

Structural Modelling: Class Diagrams

Software Systems – Design – L3T2

Dr. Vadim Zaytsev aka @grammarware, November 2020

Class Diagram is . . .

- a static **structural diagram** that
 - describes the structure of a system
 - by showing the system's
 - classes
 - attributes
 - methods
 - relationships
- The main building block of
 - **object-oriented** modelling



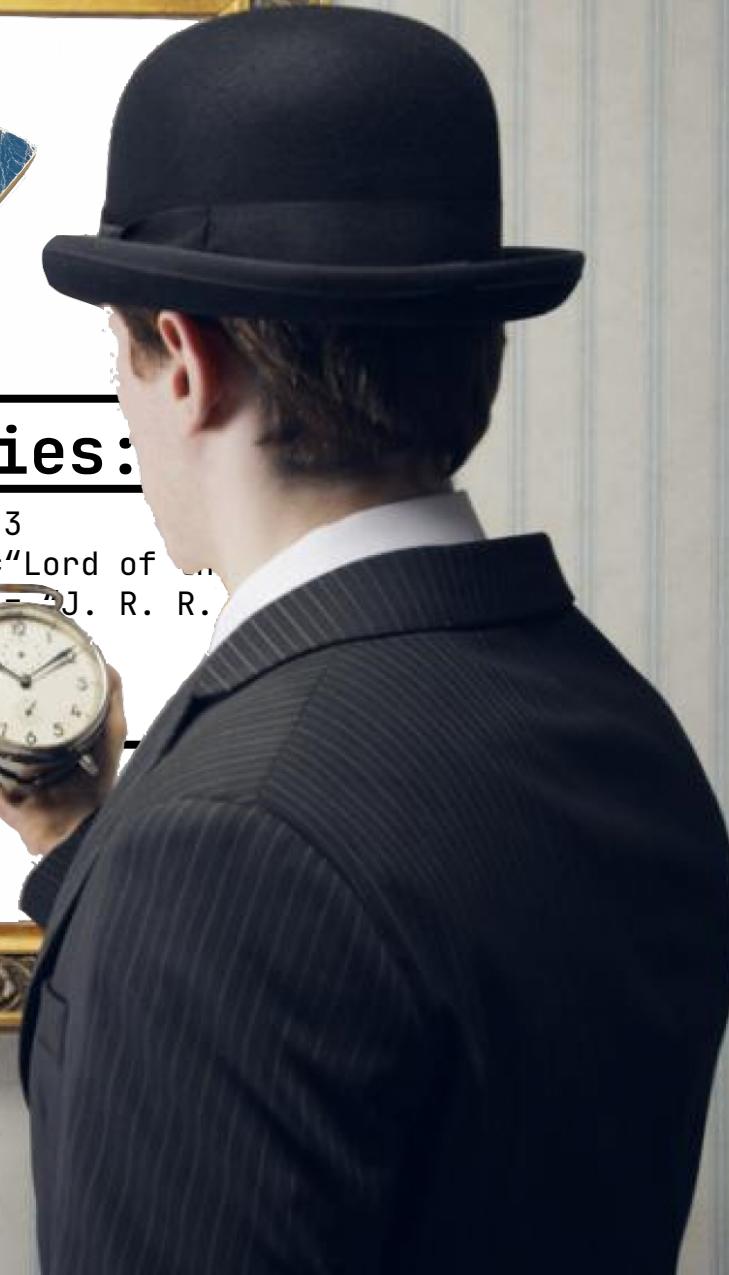
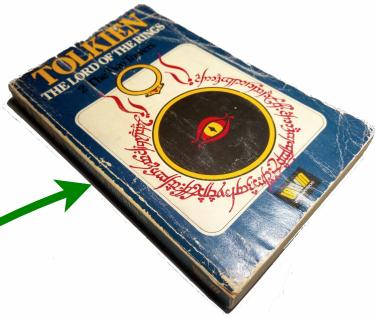
twoTowers:Book

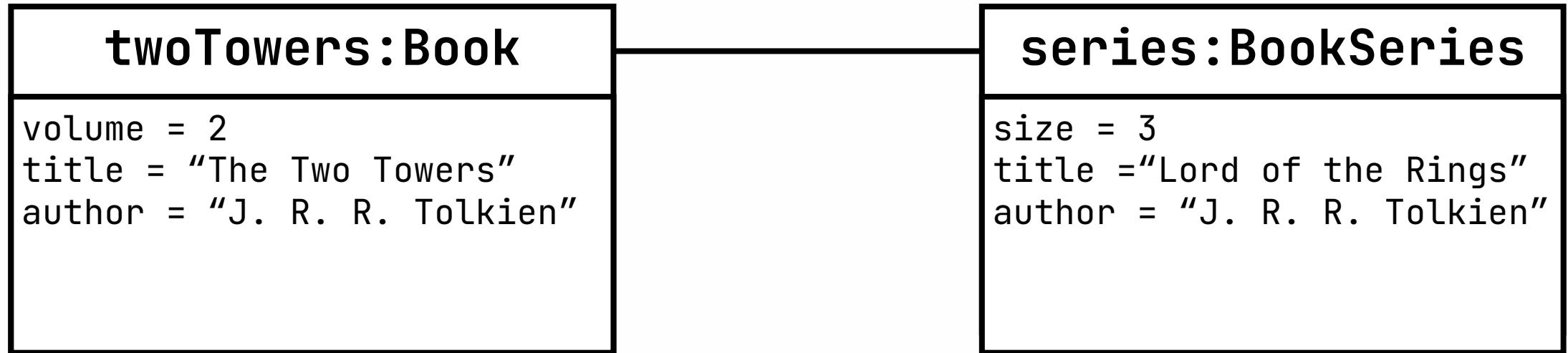
```
volume = 2  
title = "The Two Towers"  
author = "J. R. R. Tolkien"
```

series:

