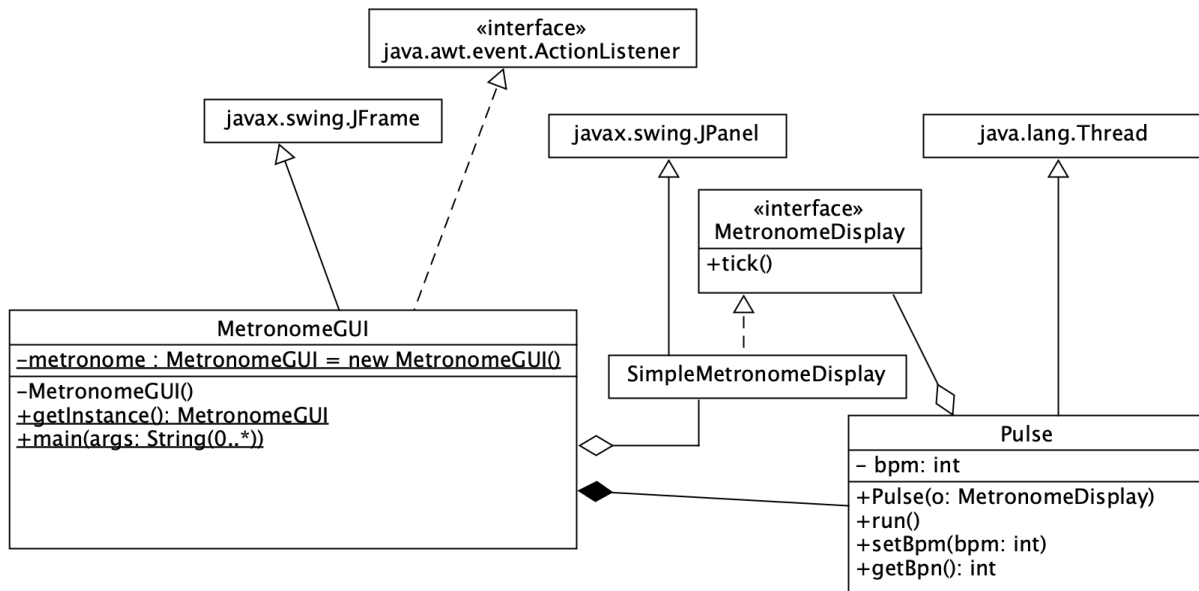


INFO090 - Tarea - INFO090-17 2022 Semestre 2

Un metrónomo es un dispositivo ampliamente usado por músicos que marca con precisión una frecuencia o *pulso* determinado. El pulso del metrónomo se establece en BPM (Beats Per Minute, latidos por minutos). Por ejemplo, en un pulso de 60 bpm cada *beat demora* un segundo. Como es una medida de frecuencia, valores más altos de BPM equivalen a intervalos menores, es decir, una frecuencia más rápida. Por ejemplo, en un bpm de 120, los beats ocurren cada 0,5 segundos. En esta tarea construiremos un metrónomo en Java.

Diseño general

El siguiente diagrama representa el diseño general de la aplicación.

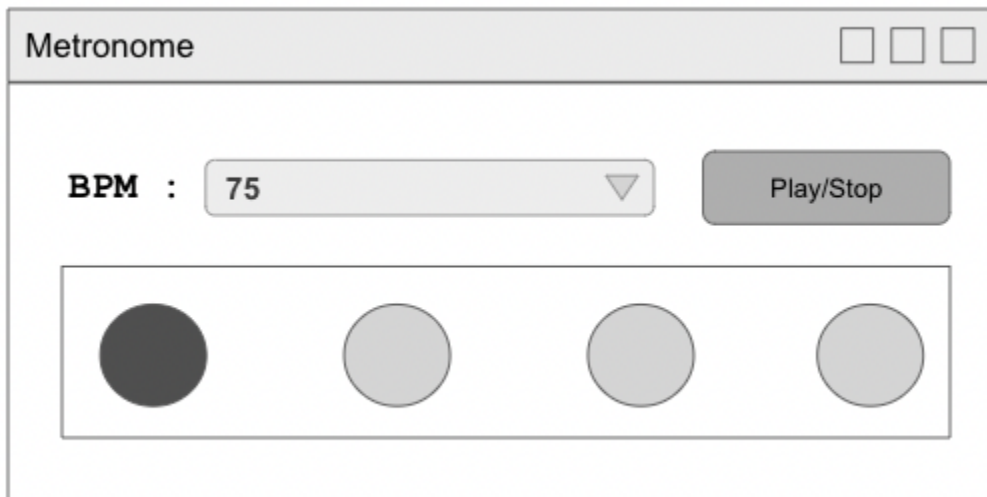


La clase `MetronomeGUI` es Singleton, representa la interfaz gráfica, y es también el programa (tiene un método static `main`). Contiene un panel especial de la clase `SimpleMetronomeDisplay` donde se representan los beats del metrónomo gráficamente y sonoramente. Contiene también un objeto de `Pulse`, un hilo de ejecución (`Thread`) que permite "llevar el pulso" en la frecuencia especificada (en bpm). El hilo está en ejecución todo el tiempo, y en su método sobrescrito `run()` itera estableciendo pausas de acuerdo con el bpm establecido. De esta forma cada 60/bpm segundos invoca al método `tick` de un objeto `MetronomeDisplay`. Para lograr esto, al construir el thread `Pulse`, `MetronomeGUI` pasa como parámetro el `SimpleMetronomeDisplay`. De esta

