

Colecciones - Trabajo Práctico 1

Colecciones

Guía de Ejercicios

Ejercicio 1.

Implementará la clase **Grupo**, la cual representa al grupo de Trabajos Prácticos y sus integrantes. Para eso, la clase tendrá un atributo *nombre* (String) para guardar el nombre del grupo y otro atributo llamado *integrantes* (colección de Strings), donde cada elemento será el nombre de un integrante de tu grupo de trabajos prácticos. Debe cumplir con el siguiente diseño (explotar/implementar los métodos que se mencionan a continuación):

- **public Grupo(String nombre)** Es el constructor. Debe inicializar correctamente los atributos de la clase.
- **public String getNombre()** Devuelve el nombre del grupo.
- **private void setNombre(String nombre)** Setea el nombre del grupo a partir del parámetro recibido.
- **public int getCantidadIntegrantes()** Devuelve la cantidad actual de integrantes del grupo.
- **public void agregarIntegrante(String nombreIntegrante)** Agrega al grupo el nombre de un nuevo integrante, siempre y cuando éste no haya sido cargado con anterioridad. Cuando ya exista, no debe volver a agregarlo.
- **private int obtenerPosicionIntegrante(String nombreIntegrante)** Devuelve la posición en el grupo del integrante cuyo nombre se ha recibido por parámetro. En caso de no encontrarlo devolverá -1.
- **public String obtenerIntegrante(int posicion)** Recibe un valor entero que representa la posición del integrante en la lista (entero igual o mayor que 1) y devuelve el integrante guardado en (posicion - 1) dentro de la colección de integrantes. Si el integrante no existe se debe devolver *null*.
- **public String buscarIntegrante(String nombre)** Busca al integrante del grupo cuyo nombre coincida con el parámetro de entrada. En caso de que haya coincidencia, lo devuelve; caso contrario, el método devolverá *null*.
- **public String removerIntegrante(String nombreIntegrante)** Remueve de la colección de integrantes a aquel cuyo nombre coincide con el nombre recibido como parámetro. Si lo encontró, luego de removerlo debe devolverlo con *return*, y si no debe devolver *null*.
- **private void mostrarIntegrantes()** Muestra por consola la cantidad de integrantes y los nombres de cada uno de ellos.
- **public void mostrar()** Muestra por consola el nombre del grupo, la cantidad de integrantes y el nombre de cada uno de ellos.
- **public void vaciar()** Elimina todos los elementos de la lista de integrantes.

Desde el *main()*:

- Creá una instancia de Grupo asignándole el número, la letra o el nombre que le hayan puesto.

Página 1 de 4

Instituto de Tecnología ORT Carrera de
Analista de Sistemas Materia: Taller de
Programación 1

Colecciones y Estructuras Estáticas - Trabajo Práctico

- Agregale los miembros reales de tu grupo de trabajos prácticos. Agregale uno más, ficticio. Cada vez que agregues un integrante el programa debe mostrar el mensaje “[nombreIntegrante] fue asignado al grupo [nombreGrupo]”, donde [nombreIntegrante] y [nombreGrupo] son, respectivamente, los nombres del nuevo integrante y del grupo.
- Escribí un método del programa principal que compruebe si un integrante determinado fue o no agregado al grupo. Probalo tanto con uno que exista como con uno que no.
- Mostrá todos los datos del grupo.
- Remové al integrante ficticio para que el grupo quede con los integrantes reales. Luego de removerlo de la lista, mostrá los datos del integrante removido.
- Volvé a remover el mismo integrante ficticio.
- Mostrá de nuevo los datos del grupo, actualizados.
- Vacía el grupo y volvé a mostrarlo.

Ejercicio 2.

Para los Juegos Olímpicos de la Juventud nos pidieron un programa con el cual podamos calcular quién fue el ganador de una carrera. Para eso ingresaremos un atleta y los segundos que ha empleado (con decimales) para recorrer la distancia de una especialidad determinada. Se debe tener en cuenta que puede haber más de un ganador (sus tiempos empleados fueron los mismos).

Ejercicio 3.

Modificá el ejercicio anterior para que en vez de indicar solamente quién fue el ganador obtenga cuál fue la terna ganadora. Debe tener en cuenta que puede haber atletas que “empaten” en cualquiera de las tres posiciones del podio.