Folder src

2 printable files

src/Int.java
src/Main.java

src/Int.java

```
package src;

public class Int {
    private int privateInt;

    /**
     * Constructor to initialize the Int object with a specified integer value.
     * @param newInt the integer value to initialize the Int object with.
     */
    public Int(int newInt) {
        privateInt = newInt;
    }

    /**
     * Converts the private integer field to a public integer.
     * @return the value of the private integer field.
     */
    public int toInt() {
        return privateInt;
    }

    /**
     * Sets the value of privateInt to the specified newInt.
     *
     * @param newInt the new value to set for privateInt
     */
    public void setInt(int newInt) {
        privateInt = newInt;
    }

    /**
     * Converts the private integer to its String representation.
     * This method returns the string form of the private integer.
     */
    public String toString(){
        return String.valueOf(privateInt);
    }

    /**
     * Swaps the integer value of the current object with the integer value of the specified object.
     * This method uses a temporary variable to facilitate the swap.
     */
    public void swapCurrObjValue(Int toSwap){
        int temp = toSwap.toInt();
        toSwap.setInt(this.toInt());
        this.setInt(temp);
    }

    /**
     * Swaps the elements at the specified positions in the given array.
     * This method modifies the original array.
     *
     * @param arr the array of Int objects
     * @param elem1 the index of the first element to be swapped
     * @param elem2 the index of the second element to be swapped
     */
    public static void swapArrayElem(Int[] arr, int elem1, int elem2){
        int temp = arr[elem1].toInt();
        arr[elem1].setInt(arr[elem2].toInt());
        arr[elem2].setInt(temp);
    }

    /**
     * Swaps the integer values of two Int objects.
     * This method uses a temporary variable to perform the swap.
     */
    public static void swapObjValue(Int obj1, Int obj2){
        int temp = obj1.toInt();
        obj1.setInt(obj2.toInt());
        obj2.setInt(temp);
    }

}
```

src/Main.java

```
package src;

/**
 * The Main class contains methods for converting strings to integers,
 * sorting an array of Int objects using bubble sort, and the main method
 * to execute the program.
 *
 * <p>This class demonstrates the use of custom Int objects and various
 * methods to swap their values.</p>
 *
 * <p>Usage:</p>
 * <pre>
 * java src.Main arg1 arg2 ...
 * </pre>
 *
 * <p>Where each argument is an integer.</p>
 *
 * @author Surbeck Léon, Nicolet Victor
 * @version 1.0
 */


public class Main {



    /**
     * Converts a string representation of a number to an integer.
     * Throws IllegalArgumentException for invalid input.
     * <p>
     * Note: The {@code stringToInt} method handles signed integers.
     * </p>
     *
     * @param s the string to convert
     * @return the integer value of the string
     * @throws IllegalArgumentException if the string is null, empty, or contains non-numeric characters
     */
    public static int stringToInt(String s) {
        if (s == null || s.isEmpty()) {
            throw new IllegalArgumentException("Le String ne peut pas être null ou vide");
        }

        int result = 0;
        boolean sign = s.charAt(0) == '-';
```

```java
        for(int i = sign ? 1 : 0; i < s.length(); ++i){
            char currentChar = s.charAt(i);
            if (currentChar >= '0' && currentChar <= '9') {
                result = result * 10 + (currentChar - '0');
            } else {
                throw new IllegalArgumentException("Caractère non numérique trouvé : " + currentChar);
            }
        }

        return sign ? -result : result;

    }

    /**
     * Sorts an array of Int objects using the bubble sort algorithm.
     * This method modifies the input array to be in ascending order.
     *
     * @param arr the array of Int objects to be sorted
     * @param n the number of elements in the array
     */
    public static void bubbleSort(Int[] arr, int n){
        int i, j, temp;
        boolean swapped;
        for (i = 0; i < n - 1; i++) {
            swapped = false;
            for (j = 0; j < n - i - 1; j++) {
                if (arr[j].toInt() > arr[j + 1].toInt()) {

                    // Swap arr[j] and arr[j+1] using different methods

//                    temp = arr[j];
//                    arr[j] = arr[j + 1];
//                    arr[j + 1] = temp;

//                    Int.swapArrayElem(arr, j, j+1);

//                    Int.swapObjValue(arr[j], arr[j+1]);

                    arr[j].swapCurrObjValue(arr[j+1]);

                    swapped = true;
                }
            }
            if (!swapped)
                break;
        }
    }


    public static void main(String[] args) {
        int nbArgs = args.length;
        Int[] values;
        values = new Int[nbArgs];


        // Getting the args and putting them in the Int[] array
        for(int i = 0; i < nbArgs; ++i){
            values[i] = new Int(stringToInt(args[i]));
            System.out.println(values[i].toString());
        }

        System.out.println();
        bubbleSort(values, values.length);

        for(int i = 0; i < nbArgs; ++i){
            System.out.println(values[i]);
        }


    }
}
```