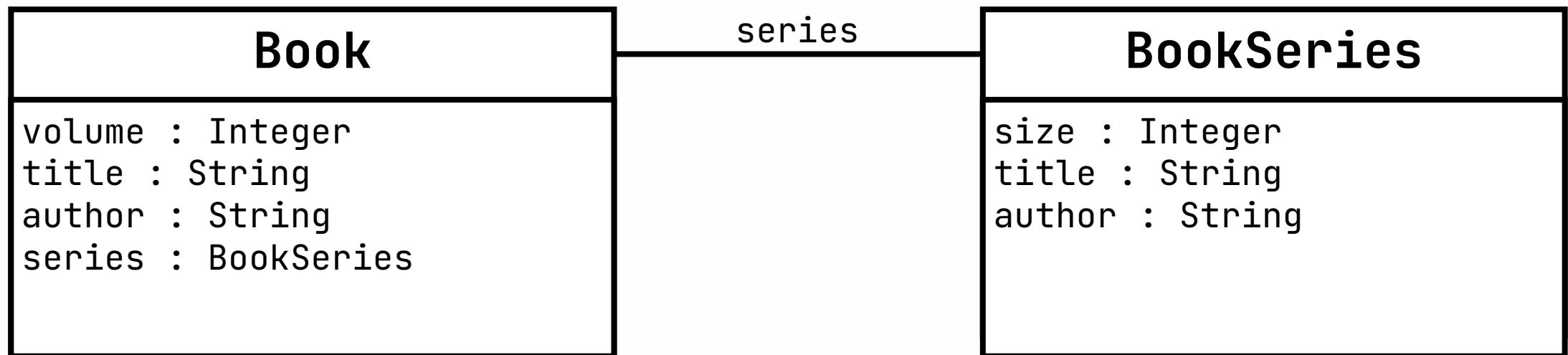
```
size = 3  
title = "Lord of the Rings"  
author = "J. R. R. Tolkien"
```

