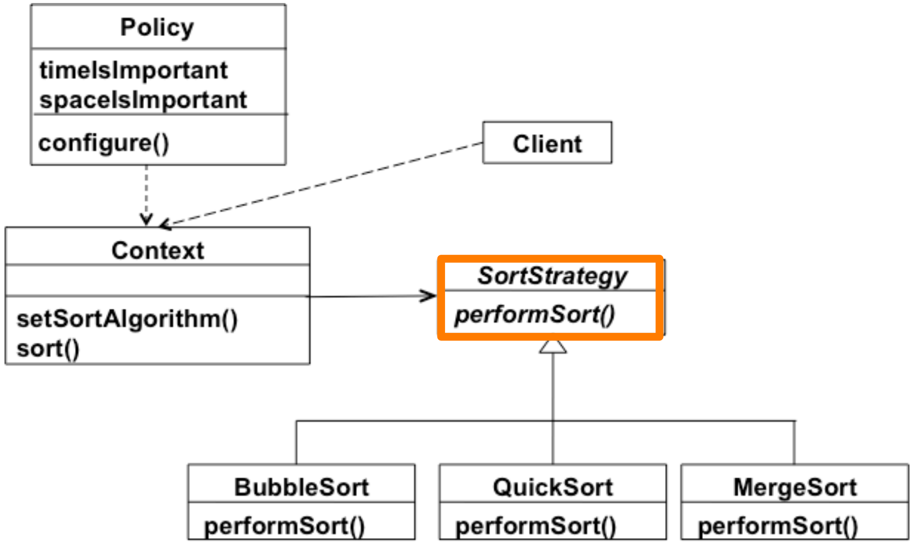
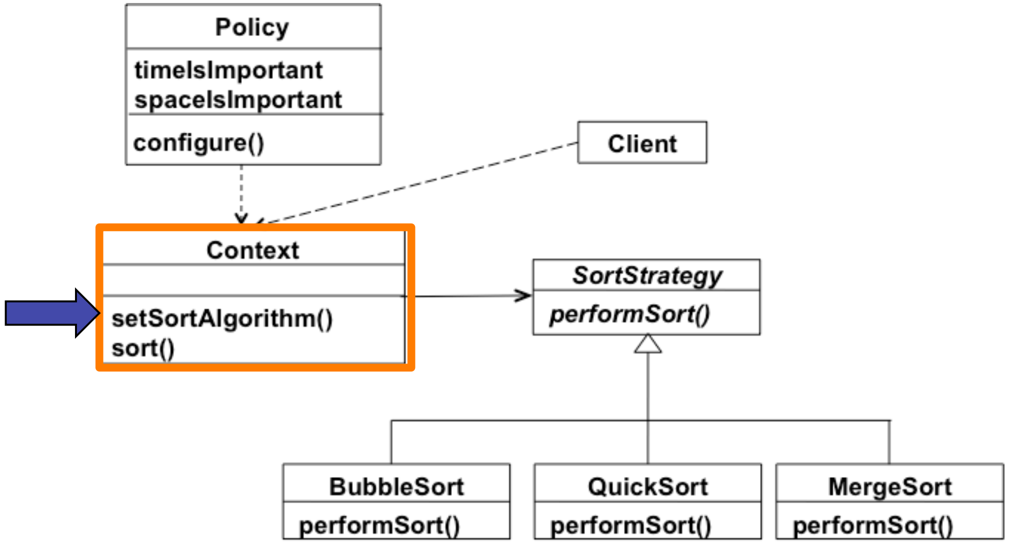
# Solution Part #1: Create Strategy Interface

```
public interface SortStrategy {
        public void performSort(int [] a);
}

public class QuickSort implements SortStrategy {…}
public class MergeSort implements SortStrategy {…}
public class BubbleSort implements SortStrategy {…}
```



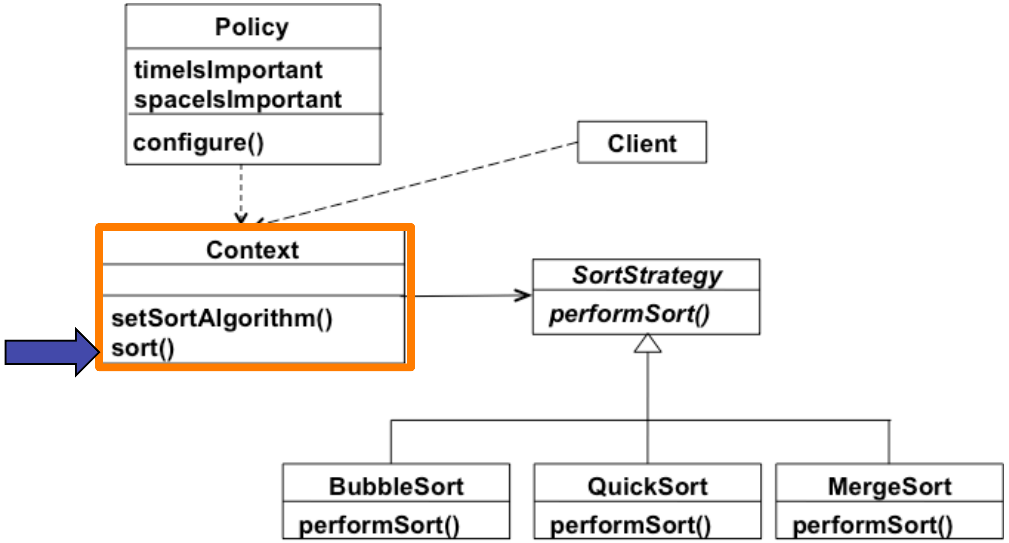
# Solution Part #2: setSortAlgorithm() method

```
public class Context {
        private SortStrategy sortAlgorithm;
        private int[] array;
…
        public void setSortAlgorithm(SortStrategy sa) {
                sortAlgorithm = sa;
        }
…
}
```



