

## **Recuperación tercer trimestre**

### **Ejercicio 1**

Escribir un programa para gestionar conjuntos de alumnos apuntados a equipos deportivos. El programa dispondrá de:

- La Clase Alumno, con los atributos: nombre y DNI de tipo String. Implementar el método constructor y los métodos toString() y equals.
- La Clase Equipo. Cada equipo deportivo se representa en una clase que contiene el nombre del equipo y un conjunto de alumnos.

El equipo tiene operaciones para:

1. Añadir un alumno (recibe como parámetro el objeto alumno a insertar). Si el alumno ya existe en el equipo debe saltar una excepción.
2. Borrar un alumno (recibe como parámetro el objeto alumno a borrar). Si el alumno no existe en el equipo debe saltar una excepción.
3. Saber si un alumno pertenece al equipo. Recibe como parámetro el objeto alumno a buscar y devuelve null si no lo encuentra y el objeto alumno si existe.
4. Mostrar en pantalla la lista de personas del equipo.
5. Unión de equipos. El método lo llamará un equipo y se le pasará por parámetro el otro equipo con el que se quiere unir. Devuelve un equipo formado por los alumnos de ambos equipos.
6. Intersección de equipos. Idem al anterior pero devuelve un equipo formado por los alumnos que están en los dos equipos

Se pide:

1. Decidir la estructura de almacenamiento más indicada para este ejercicio. No debe establecerse ningún límite de jugadores en los equipos.
2. Realizar las clases Alumno, Equipo así como un programa de prueba que cree varios equipos y pruebe todas las operaciones mostrando en pantalla los resultados.

Modificar la clase anterior para que Equipo sea una clase genérica y se pueda aplicar a cualquier tipo. Realizar dos programas principales, uno que pruebe la clase Equipo con la clase Alumno y otro que lo pruebe con la clase Integer.

### **Ejercicio 2**

Realizar un programa que permita almacenar un historial de páginas web consultadas por un usuario. De cada página web se almacenará su url y la fecha y hora en la que se visitó.

Se debe realizar un menú de este tipo

1. Nueva página consultada: Se solicitará el nombre de la página y se tomará la fecha y hora del sistema insertándola en el historial. No se permitirá introducir una fecha y hora anterior a la última almacenada.
2. Consultar historial completo.
3. Consultar historial de un día.
4. Borrar historial completo.
5. Salir.

Se pide:

1. Decidir la estructura de almacenamiento más indicada para este ejercicio. No debe establecerse ningún límite de entradas en el historial.
2. Implementar la clase PaginaWeb, así como la clase Historial y el programa Principal.

NOTA: Utilizar la clase LocalDateTime para la fecha-hora

### **Ejercicio 3**

Se desea desarrollar una aplicación que permita enviar mensajes entre los profesores y los alumnos de un centro de enseñanza.

Se debe definir una clase abstracta Persona, así como Profesor y Alumno. Se debe incluir las opciones para que las personas puedan enviarse mensajes. De cada mensaje se guardará el remitente, el texto del mensaje y la fecha y hora en la que se envió.

Todas las personas deben tener:

- Un método para poder enviar un mensaje a otra persona. Recibirá como parámetro la persona destinataria y el texto del mensaje. Si la persona que envía el mensaje es un alumno y es menor de edad sólo puede enviar mensajes a profesores, es decir, si un alumno menor de edad intenta enviar un mensaje a otro alumno se debe provocar un error.
- Un método para poder leer los mensajes del buzón. Este método devolverá un String con todos los mensajes que tiene. Si no tiene mensajes para leer saltará el correspondiente mensaje de error. Cada mensaje irá precedido de un número 1 en adelante, es decir se mostrará con este formato.  
"Mensaje X: De: nombreRemitente Texto: textoMensaje  
Fecha y hora: DD-MM-AAAA HH:MM"
- Un método para poder leer los mensajes del buzón pero ordenado por el nombre del remitente alfabéticamente.
- Un método para poder borrar un mensaje del buzón. Al método se le pasará el número de mensaje que se desea borrar. Si no existe ese número de mensaje saltará una excepción.
- Un método que realice una búsqueda para poder encontrar los mensajes de su buzón que contienen una frase Este método devolverá un String con todos los mensajes que tienen esa frase o saltará la excepción si no encuentra ningún mensaje con esa frase.

Probar estos métodos desde el main.

### **Ejercicio 4**

Realizar una clase que implemente un diccionario. Cada palabra puede tener varios significados. Se presentará un menú de esta forma

1. Añadir palabra. Se solicitará a palabra y su significado. Si la palabra ya tenía un significado se guardará este nuevo significado con los anteriores
2. Buscar palabra en diccionario: Se solicitará la palabra y se mostrarán todos sus significados
3. Borrar una palabra del diccionario: Se solicitará la palabra y se borrará, con todos sus significados.
4. Listado de palabras que empiecen por ... Se solicitará una cadena y se mostrará un listado de palabras ordenado alfabéticamente que comiencen por esa cadena.
5. Salir NOTA: No debe considerarse ningún límite máximo de palabras.

Decide el contenedor más adecuado teniendo en cuenta que la opción 2 será la más utilizada.

### **Ejercicio 5**

Realiza el mismo ejercicio con Mapas.