

UD06.EXAME Práctico - REC

DAM1-Programación 2024-25

21/05/2025

1. Programación genérica (3)	2
Music Collab (7)	3
Clase Cancion (3)	3
Clase Musico (1)	4
Clase AppMusicCollab (3)	4

- Puedes utilizar apuntes y materiales que consideres pero deberás realizar los programas individualmente. En caso contrario se retirará el examen.
- **Durante el examen apaga y coloca tu teléfono móvil encima de la CPU de tu equipo.**
- Realiza programas bien estructurados, legibles, con comentarios, líneas en blanco, identificadores adecuados, etc.
- Cuida la interacción con el usuario, presentando la información de forma clara y ordenada.
- Utiliza los casos de prueba de ejemplo para probar tu programa y definir la salida por pantalla.

- Crea un **paquete** de nombre **ud6.xxxexamenrec** donde agrupar los ficheros de código fuente, donde **xxx** son las iniciales de tu nombre y apellidos. Si es necesario crea un nuevo proyecto/carpeta Java.
- Crea dentro del paquete los paquetes y ficheros **.java** por cada ejercicio con el nombre apropiado.
- **Incluye al inicio de cada fichero de código fuente un comentario con tu nombre y apellidos.**
- **Entrega la carpeta xxxexamenrec comprimida con los archivos de código fuente .java.**
- **Tiempo estimado:** 2:20 horas

1. Programación genérica (3)

Paquete: genericos

Crea el paquete `genericos` y dentro de él una clase `UtilGenerico.java` con los siguientes métodos genéricos:

1. **(1) Método genérico `contarCoincidencias()`** que reciba dos colecciones de elementos y devuelva el número de elementos únicos (no repetidos) que existen en ambas colecciones.
2. **(2) Método genérico `invertirOrden()`** que recibe un `Map<K, V>` y devuelve un nuevo `Map<V, List<K>>` donde:
 - a. Las claves son los valores del mapa sin repetir.
 - b. Los valores son listas de claves originales asociadas a ese valor.

Ejemplo: Mapa original: {1="a", 2="b", 3="a"} Resultado: {"a"=[1, 3], "b"=[2]}.

Crea un pequeño programa principal en la propia clase `UtilGenerico` que muestre ejemplos de uso de los métodos creados.

Music Collab (7)

Paquete: musicollab

Se quiere desarrollar una aplicación para conectar músicos que quieran colaborar en versiones de canciones. Cada músico domina uno o varios instrumentos (incluyendo "voz" para cantar), y cada canción tiene un autor, un estilo musical y una lista de instrumentos requeridos.

La aplicación debe permitir:

1. Buscar músicos capaces de tocar una canción específica. (que dominen al menos un instrumento requerido y la tengan en su repertorio).
2. Listar canciones que un grupo de músicos puede interpretar juntos (cubriendo todos los instrumentos necesarios y que todos las tengan en su repertorio.).
3. La Voz es un instrumento especial. Un músico puede cantar y tocar otro instrumento simultáneamente en una misma canción. (Ej: un guitarrista que también canta puede tocar guitarra + voz).

Clase Cancion (3)

Implementa la clase según los requisitos y completa el código para que el programa principal genere la siguiente salida por consola.

Atributos:

- nombre (String).
- autor (String).
- estiloMusical (String).
- instrumentosRequeridos (List<String>, sin repetir).

Método:

- `instrumentosDistintos()`: método estático que devuelve una lista de los distintos instrumentos únicos (no repetidos) y ordenados alfabéticamente, que requiere una colección de canciones que se pasa como parámetro de entrada.

