

```
// Strategy Interface

interface SortingStrategy {

    void sort(int[] array);

}

// Concrete Strategies

class BubbleSort implements SortingStrategy {

    public void sort(int[] array) {

        // Bubble sort algorithm implementation

    }

}

class QuickSort implements SortingStrategy {

    public void sort(int[] array) {

        // Quick sort algorithm implementation

    }

}

// Context

class Sorter {

    private SortingStrategy strategy;

    public Sorter(SortingStrategy strategy) {

        this.strategy = strategy;

    }

}
```

```
public void setStrategy(SortingStrategy strategy) {  
    this.strategy = strategy;  
}
```

```
public void sort(int[] array) {  
    strategy.sort(array);  
}  
}
```

// Usage

```
public class StrategyPatternDemo {  
    public static void main(String[] args) {  
        int[] data = {5, 2, 8, 7, 1};  
  
        Sorter sorter = new Sorter(new BubbleSort());  
        sorter.sort(data); // Uses BubbleSort  
  
        sorter.setStrategy(new QuickSort());  
        sorter.sort(data); // Uses QuickSort  
    }  
}
```

Comment [A1]: