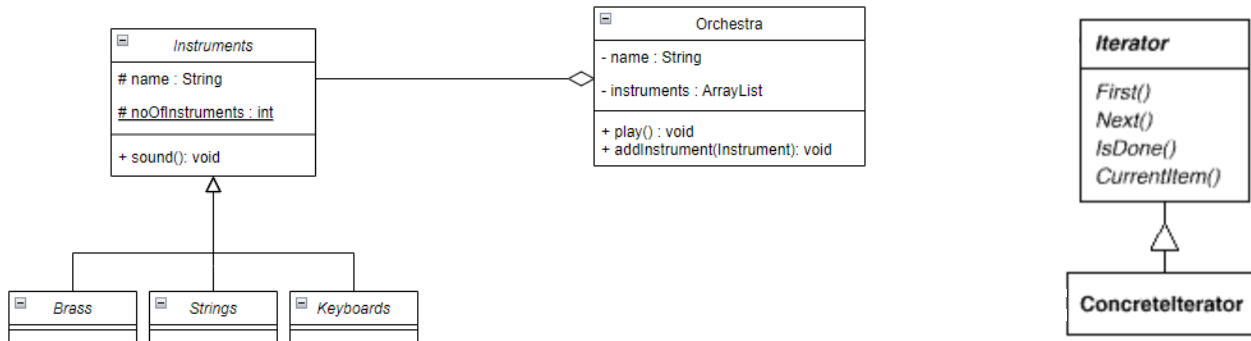


Lab Work 3 Iterator Pattern

Goal: Practice on Iterator Pattern.

1. For this lab, your task is to add an iterator to your Instrument-Orchestra example from last week's lab 2.



2. Please check the slides from Week 3 and examine Iterator Pattern Diagram. Then decide which one of your classes will be the Aggregate class and which one will be the Item from the diagram.

3. Implement your **AbstractIterator** class. Your concrete iterator should be **OrchestralIterator** that starts with the first element and iterates one instrument at a time. Implement the methods accordingly. Do not forget to add necessary methods to your aggregate class also.

You may get help from Iterator Pattern example code from Week 3.

4. Write a function called `"play2(OrchestralIterator)"`. Use your **OrchestralIterator** to replicate the Orchestra class' `play()` method but it will traverse in reverse order. This function will be called instead of Orchestra class's `play()` method.

You may get from the `printAggregate` method from the example code from Week 3.

5. Then in your `main()` after your code from before, use your newly implemented `"play2(OrchestralIterator)"` function instead to traverse the instruments in the orchestra.

6. Create a new concrete iterator: **BrassIterator**. This iterator must traverse only "Brass" instruments in the Orchestra. Implement the class accordingly.

Hint: `"createIterator"` method returns **OrchestralIterator** but now you need to have a second Iterator. Therefore, take a parameter to indicate which iterator you want to get.

7. Use `"play2(AbstractIterator)"` one last time with **BrassIterator** object as a parameter to test your new iterator.