

EXAMEN PARCIAL 2

Leé con cuidado el enunciado y por lo menos dos veces para resolver lo pedido. Pensá bien la estrategia de resolución antes de comenzar el desarrollo de lo que te solicitan. El objetivo de este examen es **evaluar la correcta aplicación de los conceptos y técnicas** vistos hasta el momento:

- Correcta implementación de constructores.
- Modularización reutilizable y mantenible con uso de métodos con correcta parametrización y correcto encapsulamiento, publicando *setters* y *getters* sólo cuando corresponda.
- Manejo de clases, enumerados y colecciones.
- Importación y Exportación de proyectos Java desde Eclipse.

Antes de comenzar creá un proyecto JAVA cuyo nombre cumpla con el siguiente formato:

THP-2P-SEDE-CURSO-APELLIDO-Nombre

Ejemplos:

THP-2P-BE-11A-GONZALEZ-Benito

THP-2P-YA-11A-HARA-Jaime

Este proyecto deberá ser exportado al entregar con este nombre. Tené en cuenta que esto también forma parte del examen y se califica.

NOTA IMPORTANTE:

- No te quedes con dudas, preguntá lo que haga falta.
- Al entregar el examen asegurate de finalizar la entrega. y confirmá la recepción con tu profesor.
- **No se aceptarán exámenes por fuera de la página de entrega oficial de la plataforma.**

Enunciado

El club Sportivo Vecinal nos solicita desarrollar un programa para registrar a todos los socios junto con su edad. Sabemos que no hay socios con el mismo nombre.

Además cada socio tiene amigos. Si A es amigo de B, entonces también B es amigo de A. Una persona no puede ser amiga de sí misma.

El programa debe:

- Permitir que se agreguen socios verificando que no estén registrados previamente.
- Permitir que se establezca amistad entre dos socios verificando que no se haya registrado previamente.
- Permitir que se deshaga una amistad previamente establecida.

Considerar:

- Las amistades **siempre son recíprocas**.
- Los socios deben sí o sí tener nombre y una edad lógica (si no será ERROR). No hay dos socios con el mismo nombre.
- Definir cada método con su modificador de acceso (private/public) correspondiente, creando los métodos auxiliares necesarios en la clase que considere.
- Cargar todos los datos desde una clase **Test** contemplando los casos posibles de error (son los que se muestran en la ejecución).
- Crear un Enum llamado **Resultado** que codifique todos los casos distintos de resultado posible tales como OK, YA_EXISTE, NOMBRES_IGUALES, etc. para los métodos de agregar socios, establecer amistad y deshacer amistad.

Se desea que el programa pueda:

EXAMEN PARCIAL 2

- Devolver una colección con el nombre y la edad de los amigos de uno de los socios a partir de su nombre (se recibe como parámetro). Si no existe ningún socio con ese nombre devolver *null*.
- Generar por pantalla un listado de todos los socios junto con el promedio de edad de sus amigos.

Para probar el programa desarrollé la clase Test agregando los casos de prueba que se ven a continuación. No hace falta que cargues los socios a mano. Es deseable que se haga la validación de los datos resultantes (chequear si dio lo esperado) pero no es obligatorio.

El resultado por pantalla debería ser similar al siguiente:

```
agregarSocio (Juan, 23): OK
agregarSocio (Laura, 18): OK
agregarSocio (Maria, 26): OK
agregarSocio (Joaquin, 20): OK
agregarSocio (Belen, 20): OK
agregarSocio (Daniel, 21): OK
agregarSocio (Valeria, 20): OK
agregarSocio (Ana, 19): OK
agregarSocio (Javier, 22): OK
agregarSocio (Javier, 22): YA_EXISTE

establecerAmistad (Juan/Ana): OK
establecerAmistad (Laura/Ana): OK
establecerAmistad (Laura/Juan): OK
establecerAmistad (Maria/Ana): OK
establecerAmistad (Belen/Joaquin): OK
establecerAmistad (Belen/Daniel): OK
establecerAmistad (Valeria/Ana): OK
establecerAmistad (Ana/Daniel): OK
establecerAmistad (Javier/Juan): OK
establecerAmistad (Javier/Ana): OK
establecerAmistad (Ana/Javier): YA_SON_AMIGOS
establecerAmistad (Ana/Ana): NOMBRES_IGUALES
establecerAmistad (Laura/Marcelo): NO_EXISTE
establecerAmistad (Toribio/Ana): NO_EXISTE

deshacerAmistad (Ana/Javier): OK
deshacerAmistad (Ana/Valeria): OK
deshacerAmistad (Ana/Ana): NOMBRES_IGUALES
deshacerAmistad (Laura/Marcelo): NO_EXISTE
deshacerAmistad (Toribio/Ana): NO_EXISTE

Socios con edad promedio de sus amigos:
Juan - 19.666666666666668
Laura - 21.0
Maria - 19.0
Joaquin - 20.0
Belen - 20.5
Daniel - 19.5
Valeria - no tiene amigos
Ana - 22.0
Javier - 23.0
```

EXAMEN PARCIAL 2

```
Amigos de Laura
Amigo [nombre=Ana, edad=19]
Amigo [nombre=Juan, edad=23]

Amigos de Walter
No hay nadie que se llame Walter
```

Cuando termines y/o antes de que expire el tiempo del examen exportá el proyecto tal como se pide al principio. Luego subí el archivo al Aula Virtual como entrega del examen. Asegurate de haber adjuntado lo correcto y finalizá el examen.