SOLID principles

S	Single responsibility principle
O	Open/closed principle
L	Liskov substitution principle
I	Interface segregation principle
D	Dependency inversion principle

SOLID principles

S — Single responsibility principle
O — Open/closed principle
L — Liskov substitution principle
I — Interface segregation principle
D — Dependency inversion principle
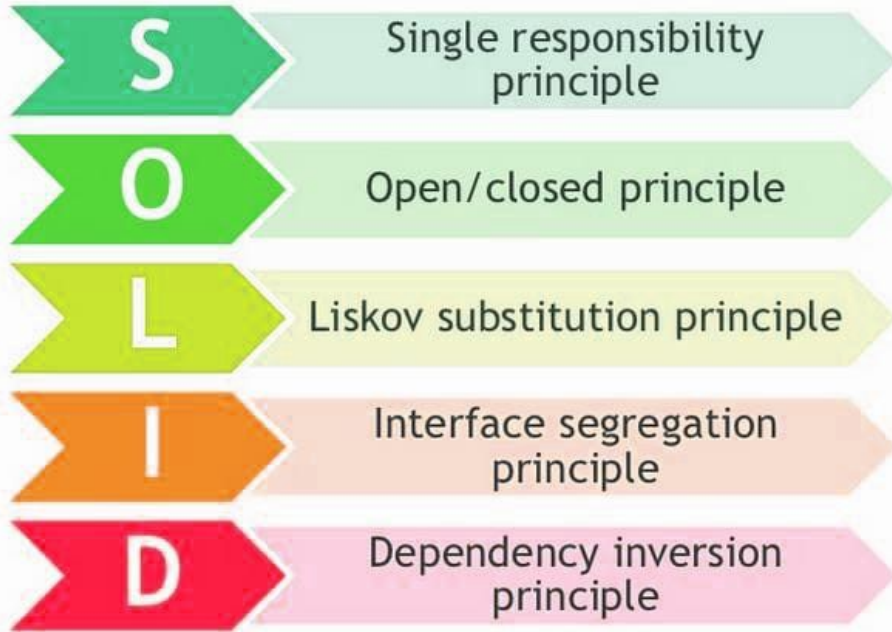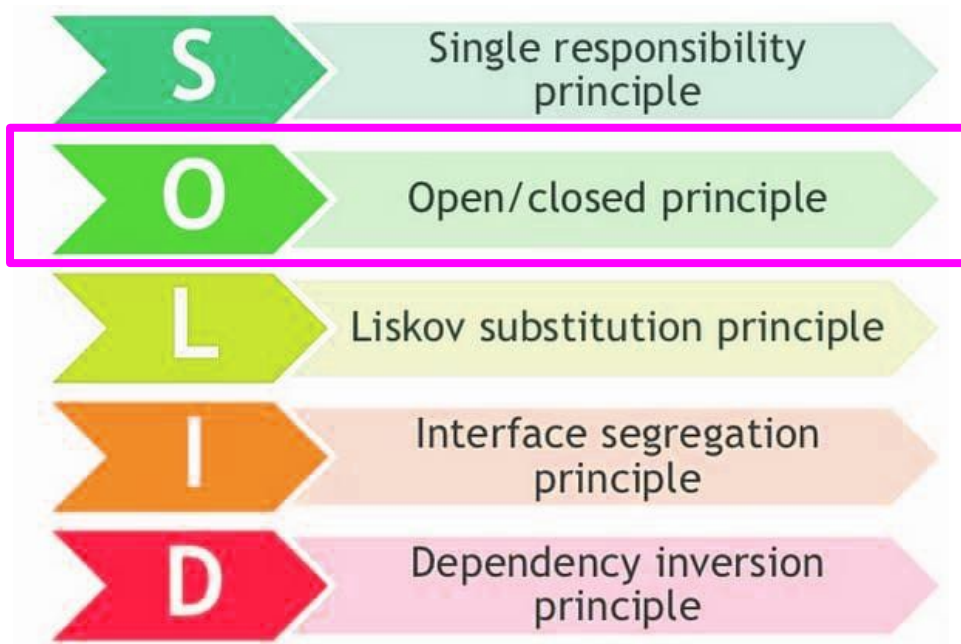
Software entities (classes, modules, methods, etc.) should be **open** for extension, but **closed** for modification

OCP violation problems

- Classes are tightly coupled
  - Changing one bit of code affects multiple entities
- Code is difficult to test in isolation

