Task 4: Strategy Develop a Context class that can use different SortingStrategy algorithms interchangeably to sort a collection of numbers

```
package com.Day28;
import java.util.Arrays;
interface SortingStrategy {
  int[] sort(int[] numbers);
}
class BubbleSortStrategy implements SortingStrategy {
  @Override
  public int[] sort(int[] numbers) {
    int[] sortedNumbers = numbers.clone();
    boolean swapped;
    do {
      swapped = false;
      for (int i = 0; i < sortedNumbers.length - 1; i++) {
        if (sortedNumbers[i] > sortedNumbers[i + 1]) {
          int temp = sortedNumbers[i];
          sortedNumbers[i] = sortedNumbers[i + 1];
          sortedNumbers[i + 1] = temp;
          swapped = true;
        }
      }
    } while (swapped);
    return sortedNumbers;
  }
}
class QuickSortStrategy implements SortingStrategy {
  @Override
  public int[] sort(int[] numbers) {
    int[] sortedNumbers = numbers.clone();
    quickSort(sortedNumbers, 0, sortedNumbers.length - 1);
    return sortedNumbers;
  }
  private void quickSort(int[] arr, int low, int high) {
    if (low < high) {
      int pi = partition(arr, low, high);
      quickSort(arr, low, pi - 1);
      quickSort(arr, pi + 1, high);
    }
  }
  private int partition(int[] arr, int low, int high) {
    int pivot = arr[high];
    int i = (low - 1);
    for (int j = low; j < high; j++) {
      if (arr[j] < pivot) {
        i++;
        int temp = arr[i];
        arr[i] = arr[j];
        arr[j] = temp;
      }
    }
```

```
        int temp = arr[i + 1];
        arr[i + 1] = arr[high];
        arr[high] = temp;
        return i + 1;
    }
}
class Context {
    private SortingStrategy strategy;
    public Context(SortingStrategy strategy) {
        this.strategy = strategy;
    }
    public void setStrategy(SortingStrategy strategy) {
        this.strategy = strategy;
    }
    public int[] sortNumbers(int[] numbers) {
        return strategy.sort(numbers);
    }
}
public class StatergyDevelopment {
    public static void main(String[] args) {
        int[] numbers = {4, 2, 7, 1, 9, 5};
        SortingStrategy bubbleSort = new BubbleSortStrategy();
        SortingStrategy quickSort = new QuickSortStrategy();
        Context context = new Context(bubbleSort);
        int[] sortedNumbers = context.sortNumbers(numbers);
        System.out.println("Sorted using Bubble Sort: " + Arrays.toString(sortedNumbers));
        context.setStrategy(quickSort);
        sortedNumbers = context.sortNumbers(numbers);
        System.out.println("Sorted using Quick Sort: " + Arrays.toString(sortedNumbers));
    }
}
```