Nombre y Apellido: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Cantidad de hojas entregadas: \_\_ Nota:\_\_\_\_\_\_\_\_

|  |
| --- |
| **Lea atentamente TODO el enunciado antes de empezar a desarrollar, prestar mucha atención a las firmas de las funciones, y a los nombres de las clases. Todas deben ser respetadas para la correcta corrección del examen.** |

1. Lionel Scaloni (DT de la Selección Argentina de Futbol), todavía tiene dudas sobre como armar la lista para la copa América 2019. Para facilitarle la tarea al DT, nos pide que le hagamos una clase para solucionar su problema. (6ptos)

Se solicita crear la clase Jugador. El mismo debe recibir y almacenar 3 números enteros, de valor entre 0 y 99, correspondientes a fuerza, velocidad y definición y un String que contenga la posición.

(No olvidar los getters correspondientes getFuerza, getVelocidad, getDefinicion, getPosicion)   
Crear la clase SeleccionArgentina, donde al iniciar cada objeto la misma clase exigirá al usuario que complete la cantidad de Defensores, Mediocampistas y Delanteros, siendo 10 la cantidad total de jugadores de campo (Por ejemplo, un equipo que tenga una formación 4-4-2 tendrá 4 DEF, 4 MED, 2 DEL). Si la cantidad de jugadores totales en el constructor es diferente a 10, o se intenta crear un jugador de atributos no validos, lanzar la excepción *IllegalArgumentException()*

La clase deberá almacenar a los jugadores dentro de un mismo conjunto, cuando se instancie una nueva SeleccionArgentina se deberán crear a los jugadores correspondientes a las posiciones solicitadas, con características random y que sean ubicados dentro del conjunto. Las características, si bien son al azar, deben cumplir una cierta reglamentación, siendo que los atributos de los jugadores son números de entre 0 y 99

* Si el jugador es **DEF**, la FUERZA debe superar al menos un 30% a la VELOCIDAD y la DEFINICION no puede un valor de más de 40.
* Si el Jugador es **MED,** la VELOCIDAD debe ser el doble de la DEFINICION, la FUERZA no tiene restricciones.
* Si el jugador es **DEL,** la DEFINICION debe ser exactamente un 20% mayor que la FUERZA y la VELOCIDAD no puede ser menor a 20.
  1. Generar la primitiva **public Jugador getParecidoAMessi(){}** que devuelve un Jugador que tenga velocidad > 95 y definición > 89, si no encuentra ninguno con esas características devuelve null.

ACLARACION: No hay información respecto al arquero, no es necesario hacer nada

AYUDA: Para generar un número random dentro de un rango puede utilizar la siguiente función (copiarla a su código)

|  |
| --- |
| import java.util.Random;  private static int getRandomNumberInRange(int min, int max) {  if (min >= max) {  throw new IllegalArgumentException("max debe ser mayor que min");  }  Random r = new Random();  return r.nextInt((max - min) + 1) + min;  } |

3) Utilizando la clase Vector implementada durante los talleres. (4 ptos)  
Genere dentro de la clase Vector el método vectorFactorial que modifica al Vector cambiando sus coordenadas X,Y. a sus coordenadas factoriales.

El factorial de un entero n, el factorial de n o n factorial se define en principio como el [producto](https://es.wikipedia.org/wiki/Multiplicaci%C3%B3n) de todos los números enteros positivos desde 1 (es decir, los [números naturales](https://es.wikipedia.org/wiki/N%C3%BAmeros_naturales)) hasta n.

https://lh3.googleusercontent.com/TW6Jl3ni2dVoD3w7RSbSBNHO2KkL09s5QQoa-AHUBkELUSGUKnBCZsKzH0F5PQpH8sbgR2IF0NsuGQD8aAHEMCHYMOq7pEsqDZsyLwOOk4bwl4XTU0NOUmbHcJIORdX1PiuwlNC_oUx1nc5MoQ

Por ejemplo:

5! = 5 ∙ 4 ∙ 3 ∙ 2 ∙ 1 = 120

6! =  6 ∙ 5 ∙ 4 ∙ 3 ∙ 2 ∙ 1 = 720

Es decir se solicita que si inicialmente se tenía el vector:

(X,Y)

El resultado luego de aplicar el método debe ser:

(X!,Y!)

Precondición: Suponga las coordenadas del vector inicial números enteros positivos.