

NOTAS PARA LA REALIZACIÓN DEL EJERCICIO:

- Al inicio del contenido de cada fichero realizado deberá aparecer un comentario con tus apellidos y nombre, titulación y grupo.
- Los diferentes apartados tienen una determinada puntuación. Si un apartado no se sabe hacer, *no debes pararte en él indefinidamente*. Puedes abordar otros.
- 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.
 - Modificar la visibilidad de las variables, constantes y métodos que aparecen en el diagrama UML.
 - Modificar el código suministrado.
- Una vez terminado el ejercicio, debéis subir (a la tarea creada en el campus virtual para ello) un fichero comprimido de la carpeta src que hayáis realizado y usáis vuestros apellidos y nombre para el nombre del mismo (**Apellido1Apellido2Nombre.rar** o **.zip**).
- 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 sincrónicas con objeto de comprobar la autoría de las soluciones entregadas.

Proyecto prEstelar

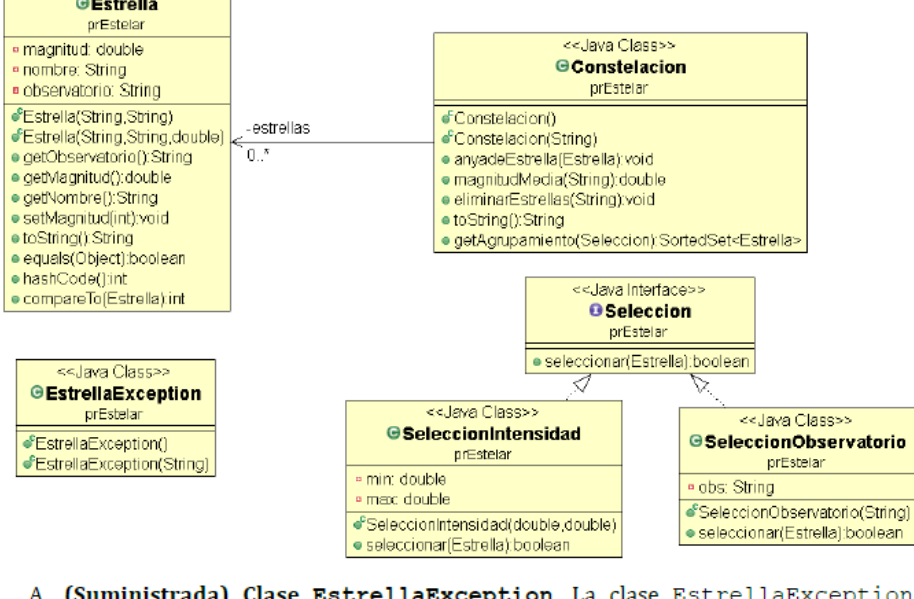
Se desea crear una aplicación para manipular la información relacionada con observaciones de estrellas. Se han de crear, dentro de un paquete con el mismo nombre del proyecto, las clases e interfaces que aparecen en el diagrama UML: *Constelacion*, *Seleccion*, *SeleccionObservatorio*, *SeleccionIntensidad*.

También se debe crear, en el paquete “por defecto” del proyecto, la clase denominada *TestConstelacion* para la prueba de las clases *Constelacion*, *SeleccionObservatorio* y *SeleccionIntensidad*.

Por otro lado, se suministra en el campus virtual la clase *EstrellaException*, la clase *Estrella* (sobre la que habrá que hacerse una modificación) y la clase *TestPrueba* para realizar pruebas de la clase *Constelacion*. Esta clase de prueba debe meterse en el paquete “por defecto” del proyecto.

Finalmente, se suministra también en el campus virtual un fichero “observaciones.txt” que debe copiarse en la carpeta principal del proyecto.

A continuación, se describen cada una de las clases pedidas:



A. (Suministrada) Clase **EstrellaException**. La clase *EstrellaException* se proporciona en el campus virtual. Dicha clase se utiliza para tratar las diferentes situaciones excepcionales. Se trata de una *excepción comprobada*.



(Suministrada) Clase **Estrella**. La clase *Estrella* se proporciona en el campus virtual. Ésta mantendrá información sobre el nombre de una determinada estrella (*String*), el nombre del observatorio donde fue registrada (*String*) y la magnitud luminosa (*double*). La clase dispone de un constructor y diversos métodos de instancia públicos como puede verse en el diagrama UML. Con el código suministrado puede observarse el funcionamiento de cada uno de ellos.

