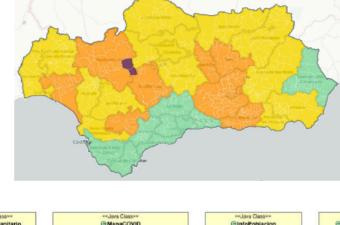
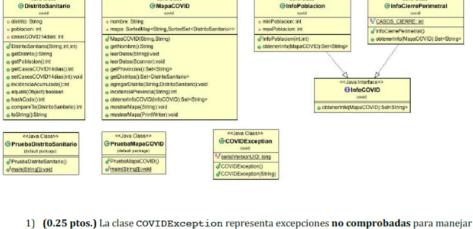


NOTAS PARA LA REALIZACIÓN DEL EJERCICIO: El alumnado debe haber comprobado si cumple los requisitos para acogerse a la modalidad

- de evaluación continua. De no ser así, además de la prueba práctica, deberá realizar un examen teórico adicional y, por lo tanto, ponerse en contacto con el profesorado. Al inicio del contenido de cada fichero realizado deberá aparecer un comentario con los
- apellidos y nombre, titulación y grupo. Está permitido:
- Consultar los apuntes (CV), la API (Internet), la guía rápida de la API (CV).
 - Añadir métodos privados a las clases. No está permitido:
- Intercambiar documentación con otros compañeros.
 - Recibir ayuda de otras personas. Se debe realizar personal e individualmente la solución del ejercicio propuesto.
 - Añadir métodos no privados a las clases.
 - Añadir variables o constantes a las clases. 0
 - 0 Modificar la visibilidad de las variables, constantes y métodos que aparecen en el
- diagrama UML. Modificar el código suministrado. Una vez terminado el ejercicio, se debe crear un archivo comprimido de la carpeta SRC del
 - proyecto y subirlo a la tarea correspondiente del campus virtual.
- La evaluación tendrá en cuenta la claridad de los algoritmos, del código y la correcta elección de las estructuras de datos, así como los criterios de diseño que favorezcan la reutilización.
- Para la corrección del ejercicio se utilizarán **programas de detección de copias/plagios**.
- Con posterioridad a la realización del ejercicio, el profesor podrá convocar a determinado/as alumno/as para realizar **entrevistas personales** con objeto de comprobar la autoría de las
- soluciones entregadas. Se desea desarrollar un proyecto que permita mantener información sobre la situación de la epidemia de COVID-19 en una determinada región, de forma que cada provincia se organizará según sus distritos

sanitarios. De este modo, se creará un proyecto prMapaCOVID, con las clases COVIDException, DistritoSanitario y MapaCOVID, así como la interfaz InfoCOVID y dos clases que la implementan: InfoPoblacion e InfoCierrePerimetral. Todas las clases estarán en el paquete covid, a excepción de las clases de prueba, que se definirán en el paquete por defecto. Se proporciona una clase de prueba, PruebaMapaCOVID, y un fichero de texto (datosCOVID-dist.csv) con información sobre los distritos sanitarios de las provincias andaluzas.





(String), la población a la que atiende (int) y el número de casos positivos COVID acumulados en los últimos 14 días (int). La clase incluirá: Un constructor al que se proporcionen como parámetros los valores iniciales de los atributos

2) (2.50 ptos.) La clase DistritoSanitario, que incluya información sobre su nombre

las situaciones excepcionales que puedan producirse en las siguientes clases.

- anteriores (en ese mismo orden). Debe validar todos los atributos, generando una COVIDException, si la población es cero o negativa, o si los casos diagnosticados son menor que cero.
- b. Tres métodos que permitan obtener los valores almacenados en los atributos nombre, población y número de casos positivos acumulados (getters), así como un método adicional que permita modificar este último atributo (setter). En este caso, si el valor que se pasa es un número negativo, se lanzará una excepción COVIDException. Además, se incluirá un método public int incidenciaAcumulada() que devuelva el número de casos positivos por cada 100.000 habitantes.
- c. Dos distritos sanitarios se consideran iguales cuando tienen el mismo nombre, sin distinguir mayúsculas y minúsculas. d. La clase proporcionará un orden natural, de forma que un distrito sanitario se considera menor
- que otro cuando el nombre (sin distinguir minúsculas y mayúsculas) es previo lexicográficamente. e. La representación textual de un distrito sanitario vendrá dada por: Distrito(nombre, número-de-casos)
- Por ejemplo, Distrito(Axarquía, 184) 3) (1,00 ptos.) La clase PruebaDistritoSanitario realizará las siguientes pruebas sobre las dos clases anteriores. Creará distritos sanitarios con la siguiente información:
 - Axarquía: 170141, args[1] Valle del Guadalhorce: 156298, args[2] Costa del Sol: 560785, args[3]

