

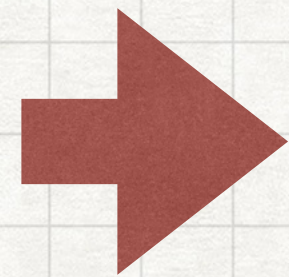
ISP

INTERFACE

SEGREGATION PRINCIPLE



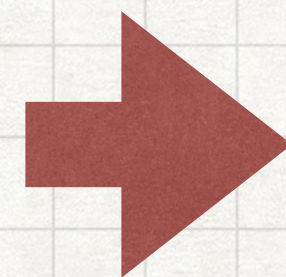
CLIENT



```
public class Communicator {  
    public void sendMessage (String message) {...}  
    public void sendMessage (Object object) {...}  
}
```



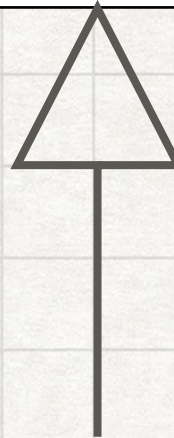
CLIENT



```
enum Level {  
    NONE, WEAK_ENCRYPTION, STRONG_ENCRYPTION  
}  
  
public class Communicator {  
    public void setEncryption (Level level) {}  
    public void sendMessage (String message) {}  
    public void sendMessage (Object object) {}  
}
```



```
enum Encryption {  
    NONE, WEAK_ENCRYPTION, STRONG_ENCRYPTION  
}  
  
public interface Communicator {  
  
    public void setEncryption (Encryption level);  
  
    public void sendMessage (String message);  
  
    public void sendMessage (Object object);  
}
```



```
public class CommunicatorImpl implements Communicator {  
  
    ...  
}
```



# *The Interface Segregation Principle*

“

Clients should not be forced to depend  
on methods they do not use

— *Robert C. Martin*

”



```
public interface Communicator {  
    public void sendMessage (String message);  
    public void sendMessage (Object object);  
}
```

```
public interface Encrypted {  
    enum Level {  
        NONE, WEAK_ENCRYPTION, STRONG_ENCRYPTION }  
    public void setEncryption (Level level);  
}
```

```
public class CommunicatorImpl implements Communicator, Encrypted {  
    public void setEncryption (Level level) {}  
    public void sendMessage (String message) {}  
    public void sendMessage (Object object) {}  
}
```

```
graph BT; CImpl[CommunicatorImpl] --|> C[Communicator]; CImpl --|> E[Encrypted];
```



ISP

INTERFACE

SEGREGATION PRINCIPLE