Task 4: Strategy

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

```java
public interface SortingStrategy {
    void sort(int[] numbers);
}
public class BubbleSortStrategy implements SortingStrategy {
    @Override
    public void sort(int[] numbers) {
        int n = numbers.length;
        for (int i = 0; i < n - 1; i++) {
            for (int j = 0; j < n - i - 1; j++) {
                if (numbers[j] > numbers[j + 1]) {
                    int temp = numbers[j];
                    numbers[j] = numbers[j + 1];
                    numbers[j + 1] = temp;
                }
            }
        }
        System.out.println("Sorted using Bubble Sort");
    }
}
import java.util.Arrays;
public class QuickSortStrategy implements SortingStrategy {
    @Override
    public void sort(int[] numbers) {
        quickSort(numbers, 0, numbers.length - 1);
        System.out.println("Sorted using Quick Sort");
    }
    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;
    }
}
public class Context {
    private SortingStrategy sortingStrategy;
    public void setSortingStrategy(SortingStrategy sortingStrategy) {
        this.sortingStrategy = sortingStrategy;
    }
    public void performSort(int[] numbers) {
        sortingStrategy.sort(numbers);
    }
}
public class Main {
    public static void main(String[] args) {
        int[] numbers = {5, 2, 8, 1, 9, 3};
        Context context = new Context();
        context.setSortingStrategy(new BubbleSortStrategy());
        context.performSort(numbers.clone());
        System.out.println("After Bubble Sort: " + Arrays.toString(numbers));
        context.setSortingStrategy(new QuickSortStrategy());
        context.performSort(numbers.clone());
        System.out.println("After Quick Sort: " + Arrays.toString(numbers));
    }
}
```