



1 Material

Para la realización de esta práctica se dispone de los siguientes elementos contenidos en el fichero zip:

- **/doc/Enunciado.pdf**: fichero PDF con este enunciado
- **/data/**: carpeta de datos
 - **/data/wine_reviews.csv**: fichero CSV con datos de valoraciones de vinos
- **/src/fp.vinos.test**: paquete Java con las clases de test para las distintas clases que habrá que desarrollar en el proyecto
- **/src/fp.utiles**: paquete Java con utilidades de la asignatura

2 Datos disponibles

En este proyecto trabajaremos con datos sobre valoraciones de vinos. En estos datos encontramos solo un tipo de entidad:

- **Vino**: contiene la información relativa a la valoración de un determinado vino, con datos sobre la procedencia, precio y puntuación

Los datos están disponibles en formato CSV. En la siguiente figura se muestran las primeras líneas del fichero de datos.

	A	B	C	D	E
1	Country	Region	Points	Price	Grape
2	US	California	96	235.0	Cabernet Sauvignon
3	Spain	Northern Spain	96	110.0	Tinta de Toro
4	US	California	96	90.0	Sauvignon Blanc
5	US	Oregon	96	65.0	Pinot Noir
6	France	Provence	95	66.0	Provence red blend
7	Spain	Northern Spain	95	73.0	Tinta de Toro
8	Spain	Northern Spain	95	65.0	Tinta de Toro
9	Spain	Northern Spain	95	110.0	Tinta de Toro
10	US	Oregon	95	65.0	Pinot Noir
11	US	California	95	60.0	Pinot Noir

3 Ejercicios

EJERCICIO 1

Crear el tipo **Vino**, implementándolo como un *record* con las siguientes propiedades

Propiedades:

- **pais:** de tipo *String* con el país del vino. Consultable
- **region:** de tipo *String* con la región del vino. Consultable
- **puntos:** de tipo entero con la puntuación obtenida en la valoración, los puntos deben estar entre cero y cien. Consultable
- **precio:** de tipo *Double* con el precio del vino, el precio debe ser mayor que cero. Consultable
- **uva:** de tipo *String* con el tipo de uva del vino. Consultable
- **calidadPrecio:** de tipo *Double*. Se calcula como la calidad dividida por el precio. Consultable.

Constructor:

- **C1:** Crea un objeto tomando como parámetros las propiedades básicas del mismo en el orden en el que se describen arriba.

Representación como cadena:

- Se muestran todas las propiedades básicas del tipo.

Criterio de igualdad:

- Dos objetos de tipo **Vino** son iguales si todas sus propiedades básicas son iguales.

EJERCICIO 2

Crear la clase **FactoriaVinos** con los siguientes métodos estáticos

- **FactoriaVinos::parsearVino:** método privado para construir un objeto **Vino** a partir de una línea CSV del fichero de entrada
- **FactoriaVinos::leerVinos:** método que devuelve un objeto **Vinos** a partir de la ruta del fichero en el que se encuentran los datos de los vinos

EJERCICIO 3

Crear la clase **Vinos** con las siguientes propiedades y operaciones

Apartado a

Propiedades:

- **vinos:** de tipo conjunto de **Vino**. No es consultable. Se modifica a partir de las operaciones que se describen más abajo.

Constructores:

- **C1:** constructor sin parámetros. Crea un objeto sin vinos.
- **C2:** constructor a partir de un *Stream* de **Vino**. Crea un objeto de tipo **Vinos** con los vinos del *Stream* que se pasa como parámetro.

Representación como cadena:

- Muestra el número total de vinos incluidos en el objeto.

Criterio de igualdad:

- Dos objetos de tipo **Vino** son iguales si los son los vinos que contienen.

Otras operaciones:

Apartado b – Operaciones de manipulación

- **Vinos::agregarVino:** añade un Vino dado como parámetro al objeto de tipo Vinos sobre el que se aplica.
- **Vinos::eliminarVino:** elimina un vino. Si no existe, lanza `IllegalArgumentException`
- **Vinos::obtenerNumeroVinos:** calcula el número total de vinos.
- **Vinos::contieneVino:** devuelve true si el contenedor contiene un objeto de tipo Vino dado como parámetro.
- **Vinos::agregarVinos:** añade todos los vinos de una colección dada como parámetro al objeto de tipo Vinos que lo invoca.
- **Vinos::contieneVinos:** devuelve cierto si el contenedor contiene todos los vinos de una colección de vinos dada como parámetro.

Apartado c – Tratamientos secuenciales simples

- **Vinos::calcularNumeroVinosDePais:** cuenta el número de vinos de un país dado como parámetro.
- **Vinos::obtenerVinosRangoPuntos:** devuelve una colección de objetos de tipo Vino solo con los vinos que estén valorados en un rango de puntos determinado. El rango vendrá especificado por dos enteros (inf y sup) dados como parámetros. Si el valor del límite inferior del rango es superior al valor del límite superior, se elevará `IllegalArgumentException`.
- **Vinos::calcularNumeroVinosDePaisConPuntuacionSuperior:** devuelve el número de vinos del país dado como parámetro que tienen una puntuación superior a un umbral dado como parámetro.
- **Vinos::obtenerVinosBaratos:** obtiene un conjunto con los vinos cuyo precio es inferior a uno dado como parámetro.
- **Vinos::existeVinoDeUvaEnRegion:** devuelve true si existe en la región dada como parámetro un vino elaborado con la uva dada como parámetro

Apartado d – Tratamientos secuenciales de acumulación

- **Vinos::calcularUvasDeRegion:** devuelve un conjunto con los nombres de las uvas que se usan en los vinos de una región dada como parámetro.
- **Vinos::calcularTotalPuntosVinosDeRegion:** devuelve la suma de las puntuaciones de todos los vinos de una región dada como parámetro.
- **Vinos::calcularMediaPuntosVinosDeUva:** devuelve la puntuación media de los vinos obtenidos a partir de un tipo de uva dado como parámetro. Si la media no se puede calcular, devuelve cero.

Apartado e – Tratamientos secuenciales con criterios de ordenación

- **Vinos::obtenerVinoMejorPuntuado:** devuelve el objeto de tipo Vino con la puntuación más alta. Si no se puede calcular eleva `NoSuchElementException`.
- **Vinos::obtenerVinoMejorPuntuadoDePais:** devuelve el objeto de tipo Vino con la puntuación más alta de un país dado como parámetro. Si no se puede calcular eleva `NoSuchElementException`.
- **Vinos::obtenerNVinosRegionOrdenadosPrecio:** devuelve una lista ordenada con los N vinos más caros de una región dada como parámetro, ordenados del más caro al más barato.

Apartado f – Tratamientos secuenciales con Map

- **Vinos::agruparVinosPorPais:** devuelve un Map que asocia a cada país una lista con los objetos de tipo Vino de ese país.
- **Vinos::agruparUvasPorPais:** devuelve un Map que asocia los países y con conjuntos que contienen los nombres de las uvas usadas en los vinos del respectivo país.
- **Vinos::calcularCalidadPrecioPorRegionMayorDe:** devuelve un Map que asocia las regiones con el número de vinos cuya relación calidad/precio supera un umbral dado como parámetro.