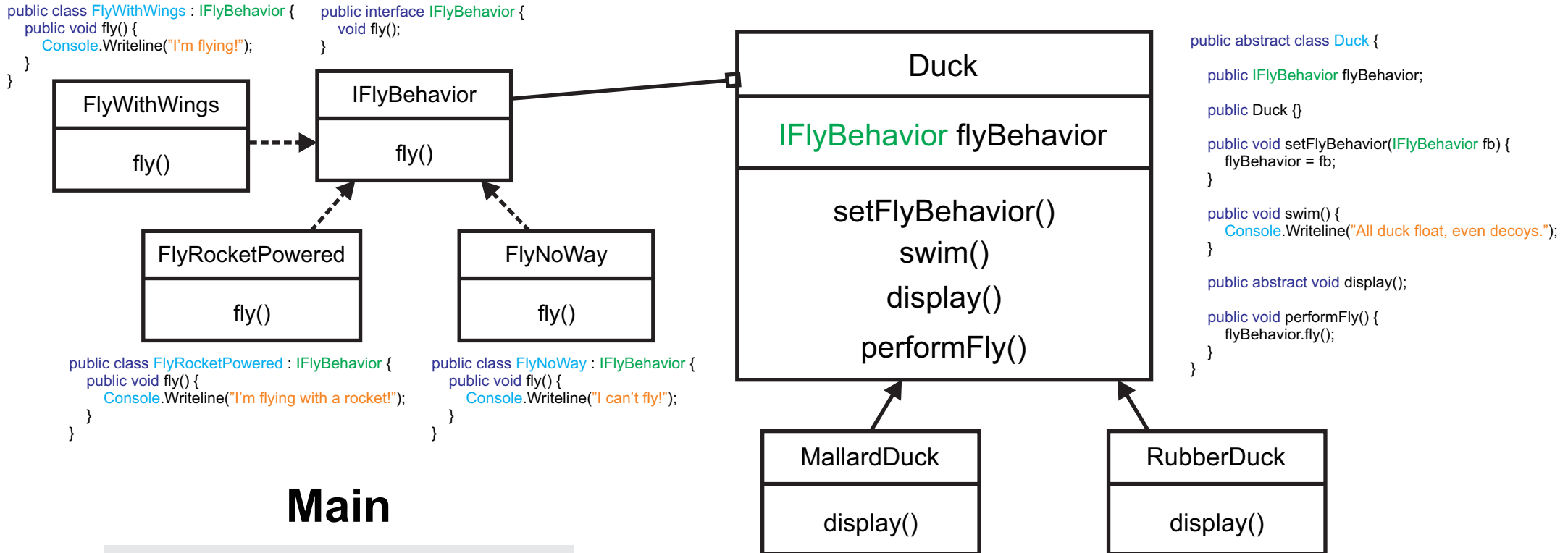


Strategy Pattern



Main

```
class Program {
    static void main(string[] args) {
        MallardDuck mallard = new MallardDuck();
        RubberDuck rubberDuckie = new RubberDuck();

        mallard.performFly();
        rubberDuckie.performFly();
        rubberDuckie.setFlyBehavior(new FlyRocketPowered());
        rubberDuckie.performFly();
    }
}
```

Output

```
I'm flying!
I can't fly!
I'm flying with a rocket!
```

Observer Pattern

Main

```
class Program
{
    static void Main(string[] args)
    {
        WeatherData weatherData = new WeatherData();

        CurrentConditionsDisplay currentDisplay
        = new CurrentConditionsDisplay(weatherData);

        weatherData.setMeasurements(80, 65, 30.4f);
        weatherData.setMeasurements(82, 70, 29.2f);
        weatherData.setMeasurements(78, 90, 27.7f);
    }
}
```

Output

Current conditions: 80 F degrees,
Humidity : 65 % and
Pressure: 30,4 P.

Current conditions: 82 F degrees,
Humidity : 70 % and
Pressure: 29,2 P.

Current conditions: 78 F degrees,
Humidity : 90 % and
Pressure: 27,7 P.

```
public class WeatherData : ISubject
{
    private List<IObserver> observers;
    private float temperature;
    private float humidity;
    private float pressure;

    public WeatherData()
    {
        observers = new List<IObserver>();
    }

    public void registerObserver(IObserver o)
    {
        observers.Add(o);
    }

    public void removeObserver(IObserver o)
    {
        int i = observers.IndexOf(o);
        if (i >= 0)
        {
            observers.RemoveAt(i);
        }
    }

    public void notifyObservers()
    {
        for (int i = 0; i < observers.Count(); i++)
        {
            IObserver observer = (IObserver)observers[i];
            observer.update(temperature, humidity, pressure);
        }
    }

    public void measurementsChanged()
    {
        notifyObservers();
    }

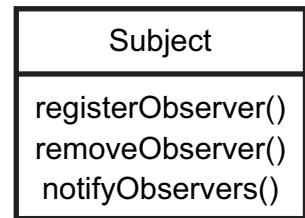
    public void setMeasurements(float temperature, float humidity, float pressure)
    {
        this.temperature = temperature;
        this.humidity = humidity;
        this.pressure = pressure;
        measurementsChanged();
    }

    public float getTemperature()
    {
        return temperature;
    }

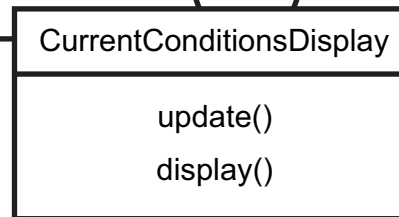
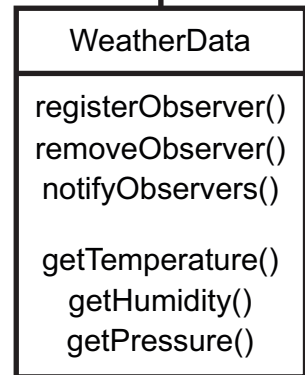
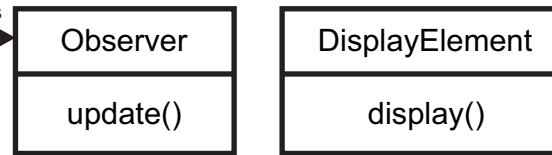
    public float getHumidity()
    {
        return humidity;
    }

    public float getPressure()
    {
        return pressure;
    }
}
```

```
public interface ISubject
{
    void registerObserver(IObserver o);
    void removeObserver(IObserver o);
    void notifyObservers();
}
```



```
public interface IObserver
{
    void update(float temperature, float humidity, float pressure);
}
```



```
class CurrentConditionsDisplay : IObserver, IDisplayElement
{
    private float temperature;
    private float humidity;
    private float pressure;
    private ISubject weatherData;
```

```
    public CurrentConditionsDisplay(ISubject weatherData)
    {
        this.weatherData = weatherData;
        weatherData.registerObserver(this);
    }
```

```
    public void update(float temperature, float humidity, float pressure)
    {
        this.temperature = temperature;
        this.humidity = humidity;
        this.pressure = pressure;
        display();
    }
```

```
    public void display()
    {
        Console.WriteLine("Current conditions: " + temperature
            + " F degrees, Humidity : " + humidity
            + " % and Pressure: " + pressure + " P.");
    }
}
```