

# Strategy Pattern

Strategy gør det muligt at have en gruppe algoritmer som er udskiftelige med hinanden.

### Strategy (SortStrategy)

- Interface til alle algoritmer der skal kunne skiftes ud.

### ConcreteStrategy (QuickSort, BubbleSort)

- Implementerer sort metoden fra SortStrategy.
- Indeholder algoritmen som laver alt arbejdet.

### Context (NumberList)

- Indeholder det data der skal manipuleres.
- Bliver konfigureret med et objekt der har et Strategy interface.

```
abstract class SortStrategy
{
    public abstract void Sort(List<int> list);
}
```

Strategy gør det muligt at have en gruppe algoritmer som er udskiftelige med hinanden.

### Strategy (SortStrategy)

- Interface til alle algoritmer der skal kunne skiftes ud.

### ConcreteStrategy (QuickSort, BubbleSort)

- Implementerer sort metoden fra SortStrategy.
- Indeholder algoritmen som laver alt arbejdet.

### Context (NumberList)

- Indeholder det data der skal manipuleres.
- Bliver konfigureret med et objekt der har et Strategy interface.

```
class QuickSort : SortStrategy
{
    public override void Sort(List<int> list)
    {
        list.Sort();
        Console.WriteLine("QuickSort");
    }
}

class BubbleSort : SortStrategy
{
    public override void Sort(List<int> list)
    {
        int temp = 0;
        for (int x = 0; x < list.Count; x++)
            for (int s = 0; s < list.Count - 1; s++)
                if (list[s] > list[s + 1])
                {
                    temp = list[s + 1];
                    list[s + 1] = list[s];
                    list[s] = temp;
                }

        Console.WriteLine("BubbleSort");
    }
}
```

Strategy gør det muligt at have en gruppe algoritmer som er udskiftelige med hinanden.

### Strategy (SortStrategy)

- Interface til alle algoritmer der skal kunne skiftes ud.

### ConcreteStrategy (QuickSort, BubbleSort)

- Implementerer sort metoden fra SortStrategy.
- Indeholder algoritmen som laver alt arbejdet.

### Context (NumberList)

- Indeholder det data der skal manipuleres.
- Bliver konfigureret med et objekt der har et Strategy interface.

```
class NumberList
{
    private List<int> list = new List<int>();
    private SortStrategy strategy;

    public void SetStrategy(SortStrategy strategy)
    {
        this.strategy = strategy;
    }

    public void Add(int number)
    {
        list.Add(number);
    }

    public void Sort()
    {
        strategy.Sort(list);

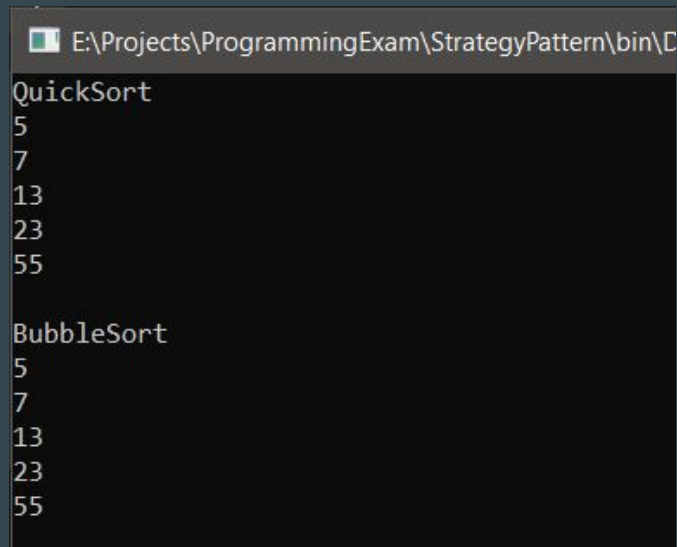
        foreach (int number in list)
        {
            Console.WriteLine(number);
        }
        Console.WriteLine();
    }
}
```

```
static void Main(string[] args)
{
    NumberList numbers = new NumberList();
    numbers.Add(13);
    numbers.Add(23);
    numbers.Add(7);
    numbers.Add(5);
    numbers.Add(55);

    numbers.SetStrategy(new QuickSort());
    numbers.Sort();

    numbers.SetStrategy(new BubbleSort());
    numbers.Sort();

    Console.ReadLine();
}
```



E:\Projects\ProgrammingExam\StrategyPattern\bin\Debug\...

QuickSort

5  
7  
13  
23  
55

BubbleSort

5  
7  
13  
23  
55