models

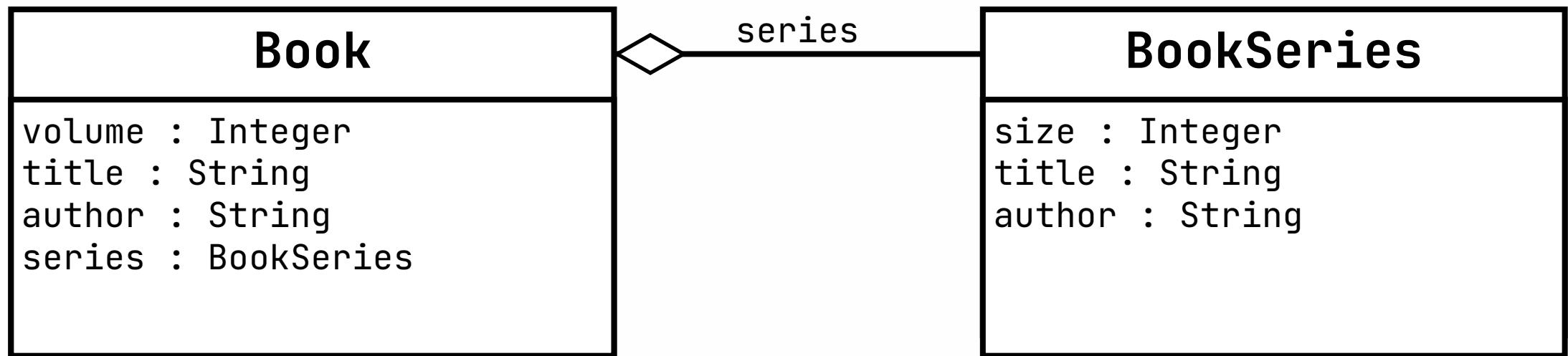




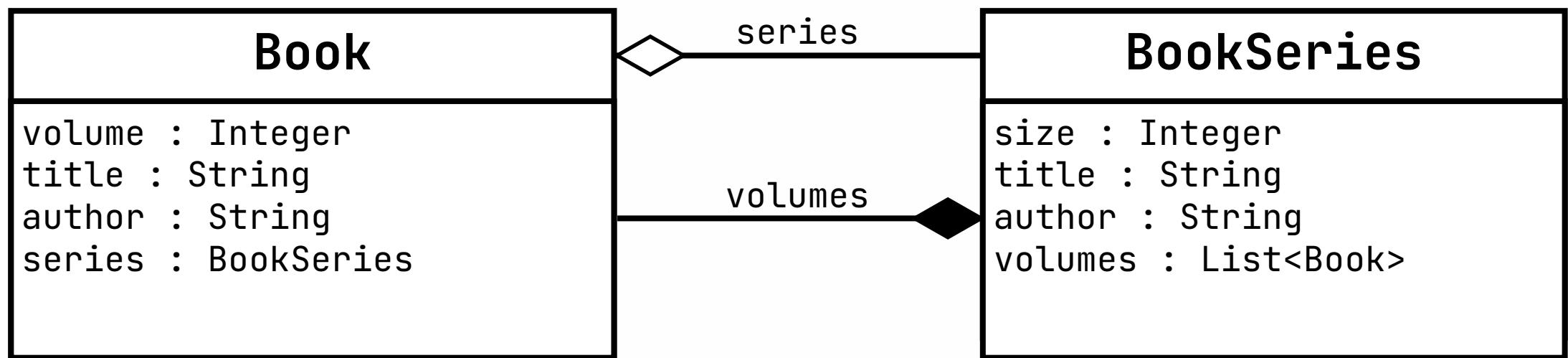
Object Diagram



