



## Sesión 1. Números racionales

Crea un proyecto Java de nombre, *sesión-01*, y cambia las propiedades de éste abriendo el menú contextual (pulsa sobre el proyecto el botón derecho del ratón) y cambia la codificación de caracteres a UTF-8 (los archivos que se proporcionen para las prácticas siempre utilizaran este código de caracteres). Otra opción es cambiar la codificación de caracteres del *workspace* a UTF-8 antes de crear el proyecto, para hacer esto hay que cambiar las preferencias del Eclipse y expandir *General*, las preferencias del *Workspace* están al final de árbol *General*.

En todos los proyectos Java para prácticas se crearán, al menos, dos *packages*: *estDatos* y *app*. El primero para las abstracciones y el segundo para los programas cliente.

En esta primera práctica se implementará, al menos, un tipo de dato para números racionales y se hará alguna modificación al TDA Racional (interfaz) que se proporciona.

*Material proporcionado para la práctica* (descarga del CV y descomprime el archivo *sesión-01.zip*):

1. El archivo *pares.jar* que contine los archivos *class* del TDA  $\text{Pair}\langle K, V \rangle$  de pares de elementos, que se está utilizando en las clases de teoría, la clase abstracta  $\text{AbstractPair}\langle K, V \rangle$  que implementa parcialmente la interfaz previa para proporcionar las operaciones *equals* y *toString* para los pares y, por último, dos implementaciones concretas, los tipos  $\text{PairImp}\langle K, V \rangle$  y  $\text{PairImpAlt}\langle K, V \rangle$ . Para la práctica pueden utilizarse indistintamente cualquiera de estos dos tipos de datos, ya que la única diferencia entre ambos es la forma de representar los pares y esto es irrelevante para su uso en los clientes.
2. Una carpeta de nombre *doc* con la documentación de la interfaz y clases citadas en el apartado previo. Para acceder a la documentación abrir el archivo *index.html*.
3. Dos carpetas de nombre *app* y *estDatos* con archivos fuente de Java que deberás copias a los *packages* del mismo nombre. En la carpeta *app* puedes encontrar un programa de ejemplo de uso de racionales al que puedes añadir más código para probar los racionales. La carpeta *estDatos* contiene el TDA *Racional* (nótese que el TDA establece que el signo del número racional será el signo de su numerador; es decir, que el denominador siempre quedará como positivo).

Se pide:

1. Crear una clase abstracta de nombre *AbstractRacional* que implemente parcialmente el TDA *Racional*. Inicialmente esta clase únicamente proporcionará la operación *toString* para números racionales. La cadena de caracteres que representará a un racional deberá ser la habitual; es decir,  $n/d$  siendo  $n$  su numerador y  $d$  su denominador, excepto si  $d=1$ , en cuyo caso simplemente será  $n$ .
2. Implementar el *tipo de dato inmutable* (o *no modificable*) *RacionalNoMod*, extendiendo la clase abstracta del apartado anterior y almacenando la información en un par (una instancia de  $\text{Pair}\langle K, V \rangle$ ). Este tipo de dato deberá incluir un *constructor de conversión* (*RacionalNoMod(Racional r)*).
3. Dado que los números racionales se pueden ordenar (tienen *orden natural*), modificar la interfaz *Racional* para que extienda la interfaz *Comparable<Racional>* (recuérdese que la interfaz es *Comparable<T>* y, en este caso, lo que se van a comparar son números racionales; es decir,  $T$  no es cualquier tipo, tiene que ser *Racional*). Implementa la operación *compareTo* donde consideres más conveniente.
4. Redefinir la operación *equals* para los números racionales en la clase abstracta.
5. (Complementario) Implementar el *tipo de dato mutable* (o *modificable*) *RacionalMod*, pero en lugar de extendiendo la clase abstracta, extendiendo el tipo *RacionalNoMod*.

**Nota:** para disponer de los pares en el proyecto deberás abrir el cuadro de diálogo **Java Build Path** y añadir el *jar* como *jar* externo.