Media Player

Steps

1. Vamos de más fácil a más difícil ¿Qué creen que es lo más fácil?
2. Partimos solamente con contratos, definimos el “lenguaje” que hablará nuestra aplicación
    1. `trait Player` y `trait Media`
    2. ¿Qué es común para todos los Player? ¿Qué tienen distinto? => Mismo contrato, distinta implementación (trait + clases concretas)
        1. `def play(m: Media): Unit`
        2. `def stop(m: Media): Unit`
        3. `def seekTo(m: Media, ms: Int): Unit`
    3. ¿Qué es común para todos los Media? ¿Qué tienen distinto? => Video es Audio++
        1. Una opción sería decir que un Video hereda de Audio ¿Qué opinan de eso?
        2. Comencemos por traits separados para Video y Audio
        3. Ahora, definimos el contrato común en Media y Audio/Video los heredan
            1. `val title: String`
            2. `val duration: Int // in milliseconds`
            3. `def play(): Unit`
            4. `def stop(): Unit`
            5. `def seekTo(ms: Int): Unit`
        4. Para Video tenemos 3 opciones para representar la resolución:
            1. 2 variables (menos reusable)
            2. tuplas (funcional, pero poco descriptivo)
            3. clase especializada Resolution (descriptivo y más fácil de extender)
                1. refreshRate
                2. `gcd` : Algoritmo de Euclides (introducir Utility)
    4. Ahora implementemos los players
        1. Comencemos por Simple
        2. Luego Verbose
        3. ¿Hay duplicación de código? ¿Y si quiero crear otro tipo de player que realice acciones “al rededor” de play/stop/etc?
        4. Verbose → Basic
    5. Media
        1. Audio

        1. Necesita current position, definir var en el trait es problemático => Mejor definirlo en una clase concreta
        2. Implementar AudioTrack
    2. Video
        1. Implementar VideoTrack
        2. currentPosition se repite 😉
    3. AbstractMedia
        1. stop repite código!
6. Seek
    1. Buena idea partir por clamp
    2. Seek solo hace clamp
7. Local vs Streaming
    1. Ya tenemos local, falta streaming
    2. trait BufferedMedia (opcional)
    3. Abstract Buffered Media con buffered (de nuevo, var puede generar complicaciones en un trait)
    4. currentPosition = Utility.clamp(ms, 0, currentPosition + buffered)