code readability :in class `StringPermutations` the code is easy to follow, logically. Each line in function `stringPermutations` does a specific job. The classes name and their functions and variables are named meaningfully (naming convention), so that they communicate intent. So that the code and its structure are clearly visible. The code also commented as needed.

For Example:

Class names:

`SwapArrayCharacters`, `ReturnSwapedCharIndexToOne`, `PrintString` , `ConvertCharactersArrayToString`,……..

Function names:

`printResultedString()`, `isLessSecondeSwapedCharIndex()`, `isLessWeightIndex()`, `increasingWeightIndex_SwapedCharIndexToOne()`, `returnToInitialValue()` , `zeroWeightIndex_IncreasingSwapedCharIndex()`……..

`variables names:`

`_increasingValueByOne`, `_returnSwapedCharIndexToOne`, `_swappingCharacters` , `_printResultedString`

low coupling and high cohesion: the class is built following the single responsibility principle, Interfaces (`CheckerMin`, `Convertor`, `IncreasingValue`, `PrintResultedString`,…..), and Façade , classes (`StringPermutations`, , `ModifiedIndexValue` , `CombinString`)

Error handling: handle the error if the input string does not have a length, that is empty in class `CombinString`,