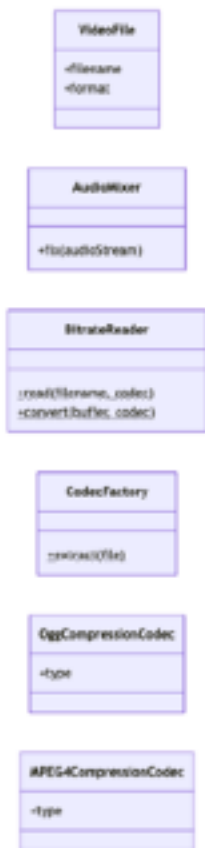- We'll implement the **Facade Pattern** to simplify interacting with a complex video processing system. The facade class offers an easy interface for converting videos, hiding the subsystem's multiple classes.

Step 1: Define the Subsystem Classes

VideoFile
- +filename
- +format

AudioMixer
- +fix(audioStream)

BitrateReader
- :read(filename, codec)
- +convert(buffer, codec)

CodecFactory
- :extract(file)

OggCompressionCodec
- +type

MPEG4CompressionCodec
- +type

- First, let's define the classes of our video processing subsystem. These classes (`AudioMixer`, `VideoFile`, `BitrateReader`, `CodecFactory`, `OggCompressionCodec`, and `MPEG4CompressionCodec`) will each have specific roles, but for simplicity, they'll have basic, hypothetical methods.