Folder src

3 printable files

(file list disabled)

src\Ex1.java

```java
import java.util.Arrays;

/**
 * @author Calum Quinn
 * @author Dylan Ramos
 */
public class Ex1 {
    public static void main(String[] args) {
        if (args.length == 0) {
            System.out.println("No arguments provided.");
            return;
        }

        // So as not to modify args directly
        int[] argsValues = new int[args.length];

        // Check each number individually
        for (int i = 0; i < args.length; ++i) {
            String arg = args[i];
            int j, power = arg.length() - 1; // Power is power of 10 for first digit
            boolean isNegative = false;

            switch (arg.charAt(0)) {
                case '-':
                    isNegative = true;
                case '+':
                    j = 1; // If there is a sign, the first digit will be at index 1
                    --power; // The first digit is at index 1, so the power is one less
                    break;
                default:
                    j = 0;
            }

            int number = 0;
            // Multiply each digit of the current number by the correct power of 10
            for (; j < arg.length(); ++j) {
                number += (int) ((arg.charAt(j) - '0') * Math.pow(10, power));
                --power;
            }

            // Apply sign to negative numbers
            if (isNegative) {
                number *= -1;
            }

            // Add current number to array
            argsValues[i] = number;
        }

        // Sort and then print array
        System.out.println(Arrays.toString(bubbleSort(argsValues)));
    }

    private static int[] bubbleSort(int[] values) {
        int temp;
        boolean swapped;

        for (int i = 0; i < values.length - 1; ++i) {
            swapped = false;
```

```
            // Loop on the digits that are not yet sorted
            for (int j = 0; j < values.length - i - 1; ++j) {
                if (values[j] > values[j + 1]) {
                    // Swap the values
                    temp = values[j];
                    values[j] = values[j + 1];
                    values[j + 1] = temp;
                    swapped = true;
                }
            }
            // Breaks early if the array is already sorted
            if (!swapped) {
                break;
            }
        }

        return values;
    }
}
```

src\Ex3.java

```
/*

    A lot of code is duplicated from the Ex1.java file so that we can have both examples
    simultaneously instead of simply writing the third exercise over the first one.

*/

/**
 * @author Calum Quinn
 * @author Dylan Ramos
 */
public class Ex3 {
    public static void main(String[] args) {
        if (args.length == 0) {
            System.out.println("No arguments provided.");
            return;
        }

        // So as not to modify args directly
        Int[] argsValues = new Int[args.length];

        // Check each number individually
        for (int i = 0; i < args.length; ++i) {
            String arg = args[i];
            int j, power = arg.length() - 1; // Power is power of 10 for first digit
            boolean isNegative = false;

            switch (arg.charAt(0)) {
                case '-':
                    isNegative = true;
                case '+':
                    j = 1; // If there is a sign, the first digit will be at index 1
                    --power; // The first digit is at index 1, so the power is one less
                    break;
                default:
                    j = 0;
            }

            Int number = new Int();
            // Multiply each digit of the current number by the correct power of 10
            for (; j < arg.length(); ++j) {
                number.setNum(number.getNum() + ((arg.charAt(j) - '0') * (int) Math.pow(10, power)));
                --power;
            }

            // Apply sign for negative numbers
            if (isNegative) {
```

```
                number.setNum(-number.getNum());
            }

            // Add current number to array
            argsValues[i] = number;
        }

        // Sort and then print array
        printArray(bubbleSort(argsValues));
    }

    private static Int[] bubbleSort(Int[] values) {
        boolean swapped;

        for (int i = 0; i < values.length - 1; ++i) {
            swapped = false;
            // Loop on the digits that are not yet sorted
            for (int j = 0; j < values.length - i - 1; ++j) {
                if (values[j].getNum() > values[j + 1].getNum()) {
                    // Swap the values
                    Int.swapIntVal(values[j], values[j + 1]);
                    swapped = true;
                }
            }
            // Breaks early if the array is already sorted
            if (!swapped) {
                break;
            }
        }

        return values;
    }

    // No print function for an array of Int, so implemented here
    private static void printArray(Int[] array) {
        System.out.print("[");
        for (int i = 0; i < array.length; ++i) {
            System.out.print(array[i].toString()); // Use of overloaded toString for Int
            if (i != array.length - 1) {
                System.out.print(", ");
            }
        }
        System.out.println("]");
    }
}
```

src\Int.java

```
/**
 * @author Calum Quinn
 * @author Dylan Ramos
 */
public class Int {
    private int num;

    // Default constructor
    public Int() {
        num = 0;
    }

    // Constructor with int parameter
    public Int(int num) {
        this.num = num;
    }

    // Getter
    public int getNum() {
        return num;
    }
```

```
    // Setter
    public void setNum(int num) {
        this.num = num;
    }

    // Value to String
    public String toString() {
        return String.valueOf(num);
    }

    // Swap two Ints in an array
    public static void swapInt(Int[] array, int index1, int index2) {
        Int temp = array[index1];
        array[index1] = array[index2];
        array[index2] = temp;
    }

    // Swap the values of two Ints
    public static void swapIntVal(Int lhs, Int rhs) {
        int temp = rhs.num;
        rhs.num = lhs.num;
        lhs.num = temp;
    }

    // Swap the values of current Int and another
    public void swap(Int other) {
        swapIntVal(this, other);
    }
}
```