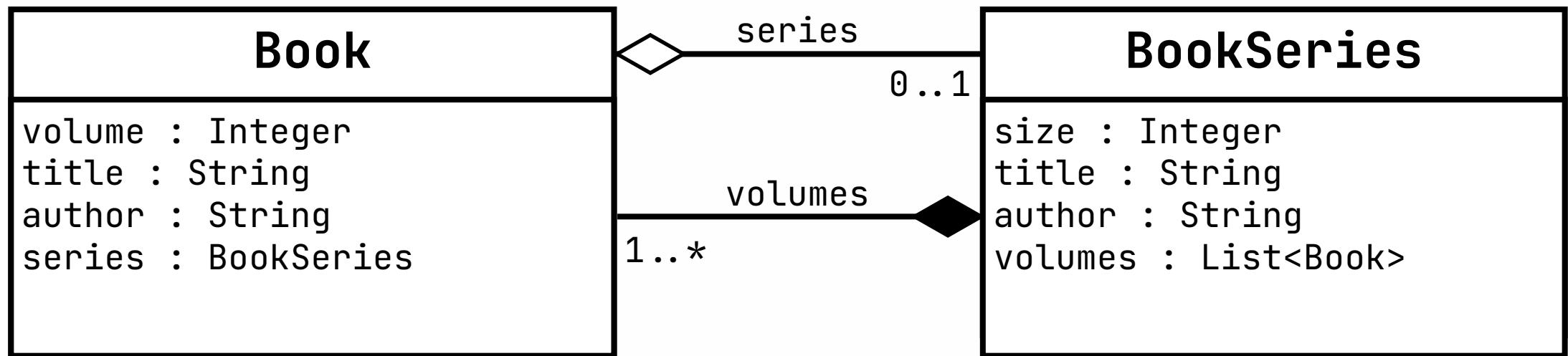
Class Diagram



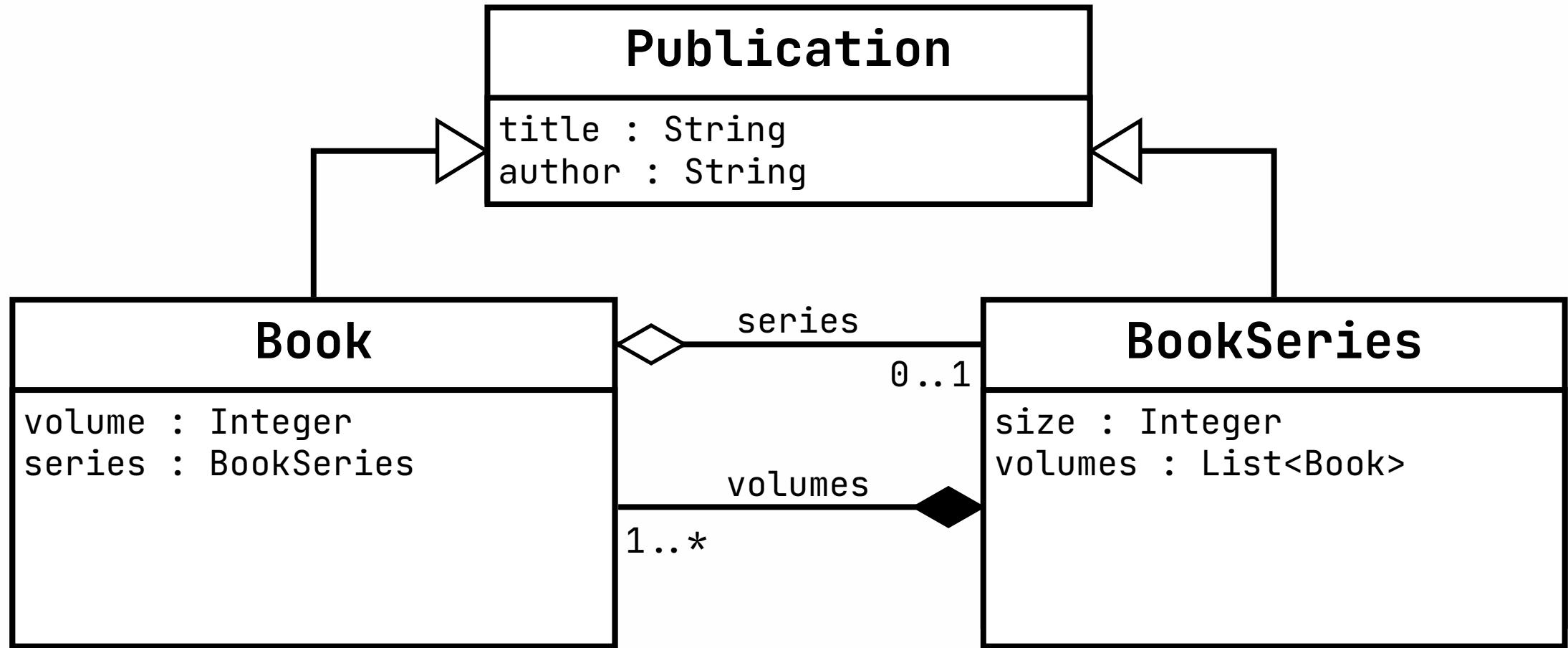
Class's Aggregation



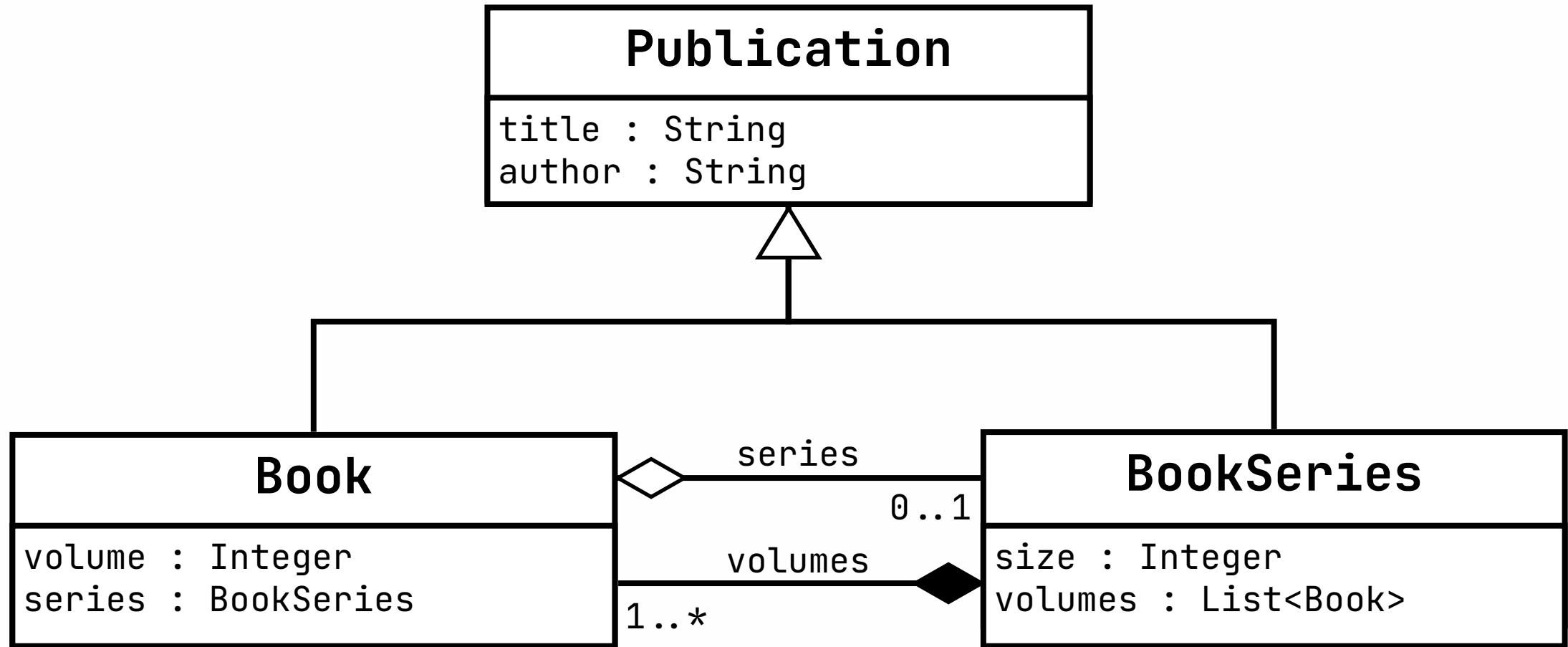
