Design Patterns

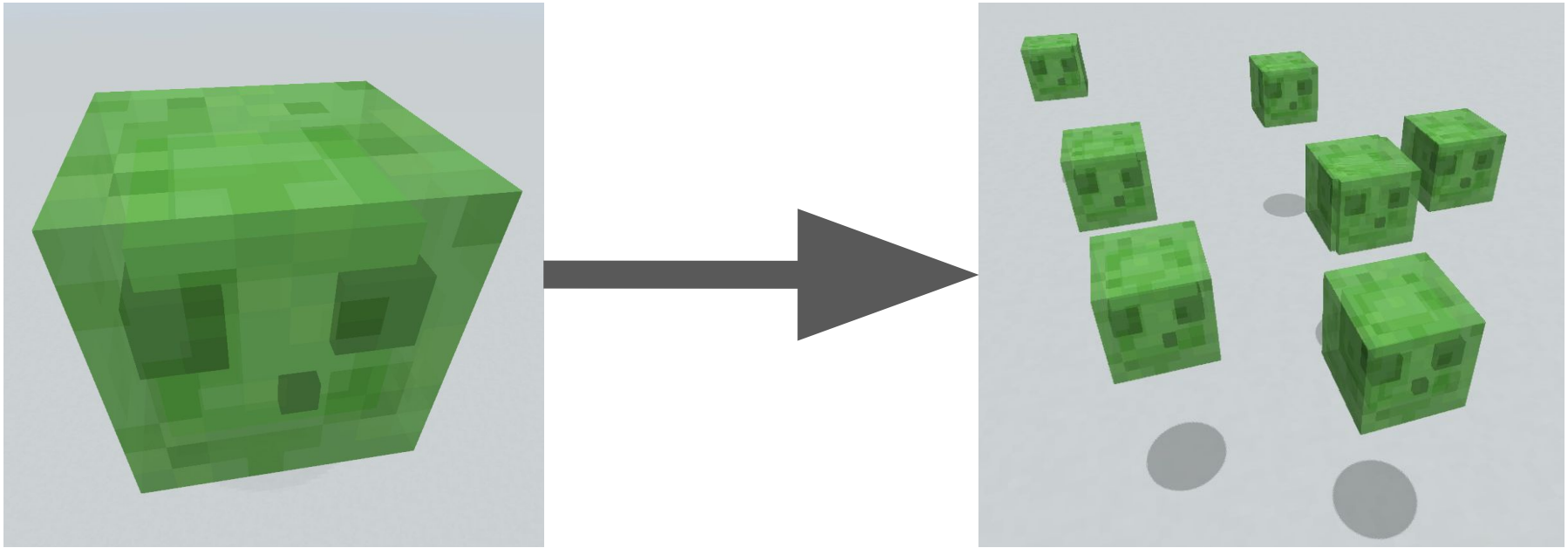
Was ist ein Design Pattern?

“In software engineering, a [...] design pattern is a general, reusable solution to a [...] problem within a given context [...].”

-- Wikipedia

Design Strategies

Divide and Conquer



Function Orientation

Object Orientation

Object Orientation

Function Orientation

Bottom-up

Top-down

Bottom-up

Top-down

Levels of Implementation

Code zu den nächsten Folien: code/iterator

Invisible

```
foreach (var s in list) Console.WriteLine(s);
```

Formal

```
for(var iterator = list.GetEnumerator(); iterator.MoveNext();) Console.WriteLine(iterator.Current);
```

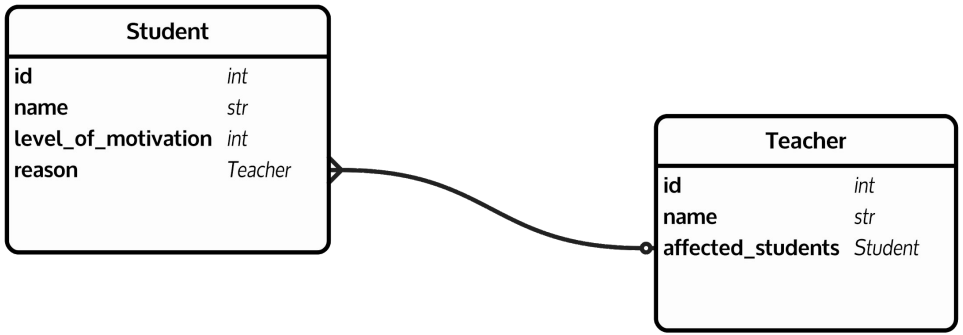
Informal

IEnumerator	vordefiniertes Interface (informelle Implementierung)
ReverseEnumerator	Implementierung für ReverseList
RandomEnumerator	Implementierung für RandomList

Aufruf: wie bei Invisible oder Formal

Arten von Design Patterns

Structural


Student
id int
name str
level_of_motivation int
reason Teacher
Teacher
id int
name str
affected_students Student

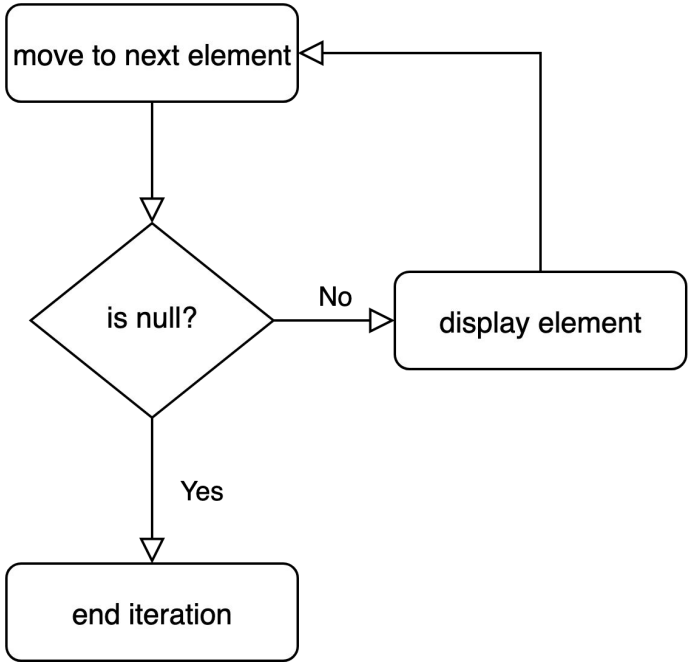
Creational

```
Fish fish = new Tuna();
```

```
Fish fish = Fish.CreateTuna();
```

```
Fish fish = TunaFactory.Create();
```

Behavioral


move to next element
is null?
No
display element
Yes
end iteration

Anti-Pattern

Entwicklung

Co-Worker: Let's create some well organized and structured code.

Me:

Architektur

Management

Hang on. Let me overthink this for a moment.