C. (0.75 *ptos.*) Clase **Estrella**. Modifica la clase *Estrella* de forma que sea una clase ordenable por su orden natural. El *orden natural* de dos objetos de la clase *Estrella* se define de la siguiente forma: es menor aquel objeto cuyo orden natural de los nombres de las estrellas sea menor (ignorando mayúsculas y minúsculas). En caso de nombres iguales, se aplica el orden natural los nombres de los observatorios (ignorando mayúsculas y minúsculas).

D. (5 *ptos.*) Clase **Constelacion**. La clase *Constelacion* mantendrá información sobre un *conjunto ordenado* de Estrellas (*SortedSet<Estrella>*), denominado *estrellas*. Puesto que los objetos de la clase *Estrella* pueden ser observaciones de la misma desde diferentes observatorios, es posible que existan dentro del conjunto *estrellas*, objetos con el mismo nombre de estrella y diferente observatorio y magnitud luminosa. La clase dispondrá de **dos** constructores y métodos de instancia públicos necesarios para:

- a. Construir un objeto de la clase. El primer constructor será un constructor sin argumentos, que creará un conjunto ordenado de estrellas vacío.

El segundo constructor tendrá un parámetro. Este será un *String* con el nombre de un fichero que contiene la información de las estrellas a almacenar en el conjunto. Si el fichero no puede abrirse (*FileNotFoundException*), se mostrará un mensaje de error por pantalla indicando tal situación, se creará un conjunto ordenado de estrellas vacío y el constructor terminará su ejecución. Si el fichero puede abrirse también se creará un conjunto ordenado de estrellas vacío, y a partir de la información contenida en el fichero se rellenará el conjunto de estrellas. En cada línea del fichero viene la información para crear una estrella con el formato:

nombre_observatorio;nombre_estrella;magnitud

Si el formato de la línea es correcto se añadirá el objeto al conjunto. Si se detecta algún error en una línea (algún dato no es del tipo correcto, falta algún dato o el objeto *Estrella* no puede construirse porque sus datos no son adecuados), dicha línea será ignorada, es decir, no se tendrá en cuenta y se seguirá con la siguiente. Cada vez que esto ocurra, se mostrará por pantalla un mensaje mostrando la línea descartada y el error correspondiente. Nota. No olvides usar el *useLocale(Locale.ENGLISH)* si usas *Scanner*, para el punto decimal de *double*.

- b. Añadir una estrella al conjunto de estrellas (*void anyadeEstrellas(Estrella)*).

c. Devolver la magnitud media para la estrella especificada como parámetro (sin distinguir entre mayúsculas y minúsculas) *void double magnitudMedia(String)*. Para ello se calcula la media de todas las magnitudes de los distintos observatorios para la estrella especificada. Por ejemplo, según los datos de la estrella *Rigel* proporcionados en el archivo de prueba (“observaciones.txt”), la magnitud media de *Rigel* sería: 0.4274 ((0.12+0.17+0.99222)/3). Si no hay ninguna estrella con ese nombre en la constelación se lanza la excepción *EstrellaException*.

d. Eliminar del conjunto ordenado *estrellas*, todas aquellas estrellas cuyo nombre coincida (sin tener en cuenta mayúsculas o minúsculas) con el que se pasa como argumento (*void eliminarEstrellas(String)*).

e. Obtener la representación del conjunto de Estrellas (*String toString()*) que viene dada por los elementos del conjunto separados por ‘,’ y encerrados entre corchetes. Ejemplo: [(Roque de los Muchachos;Rigel;0.12), (Mauna Kea;Vega;0.2), (Roque de los Muchachos;Vega;0.11), (Roque de los Muchachos;1.00789), (Mauna Kea;Sirius;0.99222)]