Class's Composition



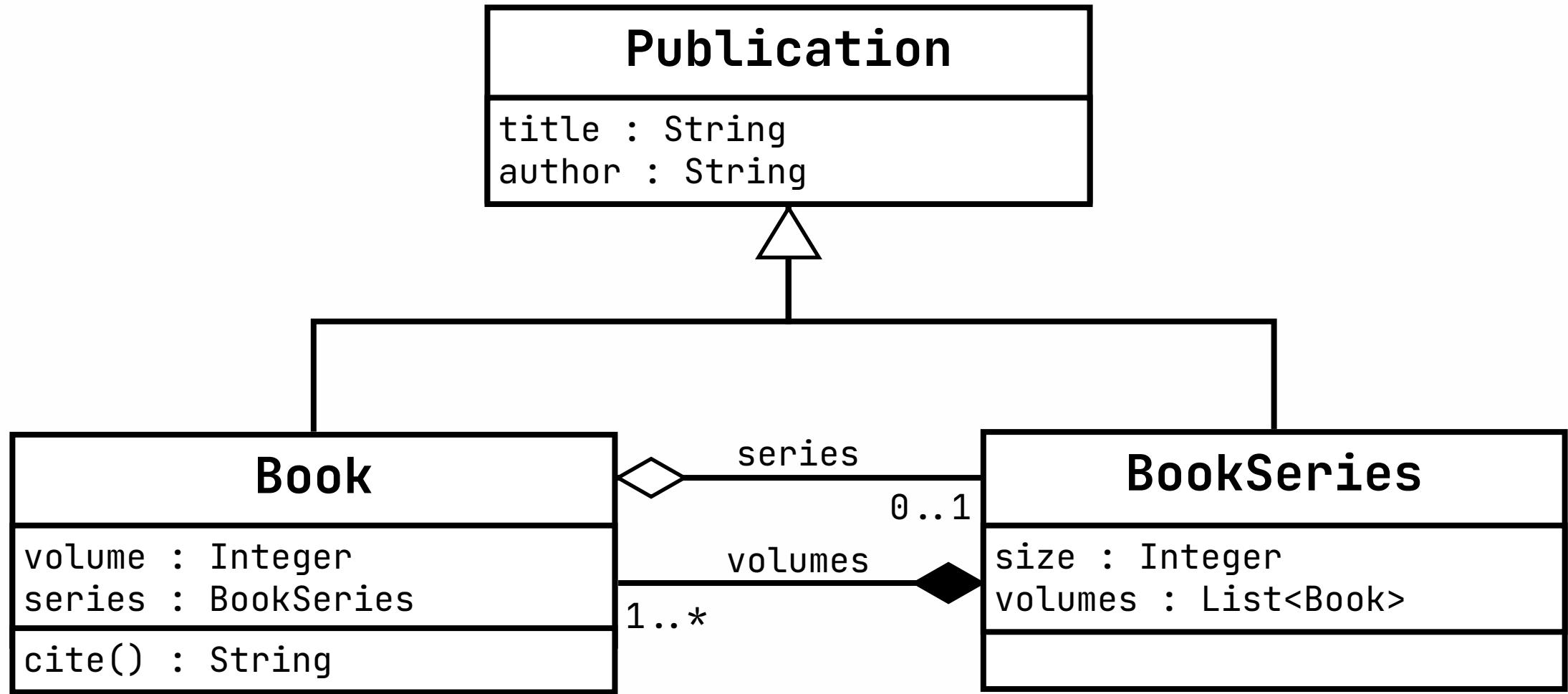
Multiplicity



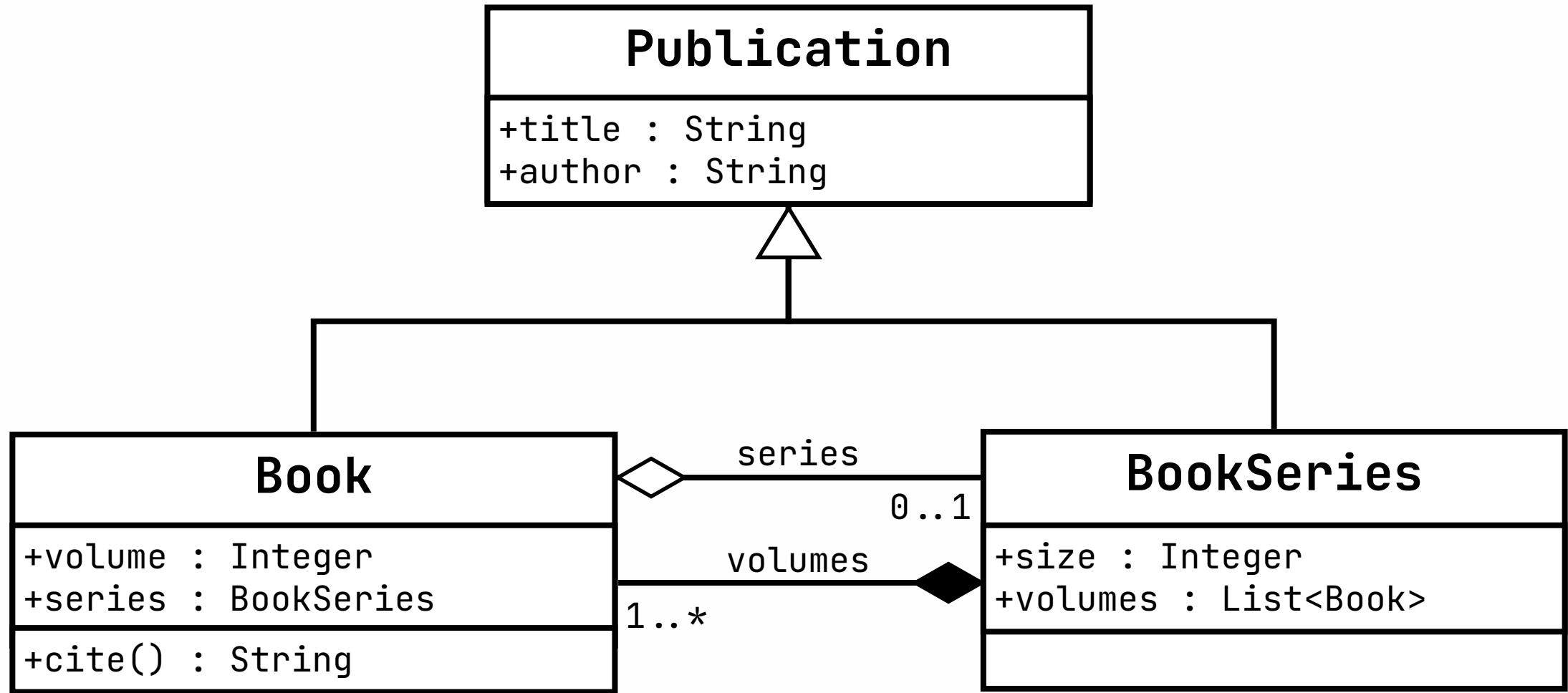
