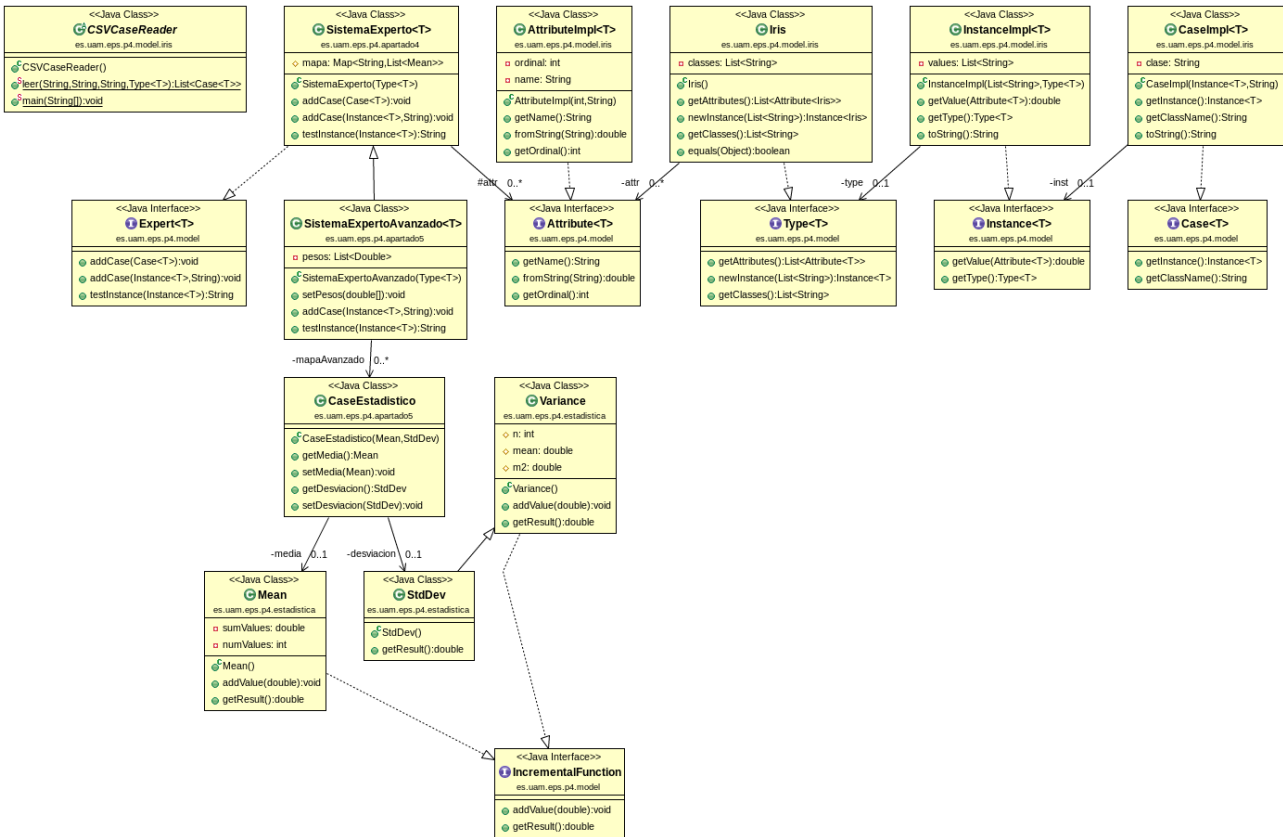


Práctica 4 ADSOF



Ejercicio 1

En el primer apartado simplemente nos dedicamos a implementar las clases e interfaces que se nos ha pedido. Cabe destacar el hecho de controlar las excepciones para que los formatos de los argumentos y los valores que puedan tomar tengan sentido.

Ejercicio 2

En el segundo apartado hemos implementado el lector del fichero y lo hemos probado en un main debajo, donde hemos impreso todos los casos leídos y vemos que funciona bien.

Ejercicio 3

En el tercer apartado hemos implementado las 3 operaciones incrementales como se nos ha pedido, y hemos probado en el TestEstadistica que los 3 calculan bien los resultados, salvo un error acumulado pues son calculos incrementales y no dan el resultado exacto al valor teórico como comprobamos al ejecutar el main de TestEstadistica.

En el caso de la media sí que lo calcula exacto pues no lo hemos hecho incremental sino que hemos ido almacenando la suma de valores y el numero de valores, con lo cual, no se calcula la media de forma incremental sino de forma exacta (aunque puede dar un error de overflow si se suman demasiados valores a `sumValues`)

Ejercicio 4

En el cuarto apartado es donde hemos tenido que pensar un poco más el diseño, pues hemos diseñado un sistema experto en el que tenemos una lista de atributos y un mapa de la clave que son las clases, y los valores que son la lista de medias de cada uno de los atributos asociadas a la clave. En el método `testInstance` es donde hemos tenido que codificar un algoritmo que se dedique a calcular la distancia euclídea entre vectores de medias para poder saber cuál de las clases tiene una menor distancia a la instancia que se pasa por argumento, y así poder clasificarlo.

Hemos probado este sistema experto en 2 testers, uno en el que las instancias se van añadiendo

según se clasifican y otro en el que no.

Podemos notar como si no las añadimos, el porcentaje de fallidos se mantiene más o menos constante alrededor del 10% en cada ejecución. Pero si lo hacemos añadiéndolo, podemos notar como el porcentaje de fallos oscila en un rango mayor, pasando de alrededor de un 2% de fallos a un 18%, esto es debido a que dependiendo de como haya sido permutada la lista de casos, puede ser que las clasificaciones originen errores consecutivos uno tras otro o que haya suerte y den pocos errores.

Ejercicio 5

En el quinto apartado hemos implementado la clase `SistemaExpertoAvanzado` el cual hereda de `SistemaExperto` del apartado anterior, para el cual hemos implementado una clase `CaseEstadistico` para poder almacenar las variables de `Mean` y `StdDev` que aparecen en nuestro `mapaAvanzado` como valores con respecto a la clave que son todos los atributos.

Si vamos probando con pesos diferentes, podemos ver que el numero de errores cambia dependiendo de a que atributo le pongas más peso, comprobando que si pones más peso a los 2 primeros atributos, los errores aumentan, y que si se lo pones al 3º, es donde menos fallos se dan. Y si usamos los pesos recomendados por el enunciado, los errores son mucho menores, hasta de un 1% en algunas de las ejecuciones.