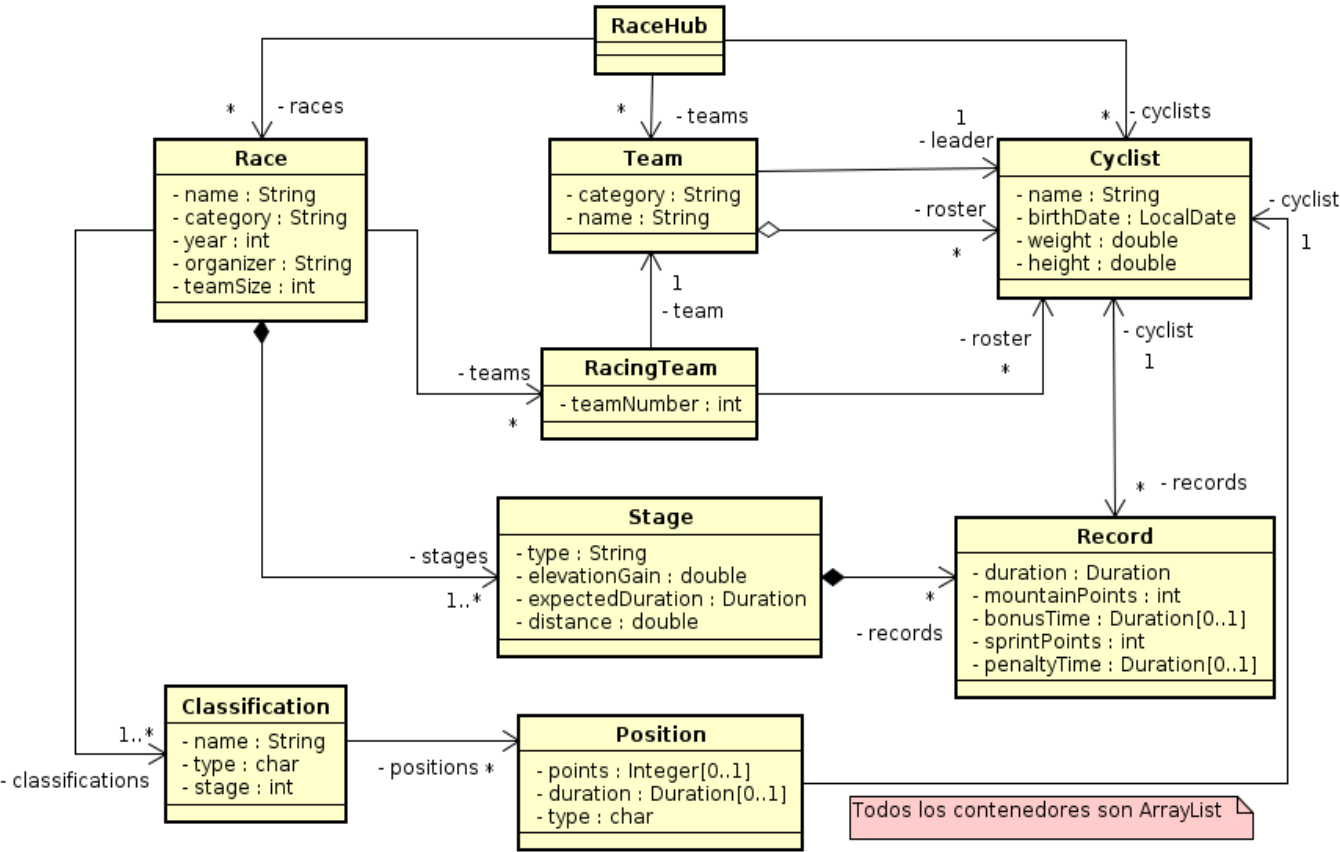


RaceHub

La Escuela quiere promover el deporte de ciclismo entre sus estudiantes. Para esto ha decidido desarrollar un sistema de información que le permita mantener la información de los diferentes equipos y gestionar las carreras que hará entre sus estudiantes. La decanatura ha designado a los estudiantes de POOB de este semestre para construir una solución que permita.

- Registrar equipos.** Los líderes deben inscribir su equipo en la categoría correspondiente, indicando la información de cada uno de los ciclistas miembros.
- Crear de carreras:** La Escuela es la responsable de crear las diferentes carreras. Una carrera consta de un conjunto de etapas.
- Inscribir equipos a carreras.** Los equipos seleccionan un subgrupo de sus corredores para que los representen en las carreras. El tamaño del subgrupo depende de la carrera.
- Grabar registros parciales:** Después de cada etapa los organizadores graban un registro parcial de cada corredor. Dicho registro contiene el tiempo total invertido, los puntos de montaña (metas intermedias o finales de montaña), puntos de sprints (metas intermedias o finales en llano), y los bonos de tiempo obtenidos. Dichos bonos de tiempo se miden en segundos y son restados del tiempo total invertido por el corredor en la etapa al momento de calcular las clasificaciones. Por el contrario los tiempos de penalización son adicionados y corresponden a castigos que se realizan a los corredores por infringir algunas reglas.
- Mantener clasificaciones.** Los organizadores de las carreras, después de cada etapa, generan o actualizan las diferentes clasificaciones. Una clasificación es el ordenamiento de los corredores que participan en una carrera de acuerdo a un criterio (T o P): tiempo acumulado o puntos (montaña o sprints). Las clasificaciones por tiempo se ordenan de forma ascendente, mientras que las clasificaciones por puntos lo hacen de forma descendente.



Class LocalDate - Method Summary		Class Duration - Method Summary	
boolean	<b>isAfter(LocalDate other)</b> Checks if this date is after the specified date.	Duration	<b>multipliedBy(long multiplicand)</b> Returns a copy of this duration multiplied by the scalar.
int	<b>getYear()</b> Gets the year field	Duration	<b>plus(Duration duration)</b> Returns a copy of this duration with the specified duration added.
boolean	<b>isEqual(LocalDate other)</b> Checks if this date is equal to the specified date.	Duration	<b>minus(Duration duration)</b> Returns a copy of this duration with the specified duration subtracted.
Static LocalDate	<b>now()</b> Obtains the current date from the system clock	Static Duration	<b>ZERO</b> Constant for a duration of zero