Generalisation



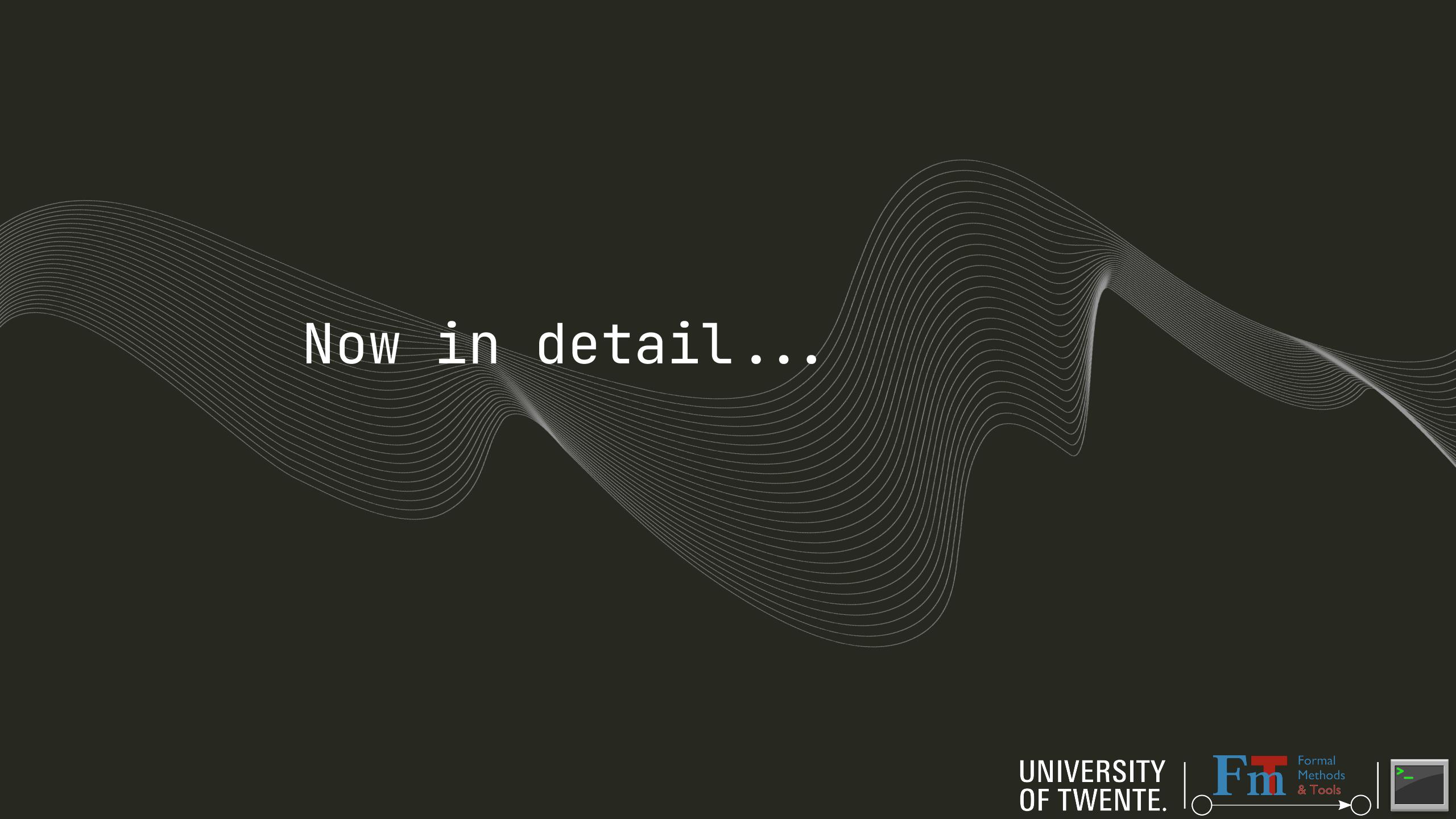
Generalisation



Methods



Access Modifiers



Now in detail...