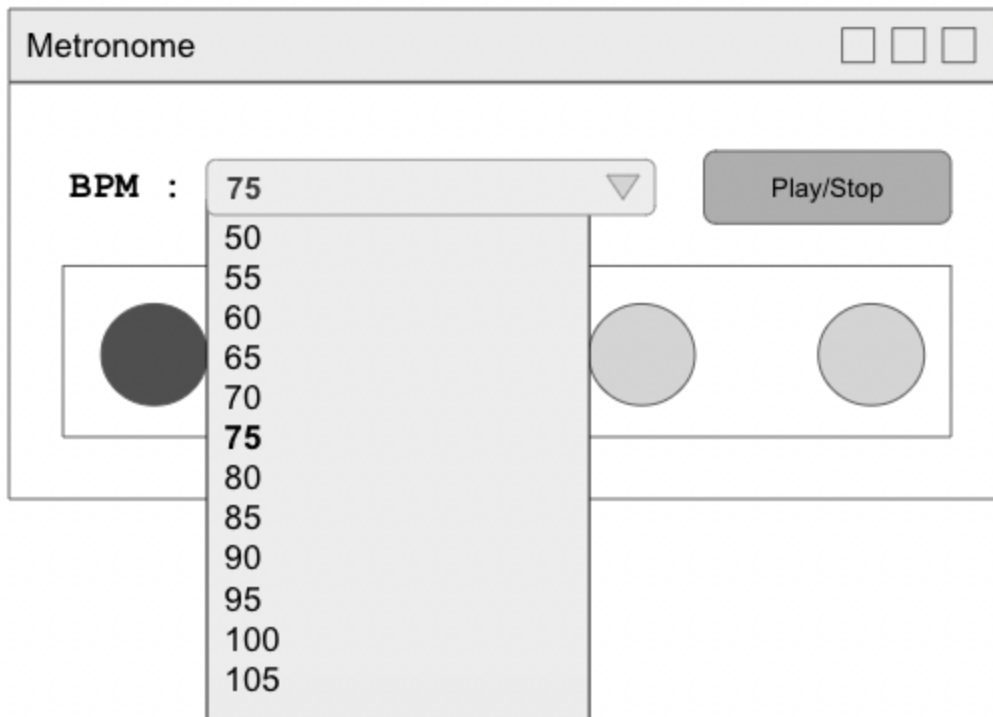
forma el objeto Pulse puede controlar el display invocando al método tick() en cada *beat*.

Interfaz gráfica

La interfaz gráfica es libre y la puedes diseñar como estimes conveniente, siempre y cuando esté contenida en la clase MetronomeGUI y SimpleMetronomeDisplay. Acá una idea simple de interfaz gráfica.



El drop-down-list muestra distintas frecuencias de menor a mayor para escoger:



La zona de abajo de la interfaz contiene el SimpleMetronomeDisplay con 4 círculos. En cada beat el siguiente círculo oscurece su background y luego vuelve al color de fondo normal. De esta forma muestra el pulso gráficamente.

Como puedes ver en las imágenes, la interfaz del metrónomo tiene varias limitaciones:

- Sólo permite un compás de 4 tiempos (4/4)
- Sólo permite escoger ciertas frecuencias.

Puedes hacer mejoras en este diseño para superar las limitaciones.

Sonido

Un buen metrónomo debe poder emitir un sonido en cada beat. Tu deberás investigar cómo generar un sonido en java e incluir esta funcionalidad en el panel SimpleMetronomeDisplay.

Aspectos formales

- Mantener indentación!!!
- Respetar convenciones de nombre

- Para variables/atributos/métodos que no estén especificados, nombrarlos de forma apropiada!
- Borrar código basura

Documentación con JavaDoc

- Cada clase construida debe llevar comentarios en formato JavaDoc. Incluir un comentario al comenzar la clase, y uno en cada método (excepto getters y setters)
- Para cada clase incluir descripción de la clase y campo @author
- Para cada método incluir descripción de qué hace el método y campos @param (para cada parámetro del método, sin @param si el método no tiene parámetros) y @return (excepto si el método es void)
- Video explicativo de JavaDoc (clase grabada):
<https://drive.google.com/file/d/1o36JJGNhLD2ODRix4Nwf1dHZp7ePIKBn/view?usp=sharing>
- Generar la documentación JavaDoc desde eclipse (también en el video ;)

Entrega

- Organizar las clases en el paquete cl.uach.info090.metronome
- Entregar el proyecto eclipse comprimido en ZIP en **siveducmd** ("Tarea")
- Se realizará una revisión de avance de la interfaz gráfica del metrónomo el ~~jueves 15~~ martes 20 de Diciembre en horario de clases (11:30 am)
- La fecha de entrega final es el 6 de Enero de 2023 antes de medianoche.