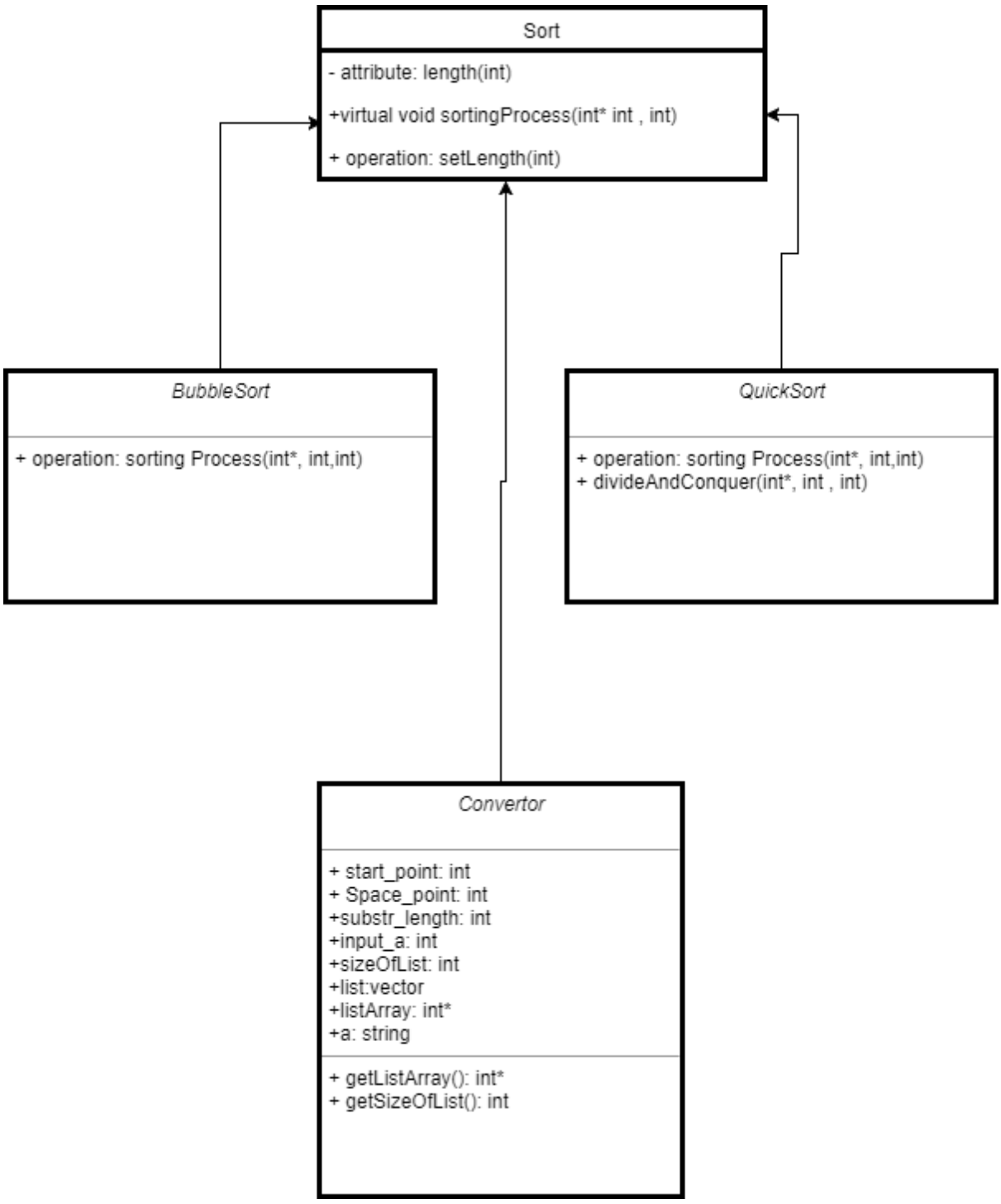
Class Diagram



Sort

Attritube:

Length(int) // initialize the length of list of input value

Operation:

Void sortingProcess(int*, int, int) //the pure virtual function make the sort as an abstract class.

Void setLength(int) // set up the length using a intger type parameter.

BubbleSort:

Operation:

Void sortingProcess(int*, int, int) // Compare each pair of adjacent elements from the beginning of an array and, if they are in reversed order, swap them. If at least one swap has been done, repeat step 1.

QuickSort:

Void sortingProcess(int*, int, int) // As a requirement, we need to set the second element as a pivot to process the whole system. Firstly, we need to use the DivideAnd Conquer() to implement two aims, confirm the location of the pivot value after comparison and make those number which is less than pivot are in front of the pivot or after the pivot otherwise.

And then do it recursively.

int DivideAndConquer(int*, int, int) //this function can tell us the position we need to divide and conquer, if the start is less than the end, e need to continue the function or stop otherwise. It will return the index of the pivot in the list. And then in the recursion, we can adjust our start and end according to the return value.

Convertor:

Attribute:

(there are some update for the attribute, I delete some unuseful variable)

sizeOfList(int)// the size of the real input_value

listArray(int*)//a pointer of int for listarray

Operation:

Convertor(string input, int length)// convertthe string and cut the string into the single piece of integer. I use <sstream> to change the type of the data from string to int.

Int* getListArray()// return the listArray

Int getSizeOFListArray()// return the length of the listArray.

Testing:

Input	Description	Output
Q 1 1 1 1	When we need to use QuickSort to process a simple list without 6	False 1 1 1 1
B -343 2 0 23 12 777	When there is a negative value	False -343 0 3 12 23 777
32 -2 2 43 3 22	When we donot select the method	No output
B 34 23 1	When there is a lot of space between the method and first element	False 1 23 34
Q 2	When there is a single element	False 2
Q 6	When we use Quick Sort to process the single input 6	True 6
B 6	When we use BubbleSort to process the single input 6	True 6
Q 9 8 7 6 5 4 3 2 1	When we type a descending order with Quick Sorting	True 1 2 3 4 5 6 7 8 9
B 9 8 7 6 5 4 3 2 1	When we type a descending order with BubbleSort	True 1 2 3 4 5 6 7 8 9