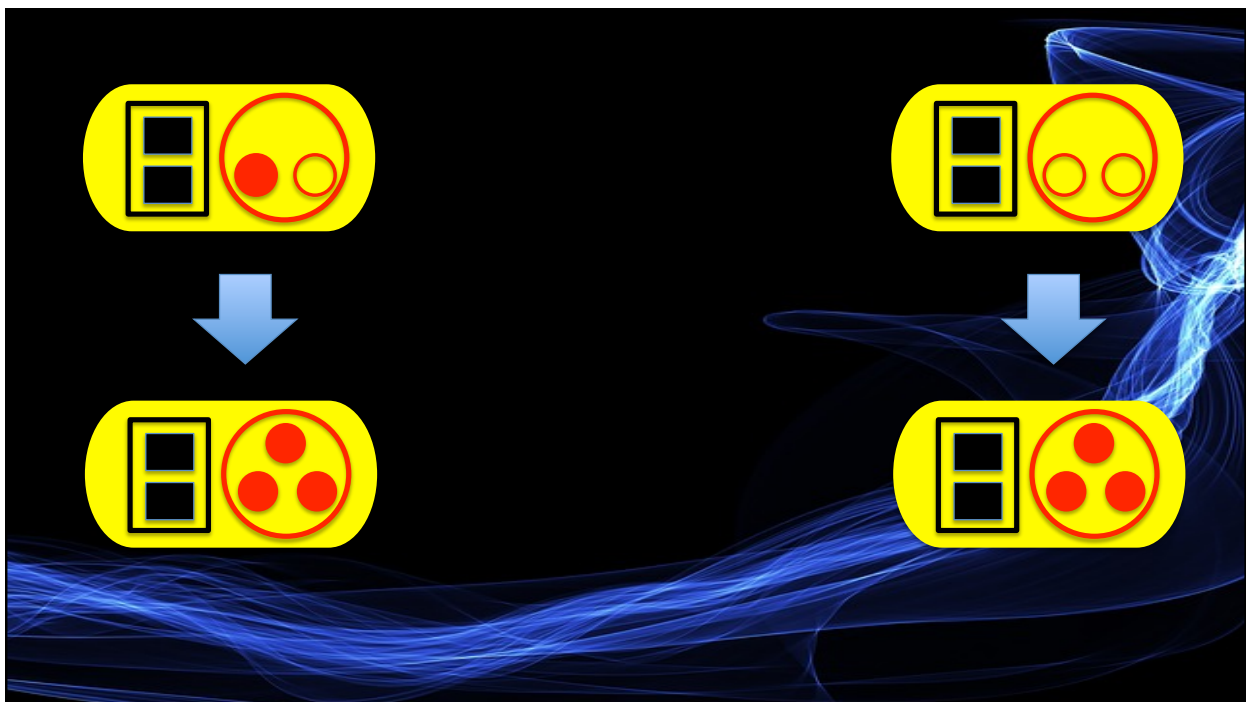


INTERFACES



```
public abstract class Vehicle{  
    private String color;  
    public Vehicle(String c) {color=c;}  
    public          void moveForward(){...}  
    public          void moveBackward(){...}  
    public abstract String toString();  
}
```



Source: Pixabay

```
public abstract class Vehicle{  
  
    public abstract void moveForward();  
    public abstract void moveBackward();  
    public abstract String toString();  
}
```



Source: Pixabay

```
public interface VehicleIF {
```

```
    public void moveForward();  
    public void moveBackward();  
    public String toString();  
}
```



Source: Pixabay

```
public interface VehicleIF {
```

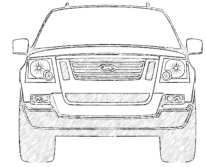
```
    void moveForward();  
    void moveBackward();  
    String toString();  
}
```



Source: Pixabay

```
public interface VehicleIF {
```

```
    void moveForward();  
    void moveBackward();  
    String toString();  
    static void goHome(){...}  
}
```



Source: Pixabay

```
public interface VehicleIF {
```

```
    int SOS_PHONE = 112;
```

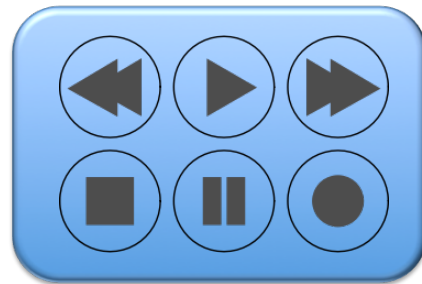
```
    void moveForward();  
    void moveBackward();  
    String toString();  
}
```



Source: Pixabay

Interface

- Design abstraction
- Services exposed to user
- Details hidden
- One interface, many implementations



Source: Pixabay

Summary

- All methods abstract (and public)
except static and default methods
- Attributes: static, final (and public)

