

**Proyecto: Premios Nobel**

Autor: Daniel Mateos, José A. Troyano

Revisor: Fermín Cruz, Toñi Reina, Mariano González, Belén Vega

Última modificación: 27/04/2022



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/nobel_prizes.csv**: fichero CSV con datos de premios nobel
- **/src/fp.nobel**, **/src/fp.nobel.test**: paquetes Java para almacenar 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 sobre datos de premios nobel. En estos datos encontramos solo un tipo de entidad:

- **Premio**: contiene la información relativa a un premio nobel, para una edición y categoría determinadas

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	F
1	year	category	firstname	surname	gender	birthyear
2	1901	physics	Wilhelm Conrad	Röntgen	male	1845
3	1902	physics	Hendrik Antoon	Lorentz	male	1853
4	1902	physics	Pieter	Zeeman	male	1865
5	1903	physics	Antoine Henri	Becquerel	male	1852
6	1903	physics	Pierre	Curie	male	1859
7	1903	physics	Marie	Curie née Sklodowska	female	1867
8	1911	chemistry	Marie	Curie née Sklodowska	female	1867
9	1904	physics	Lord Rayleigh	(John William Strutt)	male	1842
10	1905	physics	Philipp Eduard Anton	von Lenard	male	1862
11	1906	physics	Joseph John	Thomson	male	1856

3 Ejercicios

EJERCICIO 1

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

Propiedades:

- **año:** de tipo *Integer* con el año del premio. Consultable.
- **categoría:** de tipo *String* con la categoría del premio. Consultable.
- **nombre:** de tipo *String* con el nombre del premiado. Consultable.
- **apellidos:** de tipo *String* con los apellidos del premiado. Consultable.
- **genero:** de tipo *Genero* con el género del premiado. Puede tomar los valores MALE y FEMALE. Consultable.
- **añoNacimiento:** de tipo *Integer* con el año de nacimiento del premiado. Este año debe ser menor que el año del premio. Consultable.
- **edadPremiado:** de tipo *Integer* con la edad del premiado en el momento de recibir el premio. Se estimará mediante la diferencia entre el año del premio y el año de nacimiento. 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 **Premio** son iguales si lo son su año, categoría, nombre y apellidos.

EJERCICIO 2

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

- **FactoriaNobel::parsearPremio:** método privado para construir un objeto **Premio** a partir de una línea CSV del fichero de entrada
- **FactoriaNobel::leerPremios:** método que devuelve un objeto **Premios** a partir de la ruta del fichero en el que se encuentran los datos de los premios

EJERCICIO 3

Crear la interfaz **Premios** con los siguientes métodos

- **Premios::añadirPremio:** añade un **Premio** al objeto de tipo **Premios** que lo invoca.
- **Premios::obtenerPremiosDeGenero:** devuelve una colección de **Premio** que contiene los premios del género dado como parámetro.
- **Premios::calcularNumeroPremiadosMasJovenesDe:** devuelve el número de premiados que recibieron el premio con una edad inferior a la dada como parámetro.
- **Premios::calcularNumeroPremiosPorGenero:** devuelve un Map que asocia los géneros (clave) con el número de premiados de ese género (valor).
- **Premios::calcularPremiosPorEdad:** devuelve un Map que asocia una edad (clave) una lista de objetos de tipo **Premio** con los premios que se recibieron con esa edad.
- **Premios::calcularMediaEdadPorCategoria:** devuelve un Map cuyas claves son las categorías y los valores son la media de edad de los premiados en esa categoría

EJERCICIO 4

Crear la clase **PremiosStream** que implemente la interfaz **Premios**, con los siguientes atributos y métodos. En esta clase, aquellos métodos que requieran tratamientos secuenciales se implementarán con *streams*.

Propiedades:

- **premios:** de tipo conjunto de Premio. No consultable.

Constructores:

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

Representación como cadena:

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

Criterio de igualdad:

- Dos objetos de tipo Premios son iguales si los son los premios que contienen.

EJERCICIO 5

Crear la clase **PremiosBucle** que implemente la interfaz **Premios**, con los siguientes atributos y métodos. En esta clase, aquellos métodos que requieran tratamientos secuenciales se implementarán con bucles.

Propiedades:

- **premios:** de tipo conjunto de Premio. No consultable.

Constructores:

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

Representación como cadena:

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

Criterio de igualdad:

- Dos objetos de tipo Premios son iguales si los son los premios que contienen.