La Vega: 110176, args[0]

- Málaga Distrito: 633521, args[4] Serranía: 54999, args[5] Siendo String args[] el argumento de main.
 - Deben tratarse las distintas situaciones erróneas que puedan darse emitiendo un mensaje diferenciado de error (System.err). Estas situaciones incluirán: No se proporciona valores suficientes como argumentos del main. Alguno de los valores introducidos no se corresponde con un número entero. Alguno de los valores introducidos es negativo.

public void leerDatos(Scanner sc)

sanitarios de que consta. También incluirá un atributo con el nombre de la comunidad autónoma (String). Tanto las provincias como los distritos deberán mantenerse ordenados, según su orden natural. La clase incorporará: a. Un constructor que recibe como argumento el nombre de la región y el nombre de un fichero con

4) (3.75 ptos.) La clase MapaCOVID almacenará en una estructura la información del mapa COVID de una comunidad autónoma, asociando a cada provincia (String) el conjunto de los distritos

- información sobre provincias y distritos sanitarios (ver apartado (b)). b. Los métodos: public void leerDatos(String nomFich) throws FileNotFoundException
- añade la información almacenada en el fichero denominado nomFich o procedente del Scanner sc, respectivamente. Dicha información estará organizada por líneas con el siguiente Distrito-Sanitario(Provincia): población: número-de-casos Por ejemplo, Axarquía(Málaga):170141:184

Es decir, los delimitadores responderán a la expresión "[():]+". En caso de que un distrito sanitario aparezca más de una vez, se añadirá (ver apartado (c)) la información correspondiente a la

- primera aparición. Si alguna línea no tuviese el formato correcto, se omitirá su procesamiento, y se continuará con la siguiente línea.
- que agrega el distrito sanitario del segundo argumento a la provincia dada por el primer argumento. En caso de que el distrito ya estuviese asociado a la provincia, no se añade nada. d. Los métodos

public Set<DistritoSanitario> getDistritos()

public int incidenciaProvincia(String prov)

que devuelven el nombre de la comunidad autónoma, el conjunto de provincias y el de distritos sanitarios, respectivamente, almacenados en el mapa.

devuelve la incidencia acumulada cada 100.000 habitantes de la provincia indicada en el argumento. Para calcular la incidencia de una provincia se deberá obtener la población total y los

public void agregarDistrito(String prov, DistritoSanitario ds)

casos acumulados de todos los distritos de la provincia. Y con ambos datos, calcular la incidencia correspondiente a 100.000 habitantes. La incidencia de una provincia será 0 cuando no conste población o la provincia no esté en el mapa.

sangrados con un tabulador. Por ejemplo:

public String getNombre()

public Set<String> getProvincias()

c. El método

- f. Los métodos public void mostrarMapa(String nomFich) throws FileNotFoundException public void mostrarMapa(PrintWriter pw) que permiten mostrar la información del mapa COVID, bien sobre un fichero o a través de un PrintWriter, respectivamente. La información incluirá una línea inicial con el nombre de la comunidad autónoma en mayúsculas y líneas para cada provincia y sus distritos sanitarios,
- Distrito(Almería Distrito, 246) Distrito(Levante-Alto Almanzora, 119) Distrito(Poniente de Almería, 157) Cádiz Distrito(Bahía de Cádiz-La Janda, 542) Distrito(Campo de Gibraltar Este,93)
- 5) (2,5 ptos.) Con objeto de obtener información de naturaleza diversa a partir de un mapa COVID, defínanse la siguiente interfaz y clases: a. La interfaz InfoCOVID con un método
- Set<String> obtenerInfo(MapaCOVID mapa) que devolverá un conjunto de provincias o nombres de distritos del mapa que se pasa como argumento, cumpliendo determinadas condiciones a fijar en las clases que implementan la interfaz.

d. Añadir un método

ANDALUCÍA: Almería

- La clase InfoPoblacion que implementa la interfaz InfoCOVID, y proporcione:
 - Dos variables de instancia que establezcan un rango de población. Un constructor donde se proporcione el mínimo y el máximo de ese rango. Una implementación del método obtenerInfo(MapaCOVID), que devuelva el conjunto de
 - La clase InfoCierrePerimetral que implementa la interfaz InfoCOVID, y proporcione: Una implementación del método obtenerInfo(MapaCOVID), que devuelva el conjunto de provincias cuyas incidencias acumuladas superen los 500 casos.

nombres de distritos cuya población se encuentra entre los límites establecidos.

a la clase MapaCOVID que devuelva el conjunto de nombres de provincias o distritos determinado por el criterio definido en el objeto info que se pasa como argumento.

public Set<String> obtenerInfoCOVID(InfoCOVID info)