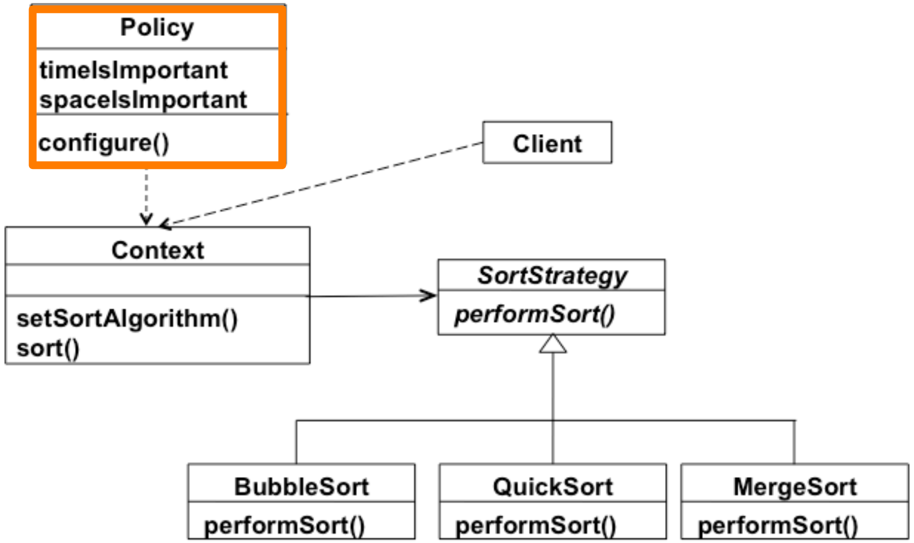
# Solution Part #2: sort() method

```
public class Context {
        private SortStrategy sortAlgorithm;
        private int[] array;
…
        public void sort() {
                sortAlgorithm.performSort(this.array);
        }
…
}
```



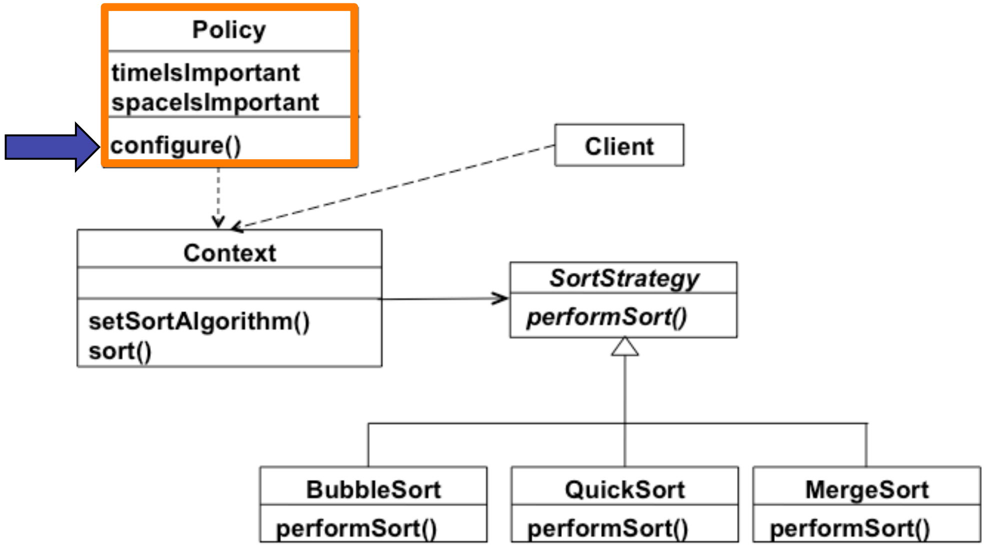
# Solution Part #3a: Create Policy

```
public class Policy {

        private boolean timeImportant;
        private boolean spaceImportant;

        private Context context;
…
}
```



# Solution Part #3b: Make Policy Configurable

```
public void configure  (boolean timeIsImportant, boolean spaceIsImportant){
   this.setTimeImportant(timeIsImportant);
   this.setSpaceImportant(spaceIsImportant);
   if (isTimeImportant() && !isSpaceImportant()) {
        System.out.println("Time is important, choosing merge sort!”);
        this.context.setSortAlgorithm(new MergeSort());
   } else if (isTimeImportant() && isSpaceImportant()) {
        System.out.println("Time and space are important, choosing quick sort!");
        this.context.setSortAlgorithm(new QuickSort());
   }
}
```



# Solution Part #4: Wire it all together

```
ArraySorter sortingContext = new ArraySorter();
Policy policy = new Policy(sortingContext);

int[] array = createIntegerArray();
for (int i = 0; i < 10; i++) {
        array = scrambleArray(array);
        sortingContext.setArray(array);
        simulateRuntimeConfigurationChange(policy);
        System.out.print("Unsorted Array a = ");
        printIntegerArray(array);
        sortingContext.sort();
        System.out.print("Sorted Array a = ");
        printIntegerArray(array);
}
```