LISTADO DE ERRORES ENCONTRADOS EN LA CORRECCIÓN DE LA PRÁCTICA DE VACUNAS QUE SE DEBEN EVITAR

Vacunas:

- Los tipos de vacuna no deberían estar predefinidos.
- Al introducir una vacuna repetida, no avisa.
- Al borrar una vacuna (seguimos sin saber si hay que poner el código o el nombre..) al poner el código dice que borra la vacuna pero no hace nada ni informa de que no encontró la vacuna. La operación borra una vacuna si le das el nombre, pero si pones cualquier otro dato o si no encuentra la vacuna, no informa del problema, y muestra un mensaje "Borrando..." pero no borra nada.
- Al borrar una vacuna (seguimos sin saber si hay que poner el código o el nombre..) al poner el código dice que borra la vacuna pero no hace nada ni informa de que no encontró la vacuna.
- Al introducir una vacuna repetida, no avisa. Después no la guarda, es decir, lo hace bien pero no da ninguna información al usuario.
- El código no había que generarlo automáticamente sino que se debe introducir por pantalla y controlar que se cumpla el patrón.
- Al buscar una vacuna si no existe no informa de nada.
- Ausencia de mensajes informativos en los listados si la lista está vacía (7, 8, 1..).
- No hay control pattern del código de la vacuna.
- Las vacunas rechazadas y autorizadas deberían tener un toString() diferente.
- En el método equals() no haces un control del código y nombre que es por lo que dos vacunas serán iguales o no.

Fases:

- Al pedir la fase para introducir el resultado no te informa previamente de la fase que está pendiente. Solo cuando introduces una fase que no corresponde te indica cuál le toca.
- Las fases se pueden superar o no. Solo permites avanzar fases para superarlas, no para lo contrario.
- Si llamo a avanzar fase y aún no hay vacunas o introduzco una vacuna no existente se produce una excepción no controlada y falla el programa.
- Al buscar vacuna no indica que dato hay que introducir (es el nombre) y si le pones el código de la vacuna sale null aunque exista la vacuna con ese código...y además no informa de que no la encuentra, solo pone null.
- Si llamo a avanzar fase y aún no hay vacunas o introduzco una vacuna no existente se produce una excepción no controlada y falla el programa.
- Si llamo a avanzar fase y aún no hay vacunas o introduzco una vacuna no existente se produce una excepción no controlada y falla el programa.
- Si ponemos a una vacuna que la fase 2 no ha sido superada, no almacena esta situación y solo reconoce las fases aprobadas.

- Al introducir las fases de la vacuna, no informa de la fase que le toca a la vacuna (tenemos que adivinar) y solo permite poner como fase superada no permite indicar fase no superada.
- Permite agregar la 3ª fase de la vacuna sin haber agregado las anteriores, pero después no la cuenta
- En la opción *Ver la última fase investigada de cada vacuna almacenada*, has implementado "Ver la última fase de una determinada vacuna", esto no es correcto, hay que mostrar la última fase de todas las vacunas independientemente de que hayan sido autorizadas o no. Y sino tienen ninguna fase investigada, se debe especificar.
- Parece que permite poner una fase superada antes de la anterior, aunque después no lo hace.

Autorizar/Rechazar:

- Cuando una vacuna ya está rechazada y la vuelvo a rechazar no me dice nada de que ya está realizada la operación.
- A la hora de autorizar una vacuna ya autorizada no avisa de que ya lo está y permite realizar la operación como si nunca la hubiesen autorizado.
- Permite rechazar una vacuna que ha sido autorizada.
- Si una vacuna tiene introducida solo una fase no la considera pendiente de autorizar/rechazar. Y sí está pendiente de autorizar /rechazar.
- Una vacuna puede ser rechazada en cualquier caso, no necesita que tenga alguna fase sin aprobar.
- Permite rechazar vacunas ya autorizadas.
- Al autorizar vacuna: Como no sé si tengo que poner el código o el nombre, pongo el código de una vacuna y me salta una excepción no controlada.
- Al repetir la autorización de una vacuna, no informa de que ya está autorizada.
- Ver vacunas rechazadas: Muestra aquellas que no tienen resultado en sus fases. Una cosa es no haber finalizado las fases y otra que éstas no hayan sido superadas.
- Si una vacuna tiene 3 fases aprobadas y se rechaza() aparece como autorizada después.
- En *VacunaAutorización* el método autorizar solo debe autorizar, en ningún caso rechazar. Lo mismo sucede con rechazar, que debe ser independiente de las fases completadas.
- Me permite autorizar cualquier vacuna, independientemente de las fases que tenga superadas.
- Está mal el procedimiento de rechazar() Una vacuna se puede rechazar aunque tenga fases superadas, incluso aunque tenga todas las fases superadas.