

Desafío 1, Crear un branch con el nombre feature/desafio-java-1.

El objetivo es repasar ciclos, dobles con vectores, máximo, mínimo, promedio.

-La escuela primaria xyz solicita el desarrollo de una aplicación para poder llevar el control de notas para cada alumno.

-El programa solicitará el ingreso y la cantidad de alumnos a procesar.

-Para cada alumno se deberá ingresar los siguientes datos:

- nombre (string)
- apellido (string)
- número de examen
- nota que obtuvo el alumno (numérico $\geq 0 \leq 10$)

Al finalizar la carga de datos, se procederá a listar la siguiente información:

- 1- alumno/s con la calificación más alta
- 2- alumno/s con la calificación más baja
- 3- alumnos que promocionan, aquellos que obtuvieron un promedio ≥ 7
- 4- alumnos que deben recursar la materia, aquellos que obtuvieron un promedio < 7

Desafío 2, Crear un branch con el nombre feature/desafio-java-2.

El objetivo es asimilar los conceptos de clases y objetos, interfaces y clases abstractas.

Una torre de control tiene la responsabilidad de gestionar el aterrizaje de objetos que sobrevuelan el espacio aéreo.

Para esto, dispone de un sistema que permite que los Voladores sean capaces de aterrizar en la pista 1.

Pero solamente permite que ciertos Voladores puedan realizar esta acción, por este motivo la torre tiene un contrato que deben cumplir. Esto permite que no todo objeto aterrice.

Aquellos que son voladores, pero NO cumplan con este contrato, debe pedir autorización adicional y dependerá de la confirmación de un operario, quien dirá SI o No.

Si el operador acepta podrán aterrizar en la pista 2, de lo contrario se informará que no es posible aterrizar.

Nota: Para esto desarrollar las clases e interfaces que considere necesario.

- Cargar los objetos en una lista y recorrer cada elemento identificando quien pueden y quienes no descender.

Desafío 3, Crear un branch con el nombre feature/desafio-java-3.

El objetivo es crear una clase CollectionCustom<T>

-La clase tiene un atributo T[] “un array de T” en el cual se deben guardar los tipos una vez creada la implementación.

- debe tener los métodos

size():int

addFirst(T): void

addLast(T):void

add(T):void

remove(T): T remueve el primer elemento indicado y lo retorna.

removeAll(T): void remueve todos los elementos.

boolean empty(): indica si la colección está vacía.

Crear la clase Main para probar cada método.

Nota:No usar Collection (Set, List, Collection etc) vistas en clase, deben trabajar con array de T, uds deben implementar la lógica.