Esercizio sulle lambda

Prendere il mio esempio di `insertionSort` basato sulle lambda e farlo funzionare in Java (attenzione: serve Java 1.8!).

Sulla carta fare uno schema di come è strutturato e come evolve nel tempo lo stack e lo heap.

Infine usare il debugger per verificare la propria ipotesi sull'evoluzione di stack e heap.

Uso di lambda

```
package it.uniud.poo.abstractions;

import java.util.function.BiPredicate;

/**
 * Example with parametric choice of sort direction implemented with lambdas.
 */
public class MySortLambda {

    /**
     * Labels to specify order direction
     */
    public  enum SortDirection {
        INCREASING,
        DECREASING
    }


    /**
     * Run a simple example with two sorting procedures: up and down.
     */
    public static void main (String a[]){
    int[] arr1 = {10,34,2,56,7,67,88,42};

    doParametricSort(arr1, SortDirection.INCREASING);
    System.out.format("Increasing: ");
    for(int i:arr1){
        System.out.print(i);
        System.out.print(", ");
    }
    doParametricSort(arr1, SortDirection.DECREASING);
    System.out.format("\nDecreasing: ");
    for(int i:arr1){
```

```
    System.out.print(i);
    System.out.print(", ");
    }
}
/**
 * sort the array a
 * MODIFY the array a so that values are ordered
 * @param a: an array of integers to be sorted
 * @param dir: the direction of the sort: INCREASING/DECREASING
 */
private static void doParametricSort(int[] a, SortDirection dir) {
    BiPredicate<Integer, Integer> ic = null; // we need to use Integer
    switch (dir) {
    case INCREASING:
        ic = (Integer x, Integer y) -> (x < y);// first lambda
        break;

    case DECREASING:
        ic = (Integer x, Integer y) -> (x > y);// second lambda
            break;
    }

    doInsertionSort(a, ic);
}
/**
 * Do sort the array a using the direction implied by the
 * generic predicate. MODIFY array a.
 * @param a
 * @param ic: a lambda predicate of two Integers.
 */
private static void doInsertionSort(int[] a, BiPredicate<Integer, Integer> ic) {
    for (int i = 1; i < a.length; i++) {
        for (int j = i ; j > 0 ; j--){
            if (ic.test(a[j],a[j-1])){// automatic type conversion
                                      // from int to Integer
                swap(a, j);
            }
        }
    }
}
/**
 * swap a[j] with a[j-1]; MODIFY the array a
 * @param a an array of integers, REQUIRED to have 2 or more elements
 * @param j an index of the array, REQUIRED to be a valid index and > 0.
 */
private static void swap(int[] a, int j) {
```

```
            int temp;
            temp = a[j];
            a[j] = a[j-1];
            a[j-1] = temp;
        }
    }
```