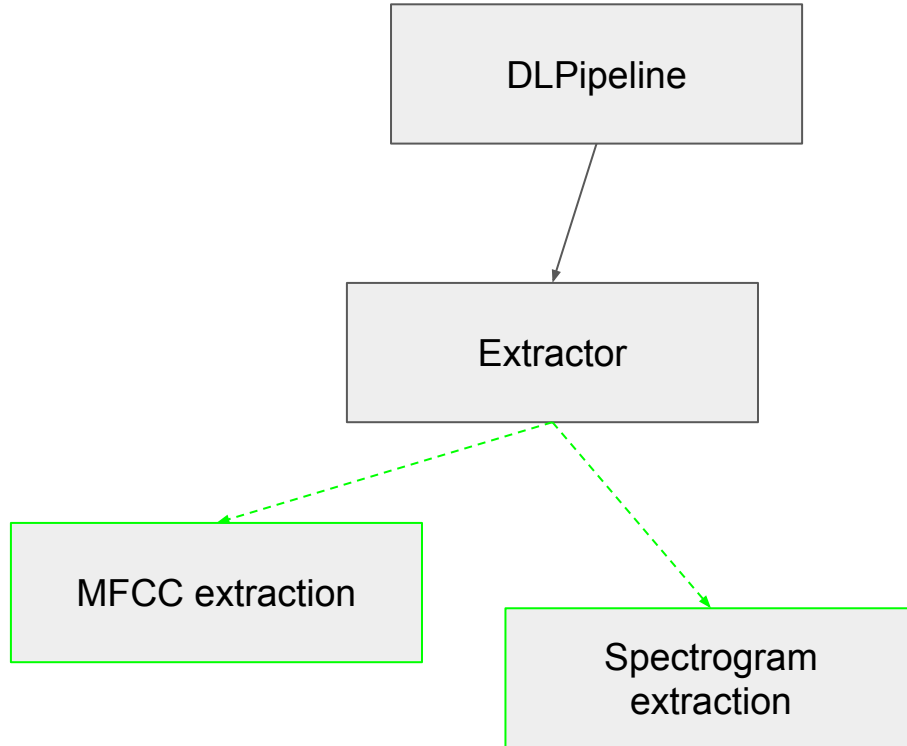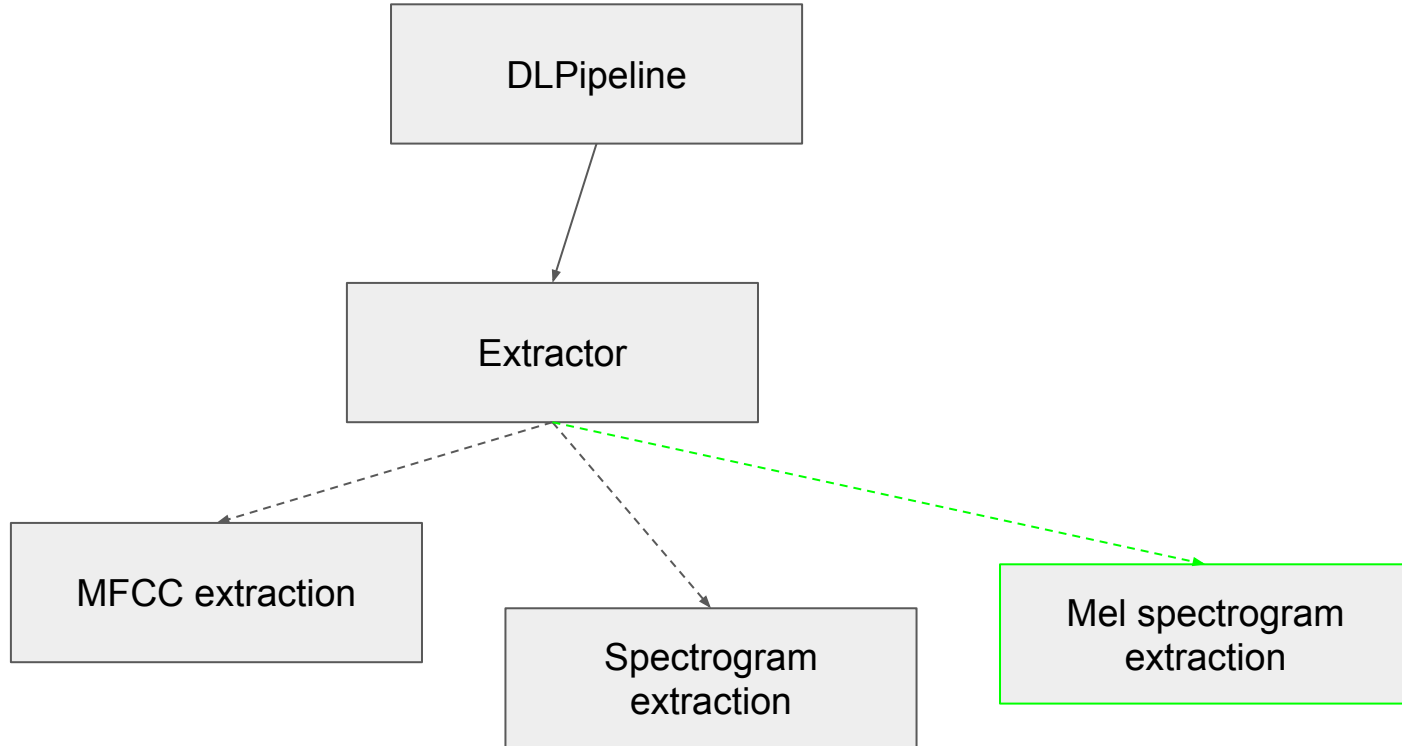# Does a class "know" too much about its dependencies?

DLPipeline

Extractor

```
DLPipeline
    |
    v
Extractor
   /    \
  v      v
MFCC extraction    Spectrogram
                    extraction
```

```
┌─────────────────┐
│   DLPipeline    │
└─────────────────┘
         │
         ▼
┌─────────────────┐
│    Extractor    │
└─────────────────┘
    │    │     ╲
    ▼    ▼      ╲
```

MFCC extraction

Spectrogram extraction

Mel spectrogram extraction

POLYMORPHISM