Access Modifiers

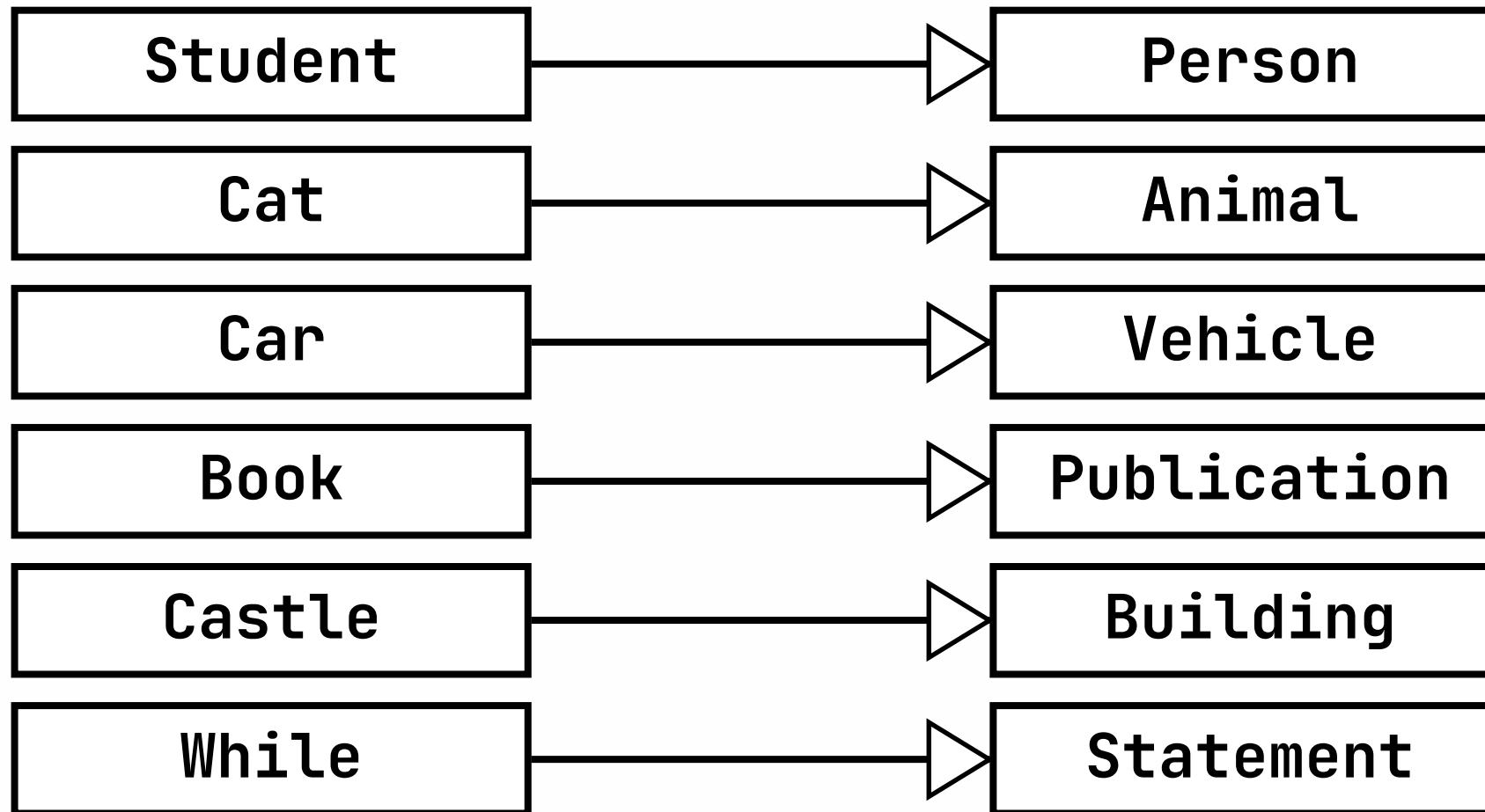
- Shows who can access the field or method
 - Anyone: + **public**
 - Nobody: - **private**
 - Children: # **protected**
 - Context: ~
- Inherited: /

Methods

- Any callable function
- Belongs to the object?
 - default
- Belongs to the class?
 - **static** in Java
 - underline in UML
- Constructors are special

