



## NORMAS DE ENTREGA

---

### **LEA DETENIDAMENTE LAS NORMAS DE LA PRUEBA**

**Cualquier incumplimiento de las normas significará una respuesta nula en el examen y por tanto una valoración de 0 puntos.**

- La resolución de los ejercicios propuestos se entregará en un único archivo formato ZIP que contendrá EXCLUSIVAMENTE los proyectos de Netbeans que compongan la solución a los problemas planteados.
- El nombre del fichero tendrá un formato específico dictado por el nombre de cada alumno. Por ejemplo, para un alumno llamado “José María Núñez Pérez” el fichero se nombrará como NunyezPerezJM.zip. Obsérvese que las tildes son ignoradas y las eñes sustituidas.
- El fichero se subirá utilizando la correspondiente tarea en el aula virtual.

**IMPORTANTE:** Cualquier envío que no respete el formato de compresión o el nombre adecuado será ignorado y, por tanto, valorado con cero puntos.



## Problema 1

En el circuito profesional de tenis la información de cada tenista se gestiona usando una aplicación. Implemente un programa que simule algunos aspectos relacionados con la mencionada gestión de jugadores de tenis, para ello:

Cree una clase **Tenista** que incluya los siguientes atributos (con sus correspondientes métodos consultores y modificadores):

- **nombre: String**
- **numTitulos: int**
- **edad: int**
- **rankingATP: int**

Implemente los métodos **toString**, **equals** y **hashCode** en la clase y considere que el orden natural es posición en el ranking de la ATP. Dos objetos **Tenista** serán iguales si coinciden exactamente los valores de sus cuatro atributos.

Cree una clase **TenistasATP** que contenga:

- Un atributo **tenistas** de tipo privado con el conjunto de tenistas.
- El método **boolean tenistaNuevo(Tenista t)** que añadirá un tenista nuevo al atributo **tenistas** devolviendo un booleano cuyo valor dependerá de si ha sido o no añadido a dicho atributo **tenistas**. Hay que indicar que en el programa de gestión no se permiten tenistas duplicados.
- El método **tenistasTop(int ranking)** que devolverá un conjunto de los mejores tenistas hasta la posición del ranking indicada como parámetro de entrada. El conjunto devuelto estará ordenado en base a su orden natural.
- El método **tenistasSenior()** que devolverá un conjunto de tenistas con edad mayor a 30 años ordenados por el número de títulos acumulados.

Implemente un programa principal que añada los siguientes tenistas usando el método **tenistaNuevo**:

Tenista 1: Rafael, 67, 33 años, 2.  
Tenista 2: Ariel, 27, 28 años, 4.  
Tenista 3: Rodrigo, 15, 21 años, 3.  
Tenista 4: Mario, 48, 31 años, 1.  
Tenista 5: Jannik, 115, 25 años, 5.  
Tenista 6: Novak, 87, 29 años, 6.  
Tenista 7: Carlos, 187, 22 años, 7.

Finalmente añada algunas líneas de código que muestren en la salida de texto las siguientes llamadas a los métodos:

- **tenistasTop(3)**
- **tenistasSenior()**