E. (Suministrada) Clase **TestPrueba**. Puedes probar la clase anterior con la clase de prueba *TestPrueba* proporcionada en el cv. El resultado de la ejecución sería el siguiente:

```
ERROR: Faltan datos en línea: Roque de los Muchachos;1.00789
Constelación:[(Roque de los Muchachos; Rigel; 0.17), (Roque de los Muchachos; Rigel; 0.12),
(Teide; Rigel; 0.99222), (Roque de los Muchachos; Sirius; 0.11), (Mauna Kea; Vega;
0.26)]
Media magnitud Luminosa :0.4274066666666667
Constelación después de eliminar Sirius:[(Mauna Kea; Rigel; 0.17), (Roque de los
Muchachos; Rigel; 0.12), (Teide; Rigel; 0.99222), (Mauna Kea; Vega; 0.26)]
```

F. (0.25 *ptos.*) Interfaz **Seleccion**. La interfaz *Seleccion* define un método que permite seleccionar una determinada *Estrella* en función de diferentes criterios. El método

boolean seleccionar(Estrella) devuelve *true* si la *Estrella* recibida debe ser seleccionado. En otro caso, devuelve *false*.

G. (0.75 *ptos.*) Clase **SeleccionIntensidad**. La clase *SeleccionIntensidad* implementa la interfaz *Seleccion*, y contiene información sobre dos valores *max* y *min* (*double*). Permite seleccionar una determinada estrella si su magnitud de brillo es mayor que *min* y menor que *max*. La clase dispondrá de un constructor y un método:

- a. El constructor *SeleccionIntensidad(double, double)* almacena en las variables *max* y *min* los valores recibidos como parámetros.
- b. El método *boolean seleccionar(Estrella)* devolverá *true* si el valor de la magnitud de brillo de la estrella que se pasa como parámetro, está comprendida entre esos dos valores (> *min* y < *max*).

H. (0.75 *ptos.*) Clase **SeleccionObservatorio**. La clase *SeleccionObservatorio* implementa la interfaz *Seleccion* y contiene información sobre el nombre de un observatorio (*String*). Permite seleccionar una estrella si ha sido observada desde el observatorio antes indicado. La clase dispondrá de un constructor y un método de instancia:

- a. El constructor *SeleccionObservatorio(String)* almacena el valor del parámetro pasado como argumentos en la variable *observatorio*.
- b. El método *boolean seleccionar(Estrella)* devolverá *true* si el observatorio de la *Estrella* recibida como parámetro coincide con el de la variable *observatorio* (sin tener en cuenta mayúsculas o minúsculas).

I. (1 *pto.*) Clase **Constelacion**. Añade a la clase *Constelacion* un método para devolver una colección de Estrellas (*Collection<Estrella> getAgrupamiento(Seleccion s)*). Éste devolverá una colección de objetos *Estrella* seleccionados de *estrellas*, de acuerdo al método *seleccionar* proporcionado por el objeto *s* recibido como parámetro, cuya clase implementa la interfaz *Seleccion*.

J. (1.5 *ptos.*) Clase **TestConstelacion**. Implementa la clase distinguida *TestConstelacion*. Ésta lleva a cabo las siguientes acciones:

1. Crea una constelación vacía
2. Añade las siguientes estrellas (nombre_observatorio,nombre_estrella,magnitud):
 - a. Teide, Sirius, 0.34.
 - b. Roque de los Muchachos, Rigel, 0.45
 - c. Mauna Kea, Sirius, 0.15
 - d. Roque de los Muchachos, Sirius, 0.16
3. Muestra la constelación
4. Muestra la magnitud media de “Sirius”
5. Utilizando el método *getAgrupamiento* muestra la colección de estrellas observadas en el observatorio de “Roque de los Muchachos”. A continuación, y utilizando el mismo método muestra la colección de estrellas cuyas magnitudes de luminosidad son mayores que 0.2 y menores que 0.4.
6. Añade por último la estrella:
 - a. Mauna kea, Vega, -0.4

La ejecución sería la siguiente:

```
Constelación:[(Roque de los Muchachos; Rigel; 0.45), (Mauna Kea; Sirius; 0.15),
(Roque de los Muchachos; Sirius; 0.16), (Teide; Sirius; 0.34)]
Media magnitud Luminosa Sirius:0.2166666666666667
Agrupamiento Roque de los Muchachos:[(Roque de los Muchachos; Rigel; 0.45), (Roque de
los Muchachos; Sirius; 0.16)]
Agrupamiento Intensidad (0.2,0.4):[(Teide; Sirius; 0.34)]
Error: Magnitud Luminosa negativa
```