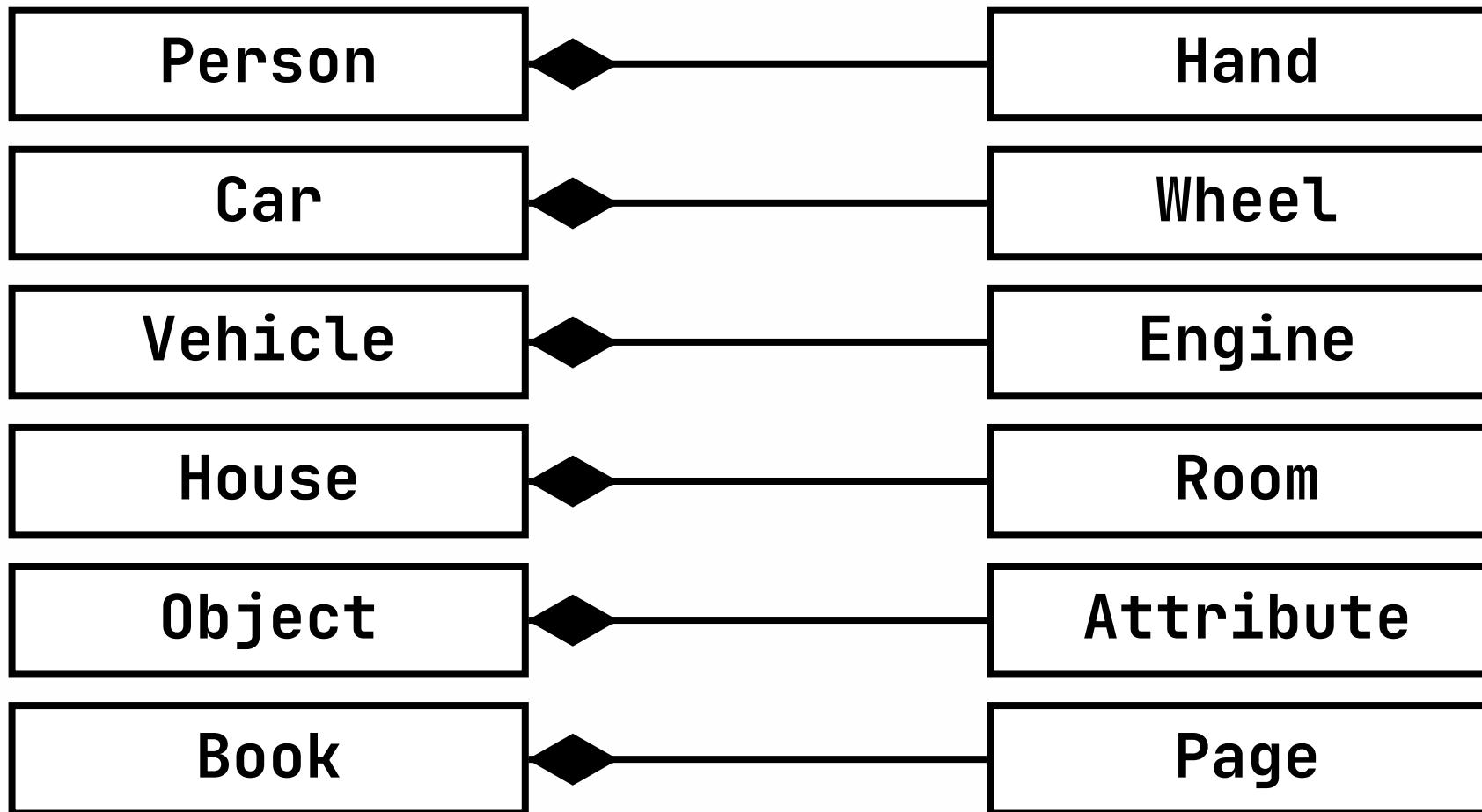
Generalisation / Specialisation

- “is a” association, shows inheritance, subclassing



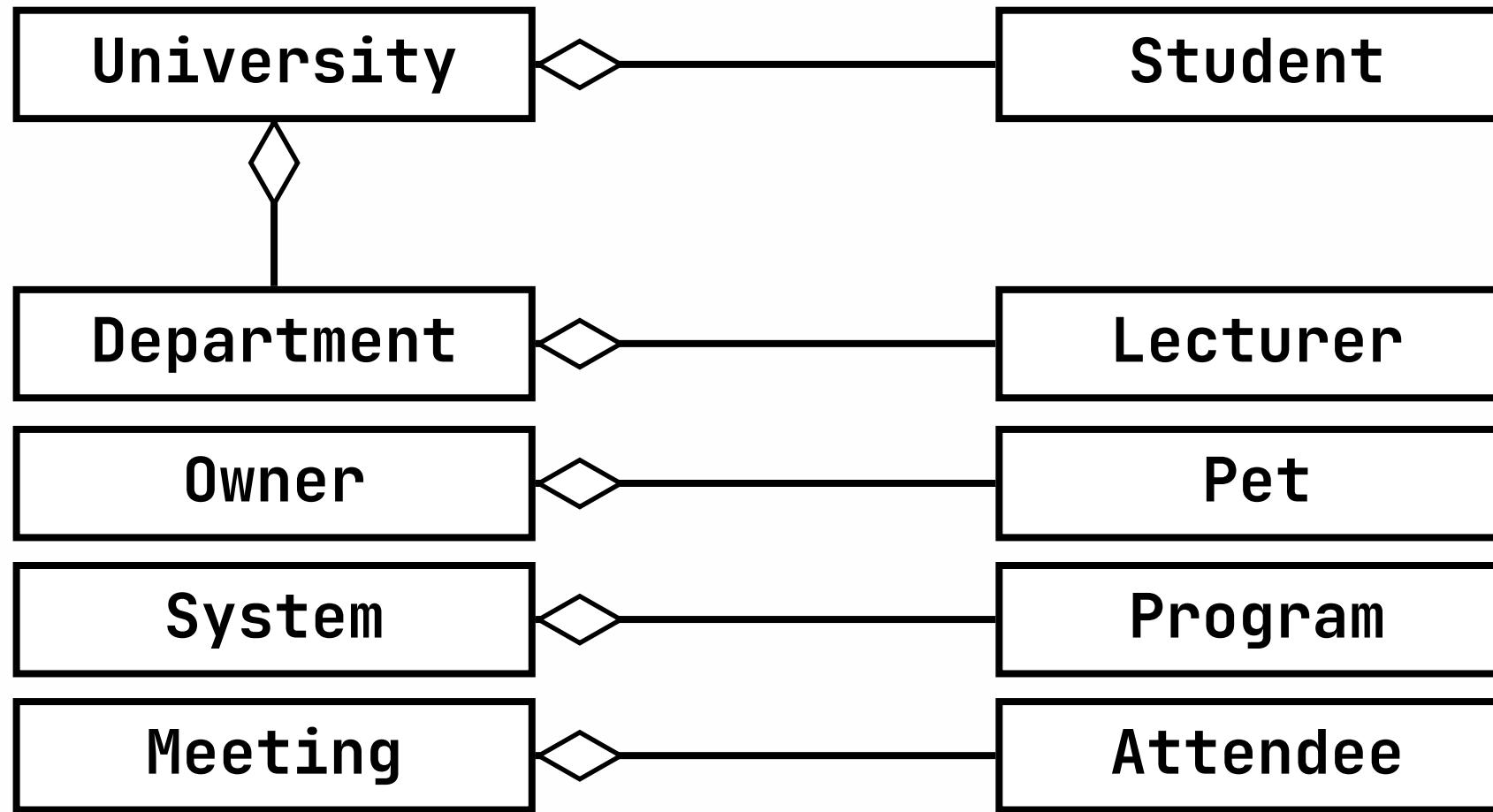
Composition

- “part of” association, shows hard containment



Aggregation

- “has a” association, shows containment



Conclusion

- Class diagrams generalise over object diagrams
- Association? Containment!
 - Aggregation 
 - Composition 
- Inheritance 
- Multiplicities
 - 0, 1, *, x..y
- Methods

Topics/slides Disclaimer

- Good ✓

- watch before Q&A
- embrace reality
- try out at labs
- ask for feedback
- apply to project
- dig deeper
- recall from slides

- Bad ✗

- slides over videos
- assumptions
- blanks
- timing

