

Crear un proyecto java y al finalizar el examen, exportarlo con el nombre:

THP-2PS-11X-ApellidoNombre.zip

donde la S se reemplaza por la sede y la X se reemplaza por la letra del curso por ejemplo:

T1-2PY-11A-MessiLionel.zip para 1º1ºA de la sede Yatay.

Subir el archivo en el aula virtual en el link correspondiente al Parcial.

Enunciado

Nos piden crear un programa para una aplicación de citas TindOrt, en el cual donde debemos encontrar coincidencias en una lista de personas. El funcionamiento es el siguiente:

Las personas se anotan indicando ciertos datos (algunos personales y otros con los criterios de búsqueda de su preferencia).

Luego, cada persona puede ver una lista con todos los perfiles que coinciden con su criterio de búsqueda y donde, además, ella misma cumple con los criterios de búsqueda del otro (coincidencia recíproca).

De las personas sabemos:

- **Nombre**(String - **NO SE PUEDE REPETIR**)
- **Mail** (String - **No se puede repetir**).
- **Género** (Femenino, Masculino)
- **Edad** (entero)
- **Género deseado** (Femenino, Masculino, Indistinto)
- **Edad mínima deseada (desde)** (entero)
- **Edad máxima deseada (hasta)** (entero)

Se pide desarrollar en la clase que corresponda:

- Un método que permita registrar personas evitando duplicados.
- Un método privado **verificarCoincidenciaParcial (...)** que reciba dos personas y devuelva true si la persona 2 cumple con los requisitos de la persona 1.
 - Es decir que solo devuelve true cuando la persona2 cumple con lo requerido por la persona1.
- Un método privado **verificarCoincidenciaTotal (...)** que, dadas dos personas, indique si ambas cumplen los criterios de búsqueda de la otra. Devolviendo true o false según corresponda
- Un método publico **mostrarListado(...)** que, dada la dirección de email de un usuario ya registrado, devuelva un listado de aquellas personas que se ajusten completamente a sus criterios de búsqueda. En el caso de que el email no pertenezca a ningún usuario, informarlo como error y devolver null.

En todos los casos, desarrollar los métodos auxiliares que consideres necesario.

Los métodos de carga y mostrarListado deben invocarse desde el método main() de la clase Test.

Lote de pruebas

- Cargar: "Juan", "juan@mail.com", Masculino,30,Femenino,25,35
- Cargar: "Carlos", "carlos@mail.com", Masculino,45,Femenino,30,50
- Cargar: "Carlos", "carlos@mail.com", Masculino,45,Femenino,25,35.
- Cargar: "Maria", "maria@mail.com", Femenino,28,Masculino,30,40
- Cargar: "Ana". "ana@mail.com", Femenino, 32, Masculino,28,42
- Cargar: "Eva". "eva@mail.com", Femenino, 32, Masculino,40,50



- mostrarListado("juan@mail.com")
 - Muestra los datos de Maria y de Ana
- mostrarListado("carlos@mail.com")
 - Muestra los datos de Ana

Nótese que, si bien Eva también cumple con el criterio de búsqueda de ambos, ellos no están dentro del rango de edad que ella eligió. Por lo tanto, Eva no aparecerá en el listado de ninguno pues la coincidencia no es mutua.

Se deben probar todos los métodos desde una clase Test, sin pedir ingresos por teclado.

Las condiciones necesarias para aprobar el examen con la nota máxima son:

- El programa debe compilar.
- Cumplir con el enunciado.
- Correcto uso de constructores.
- Correcto diseño de clases
- Uso de los ciclos que corresponda.
- Uso correcto de parámetros.
- Usar correctamente los métodos getters y setters, y sólo cuando sean necesarios.
- Lote de prueba de todos los métodos en una clase Test
- Respetar las convenciones de programación establecidas: nombres de variables, constantes, comentarios, etc.
- Entregar en el formato correcto (exportación de proyecto, sin carpeta /bin).