Otros Requisitos:

1. Orden natural: Por nombre alfabético.
2. Método `toString()`:
 - Devuelve "Nombre" (Autor) - [Estilo] | Instrumentos: [lista].

```

Canciones ordenadas por nombre:
=====

"Billie Jean" (Michael Jackson) - Pop | Instrumentos: voz, bajo, batería
"Bohemian Rhapsody" (Queen) - Rock | Instrumentos: voz, piano, guitarra, batería
"Hotel California" (Eagles) - Rock | Instrumentos: voz, guitarra, bajo, batería
"Imagine" (John Lennon) - Balada | Instrumentos: voz, piano
"Rolling in the Deep" (Adele) - Pop | Instrumentos: voz, piano, batería
"Smells Like Teen Spirit" (Nirvana) - Grunge | Instrumentos: voz, guitarra, bajo, batería
"Sweet Child O'Mine" (Guns N' Roses) - Rock | Instrumentos: voz, guitarra, bajo, batería
"Yesterday" (The Beatles) - Pop | Instrumentos: voz, guitarra

Canciones ordenadas por estilo musical, luego por autor y luego por nombre:
=====

"Imagine" (John Lennon) - Balada | Instrumentos: voz, piano
"Smells Like Teen Spirit" (Nirvana) - Grunge | Instrumentos: voz, guitarra, bajo, batería
"Rolling in the Deep" (Adele) - Pop | Instrumentos: voz, piano, batería
"Billie Jean" (Michael Jackson) - Pop | Instrumentos: voz, bajo, batería
"Yesterday" (The Beatles) - Pop | Instrumentos: voz, guitarra
"Hotel California" (Eagles) - Rock | Instrumentos: voz, guitarra, bajo, batería
"Sweet Child O'Mine" (Guns N' Roses) - Rock | Instrumentos: voz, guitarra, bajo, batería
"Bohemian Rhapsody" (Queen) - Rock | Instrumentos: voz, piano, guitarra, batería

Lista de instrumentos distintos:
=====

[bajo, batería, guitarra, piano, voz]

```

Clase Musico (1)

Implementa la clase.

Atributos:

- nombre (String).
- alias (String, único).
- Instrumentos que domina (List<String>, sin repetidos).
- Repertorio: canciones que conoce y podría tocar (por gusto o experiencia).

Otros Requisitos:

1. Orden natural: Por alias alfabético.
2. Método toString():
 - Muestra [Alias] (Nombre): [lista de instrumentos].

Clase AppMusicCollab (3)

Completa los métodos requeridos para que el programa principal genere la salida por consola.

Métodos:

1. `List<Musico> musicosParaCancion(Cancion c, List<Musico> musicos)`
Devuelve una lista de músicos que dominan al menos uno de los instrumentos requeridos por la canción `c`, ordenados por alias alfabético.
2. `List<Cancion> cancionesPosibles(List<Musico> grupo, List<Cancion> canciones)`
Devuelve las canciones de la lista en las que el grupo de músicos puede cubrir todos los instrumentos requeridos, ordenadas por nombre alfabético.
 - a. Un músico no tiene por qué tocar en una canción si sus instrumentos no son requeridos.
 - b. Un mismo músico solo puede tocar un instrumento a la vez (excepto "voz", que puede combinarse con otros).

Músicos para 'Bohemian Rhapsody':

=====

[BassQueen] (Ana López): bajo, voz | Repertorio: 3 canciones

[GuitarHero] (Carlos Ruiz): guitarra, voz | Repertorio: 3 canciones

[MultiInstrumentalist] (Elena Díaz): guitarra, piano, voz | Repertorio: 3 canciones

[PianoMaster] (Pablo Martín): piano | Repertorio: 3 canciones

Canciones posibles para el grupo:

=====

"Billie Jean" (Michael Jackson) - Pop | Instrumentos: voz, bajo, batería

"Hotel California" (Eagles) - Rock | Instrumentos: voz, guitarra, bajo, batería

"Smells Like Teen Spirit" (Nirvana) - Grunge | Instrumentos: voz, guitarra, bajo, batería

"Sweet Child O'Mine" (Guns N' Roses) - Rock | Instrumentos: voz, guitarra, bajo, batería

"Yesterday" (The Beatles) - Pop | Instrumentos: voz, guitarra