



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

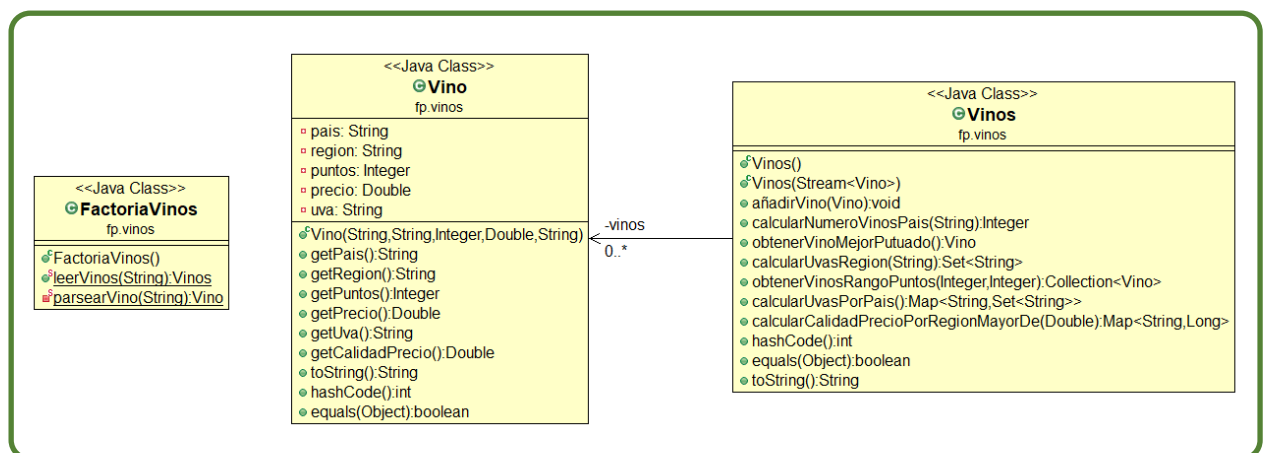
Los datos están disponibles en formato CSV. En la siguiente figura se muestran las primeras líneas del fichero de datos.

	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 Modelo

En el siguiente diagrama se muestran todos los elementos que habrá que implementar en este proyecto. Todos ellos se incluirán en el paquete **fp.vinos**. Los aspectos más destacables del modelo son:

- **Vino**: clase para implementar el tipo básico
- **Vinos**: tipo contenedor que incluye, además, algunos métodos de consulta basados en tratamientos secuenciales.
- **FactoriaVinos**: clase para dar soporte a la creación de objetos **Vino** y **Vinos** a partir de datos en un fichero CSV



Este diagrama ha sido generado con el plugin de Eclipse ObjectAid
URL de instalación: <http://www.objectaid.com/update/current>

4 Ejercicios

EJERCICIO 1

Crear la clase **Vino** con los siguientes atributos

- **pais**: atributo *String* con el país del vino
- **region**: atributo *String* con la región del vino
- **puntos**: atributo entero con la puntuación obtenida en la valoración, los puntos deben estar entre cero y cien
- **precio**: atributo *Double* con el precio del vino, el precio debe ser mayor que cero
- **uva**: atributo *String* con el tipo de uva del vino

EJERCICIO 2

Crear los siguientes métodos de la clase **Vino** comprobando las restricciones de los atributos en los casos en los que sea necesario

- **Vino**: constructor de la clase a partir de los atributos en el orden que se indica en el ejercicio anterior
- Métodos *getters*: para todos los atributos de la clase
- **Vino::getCalidadPrecio**: propiedad derivada que se calcula dividiendo los puntos por el precio
- **Vino::toString**: mostrando todos los atributos
- **Vino::equals**: usando todos los atributos para determinar la igualdad
- **Vino::hashCode**: usando la misma selección de atributos que el método **equals**

EJERCICIO 3

Crear la clase **Vinos** con los siguientes atributos y métodos

- **vinos**: atributo con un conjunto de objetos **Vino**
- **Vinos**: constructor vacío de la clase **Vinos**
- **Vinos**: constructor de la clase **Vinos** a partir de un *Stream* de **Vino**
- **Vinos::añadirVino**: método para añadir un **Vino** al conjunto de vinos
- **Vinos::toString**: mostrando todos los atributos
- **Vinos::equals**: usando el atributo **vinos** para determinar la igualdad
- **Vinos::hashCode**: usando la misma selección de atributos que el método **equals**
- **Vinos::calcularNumeroVinosPais**: cuenta el número de vinos dado un país
- **Vinos::obtenerVinoMejorPuntuado**: busca el vino con la puntuación más alta
- **Vinos::calcularUvasRegion**: conjunto de uvas usadas en los vinos de una región dada
- **Vinos::obtenerVinosRangoPuntos**: calcula una colección de **Vino** solo con los vinos que estén valorados en un rango de puntos determinado por un valor mínimo y otro máximo. Ambos parámetros son enteros, y se debe comprobar que el valor mínimo es menor o igual que el valor máximo.
- **Vinos::calcularUvasPorPais**: calcula un diccionario cuyas claves son los países y los valores el conjunto de uvas usadas en los vinos de cada país
- **Vinos::calcularCalidadPrecioPorRegionMayorDe**: calcula un diccionario cuyas claves son las regiones y los valores son el número de vinos cuya relación calidad/precio supera un umbral dado

EJERCICIO 4

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