Java | Coursework 1 | Question 3 | 1838838 | Marley Sudbury

I gave every class or sub-class at least one extra method or attribute, otherwise it wouldn't have any purpose. I gave all instruments three basic attributes: *ID* (String), *Name* (String) and *Possible_Notes* (String[]).

In Java, constructors cannot be inherited so each class calls the super() function. All methods print directly to the terminal rather than returning the value.

In the following examples I will be using the Name attribute to refer to the players name, but it could just as easily be used to name the instruments. These examples can all be run from Concert.class.

Justification of class methods:

- Instrument
    - Play(int index)—All instruments have an array of notes that they can play, this method takes an index and displays that the instrument has played the note at that position in the array, e.g. *Jack (Violin): A#5*
- Instrument/Brass
    - Bend(int start, int end)—By a variety of techniques, brass instruments can bend the note they are playing up or down. This simplified algorithm assumes that it will bend at least one notes worth, and displays the results.
      E.g. *John (Trumpet): C4->C#4->Db4*
- Instrument/Brass/Trumpet
    - Click()—By pressing the valves of a trumpet without blowing, the player can create a clicking noise. E.g. *Stanford (Trumpet): CLICK*
- Instrument/Brass/Tuba
    - Bang()—A loud bang noise can be made by striking a tuba with either a hand or the mouthpiece. E.g. *Alex (Tuba): BANG*
- Instrument/Percussion
    - Beat(int beats)—A very common purpose of percussion instruments is to keep time by beating a steady rhythm. This method represents this by playing the same not many times, E.g. *David (Drum): DUN DUN DUN DUN DUN DUN DUN*
- Instrument/Percussion/Drum
    - Roll(int n)—Drum rolls are often used to build suspense before an announcement. E.g. *Jim (Drum): Brrrrrrrrrrrrrrrrrrrrrrrrr-tish* (the number of 'r's is n)
- Instrument/Percussion/Triangle
    - Roll(int n)—This is more similar to the beat of the drum than it is to the drum's roll. This is where you quickly strike all internal sides of the triangle in a circular motion. E.g. *James (Triangle): TING->TING->TING->TING->TING* (the arrows symbolise that it is a continuous sound rather than a steady beat)
- Instrument/String (In the code this is referred to as StringIns, to avoid conflict with the Java class String)
    - Pizzicato(int index)—This is a technique whereby a stringed instrument is plucked to create a sort of soft, muted note. It works by passing an integer which is the index of the note to be played in the array of notes. E.g. *Carol (Violin): !A#4* (The '!' is used to signify that it is being played pizzicato rather than regular note)
- Instrument/String/Cello
    - Drone(int index, int n)—This technique provides a long drone of the same note, e.g. *Victoria (Cello): Bb3---------->*

- Instrument/String/Violin
    - Glissando(int start, int end)—This is a technique where the player smoothly moves their finger up or down a string to change the pitch being played. Method is very similar to Brass.Bend(). E.g. *Clarissa (Violin): Eb4->E4->F4*
- Instrument/Woodwind
    - Multiphonic(int index1, int index2)—Various techniques can allow for a player to create two separate tones simultaneously. E.g. *Ariel (Flute): Bb5 & E4*
- Instrument/Woodwind/Clarinet
    - Slap_Tongue(int index)—A slapping noise made by violently releasing the tongue. It still creates a pitch but it is very different to the regular noise. E.g. *Henrietta (Clarinet): !F#3*
- Instrument/Woodwind/Flute
    - Whistle()—A high-pitched whistle noise made by blowing a small amount of air into a flute. E.g. *Abigail (Flute): WHISTLE*