

ANÁLISIS

Responda las siguientes preguntas como parte de su análisis:

- ¿Qué acciones debe poder hacer su programa? Enumerar.
 - Permitir el ingreso de datos para cada tipo de jugador de forma manual.
 - Guardar y cargar los datos por csv.
 - Mostrar todos los jugadores registrados.
 - Mostrar el top 3 de líberos.
 - Mostrar el 80% mejor de pasadores.
- ¿Con qué datos va a trabajar?
 - Con los datos brindados manualmente e ingresados en el csv
- ¿Qué clases necesitará para dar solución a la situación planteada? ¿Qué propiedades y métodos tendrá cada clase? ¿Qué tipo deben tener las propiedades y métodos de cada clase? ¿Cuáles deben ser los modificadores de visibilidad de los miembros en cada clase? ¿Qué parámetros serán requeridos por los métodos en sus clases?

Listado de clases:

Clase	Propiedad	Tipo	Nombre	Visibilidad	Parámetros
Jugador abstract (padre)	atributo	String	nombre	protected	N/A
	atributo	String	pais	protected	N/A
	atributo	int	errores	protected	N/A
	atributo	int	aces	protected	N/A
	atributo	int	totalServicios	protected	N/A
	atributo	int	efectividad	protected	N/A
	método	Jugador	Jugador	public	nombre, pais, errores, aces, totalServicios
	método	String	getNombre	public	N/A
	método	String	getPais	public	N/A
	método	int	getErrores	public	N/A
	método	int	getAces	public	N/A
	método	int	getTotalServicios	public	N/A
	método	int	getEfectividad	public	N/A
	método	void	calcularEfectividad	public	(depende del hijo)
	método	String	toString	public	N/A

Libero (Hijo de Jugador)	atributo	int	recibosEfectivos	private	N/A
	método	Libero	Libero	public	nombre, pais, errores, aces, totalServicios, recibosEfectivos
Pasador (Hijo de Jugador)	atributo	int	pases	private	N/A
	atributo	int	fintasEfectivas	private	N/A
	método	Pasador	Pasador	public	nombre, pais, errores, aces, totalServicios, pases, fintasEfectivas
Opuesto (Hijo de Jugador)	atributo	int	ataques	private	N/A
	atributo	int	bloqueosEfectivos	private	N/A
	atributo	int	bloqueosFallidos	private	N/A
	método	Opuesto	Opuesto	public	nombre, pais, errores, aces, totalServicios, ataques, bloqueosEfectivos, bloqueosFallidos
Auxiliar (Hijo de Jugador)	atributo	int	ataques	private	N/A
	atributo	int	bloqueosEfectivos	private	N/A
	atributo	int	bloqueosFallidos	private	N/A
	método	Auxiliar	Auxiliar	public	nombre, pais, errores, aces, totalServicios, ataques, bloqueosEfectivos, bloqueosFallidos

- ¿Cómo proveerá de valores iniciales a sus objetos? ¿Qué valores iniciales les asignará?
 - Crear un ArrayList inicial de Jugadores, que tendrá dentro los objetos hijos.
 - Los objetos se crearán cuando se termine de pasar los valores necesarios.

Repositorio de Github:

https://github.com/CarEsteban/Ejercicio5_POO

Diseño de CSV:

A	B	C	D	E	F	G	H	I	J	K	L
Nombre	Pais	Tipo	Errores	Aces	Total Servicios	Recibos Efectivos	Pases	Fintas Efectivas	Ataques	Bloqueos Efectivos	Bloqueos Fallidos
Nico	Guatemala	Libero	10	1	10	9	0	0	0	0	0
Esteban	Guatemala	Pasador	5	8	8	0	10	8	0	0	0

Diagrama de clases:

