

Kurs

Datenbankgrundlagen und Modellierung

Sebastian Maneth, Universität Bremen

maneth@uni-bremen.de

SoSe 2023

14.06.2023

Tutorial 6: UML Class Diagrams

Tutor: Marcel Westenberg

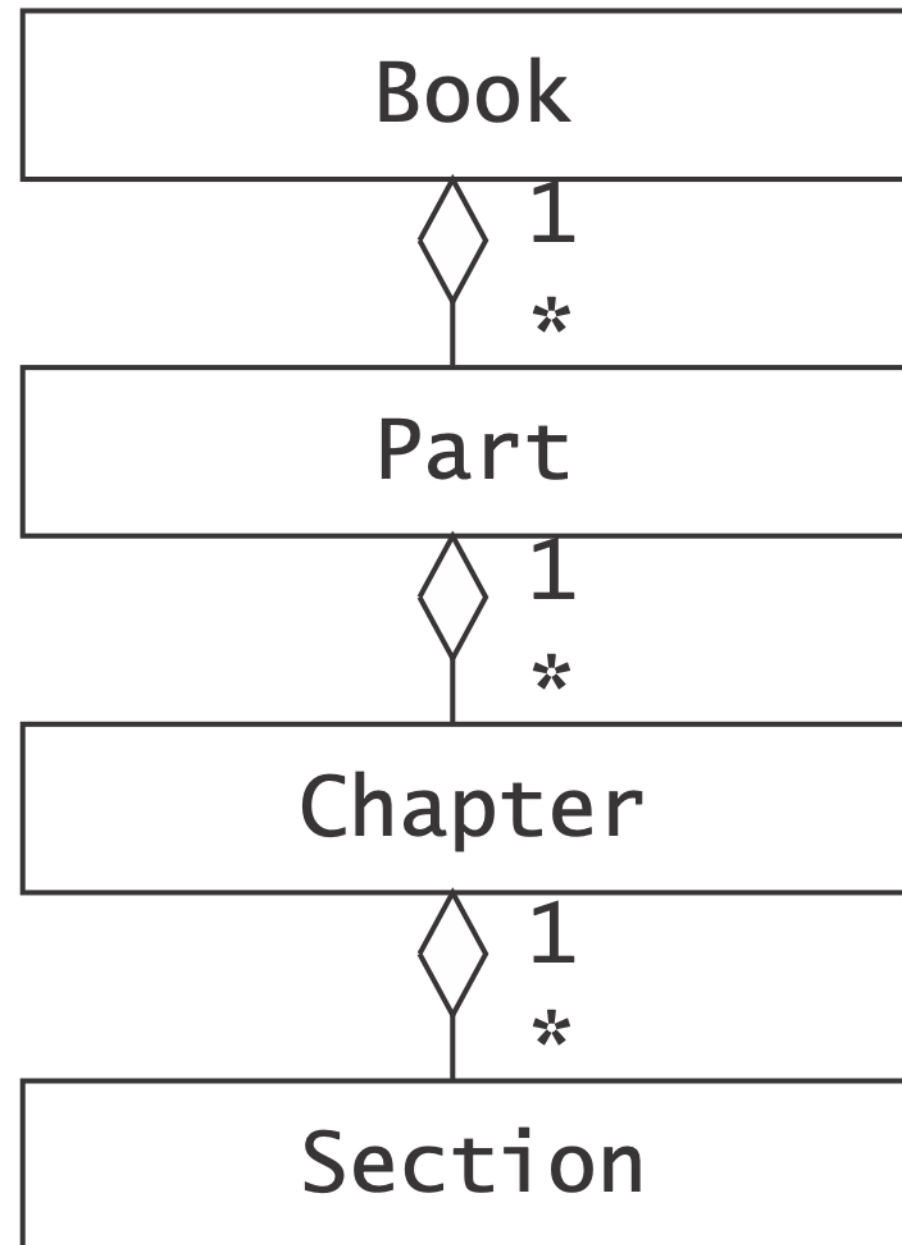
Agenda

- 1.) Modeling different kinds of Publications
- 2.) Modeling the Gregorian Calender

Draw a class diagram representing a book defined by the following statement: “A book is composed of a number of parts, which in turn are composed of a number of chapters. Chapters are composed of sections.” Focus only on classes and relationships.

Draw a class diagram representing a book defined by the following statement: “A book is composed of a number of parts, which in turn are composed of a number of chapters. Chapters are composed of sections.” Focus only on classes and relationships.

Draw a class diagram representing a book defined by the following statement: “A book is composed of a number of parts, which in turn are composed of a number of chapters. Chapters are composed of sections.” Focus only on classes and relationships.

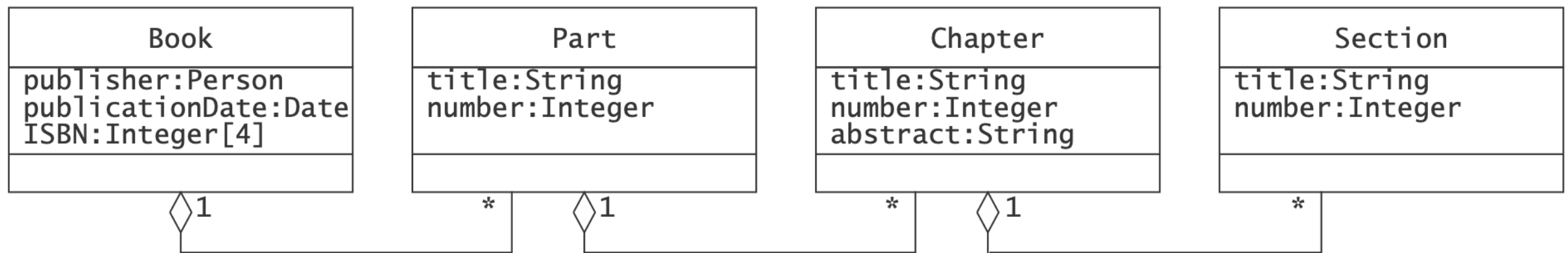


Extend the previous class diagram to include the following attributes:

- *a book includes a publisher, publication date, and an ISBN*
- *a part includes a title and a number*
- *a chapter includes a title, a number, and an abstract*
- *a section includes a title and a number*

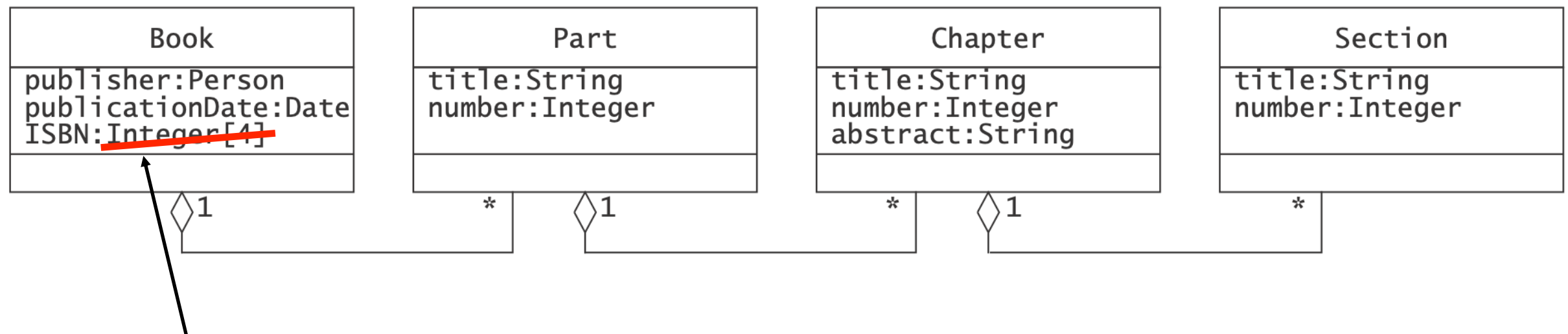
Extend the previous class diagram to include the following attributes:

- *a book includes a publisher, publication date, and an ISBN*
- *a part includes a title and a number*
- *a chapter includes a title, a number, and an abstract*
- *a section includes a title and a number*



Extend the previous class diagram to include the following attributes:

- *a book includes a publisher, publication date, and an ISBN*
- *a part includes a title and a number*
- *a chapter includes a title, a number, and an abstract*
- *a section includes a title and a number*

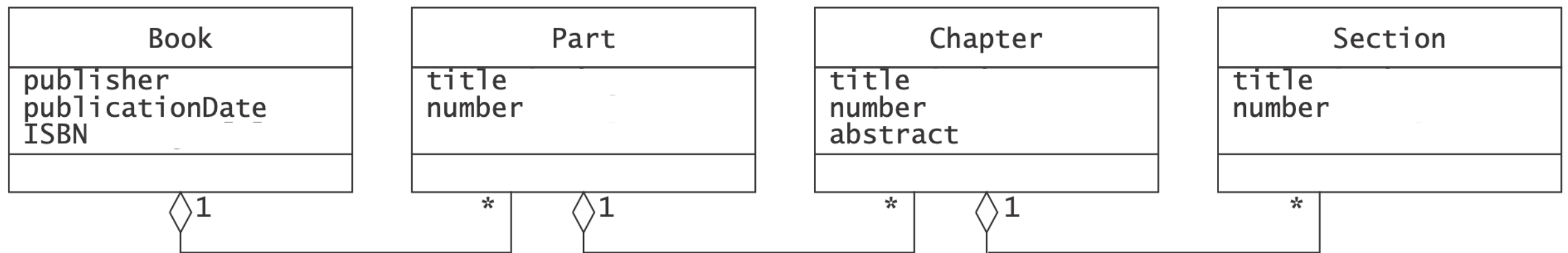


better use **String**

in fact, probably best to **remove all types**

Extend the previous class diagram to include the following attributes:

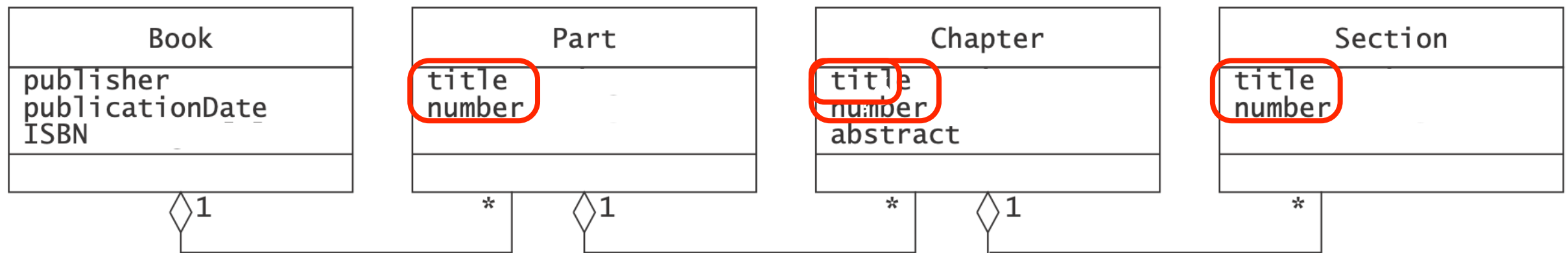
- *a book includes a publisher, publication date, and an ISBN*
- *a part includes a title and a number*
- *a chapter includes a title, a number, and an abstract*
- *a section includes a title and a number*



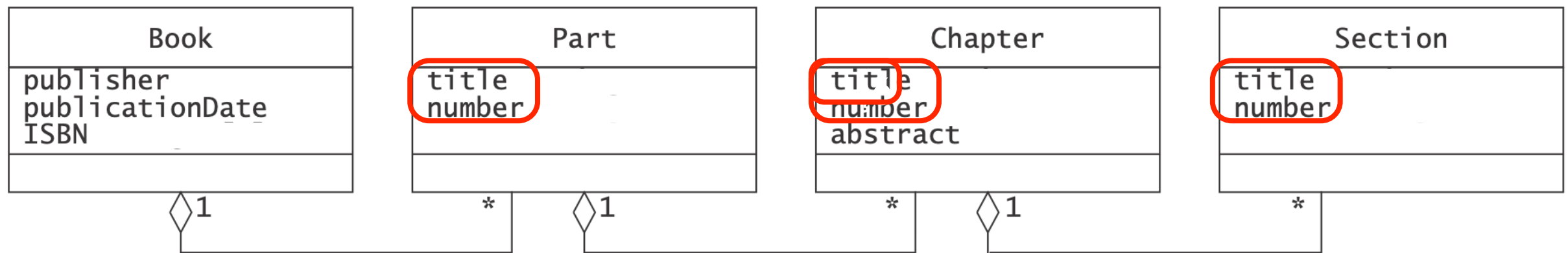
better use **String**

in fact, probably best to **remove all types**

Consider the class diagram of Exercise 2–9. Note that the Part, Chapter, and Section classes all include a title and a number attribute. Add an abstract class and a generalization relationship to factor out these two attributes into the abstract class.

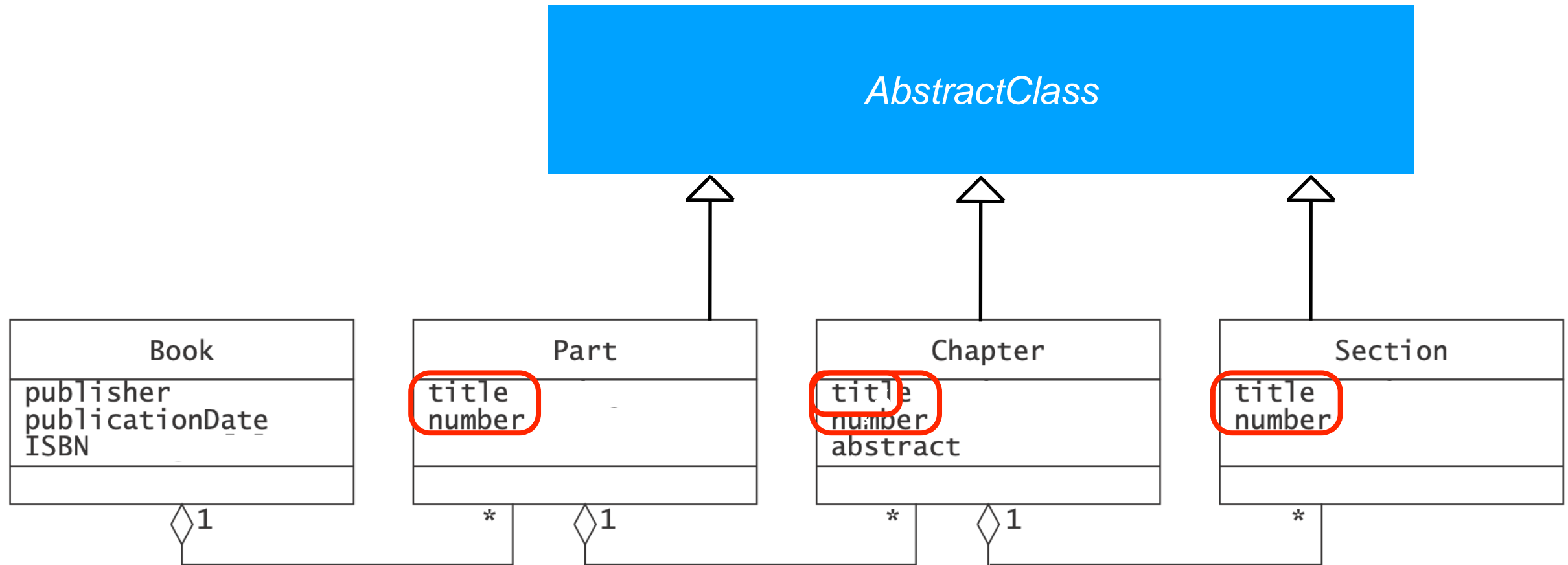


Consider the class diagram of Exercise 2–9. Note that the Part, Chapter, and Section classes all include a title and a number attribute. Add an abstract class and a generalization relationship to factor out these two attributes into the abstract class.



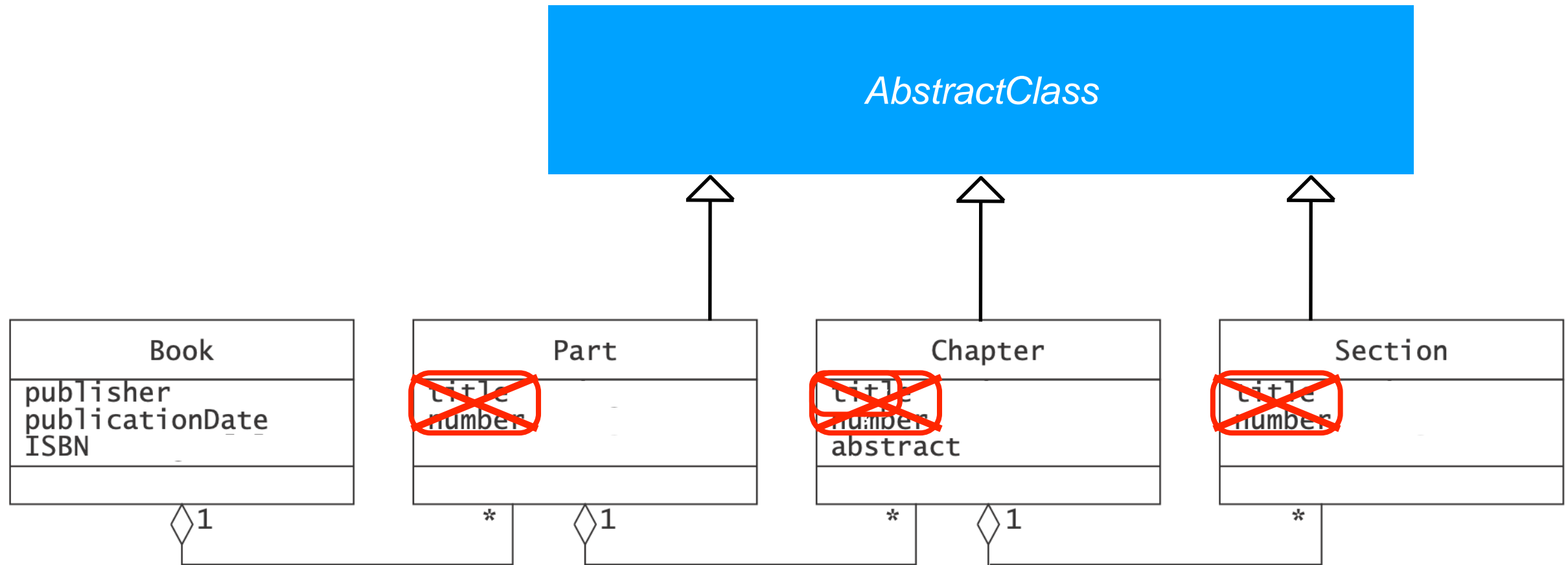
Remember: the (big white) arrow for **inheritance** points from the subclass to the superclass

Consider the class diagram of Exercise 2–9. Note that the Part, Chapter, and Section classes all include a title and a number attribute. Add an abstract class and a generalization relationship to factor out these two attributes into the abstract class.



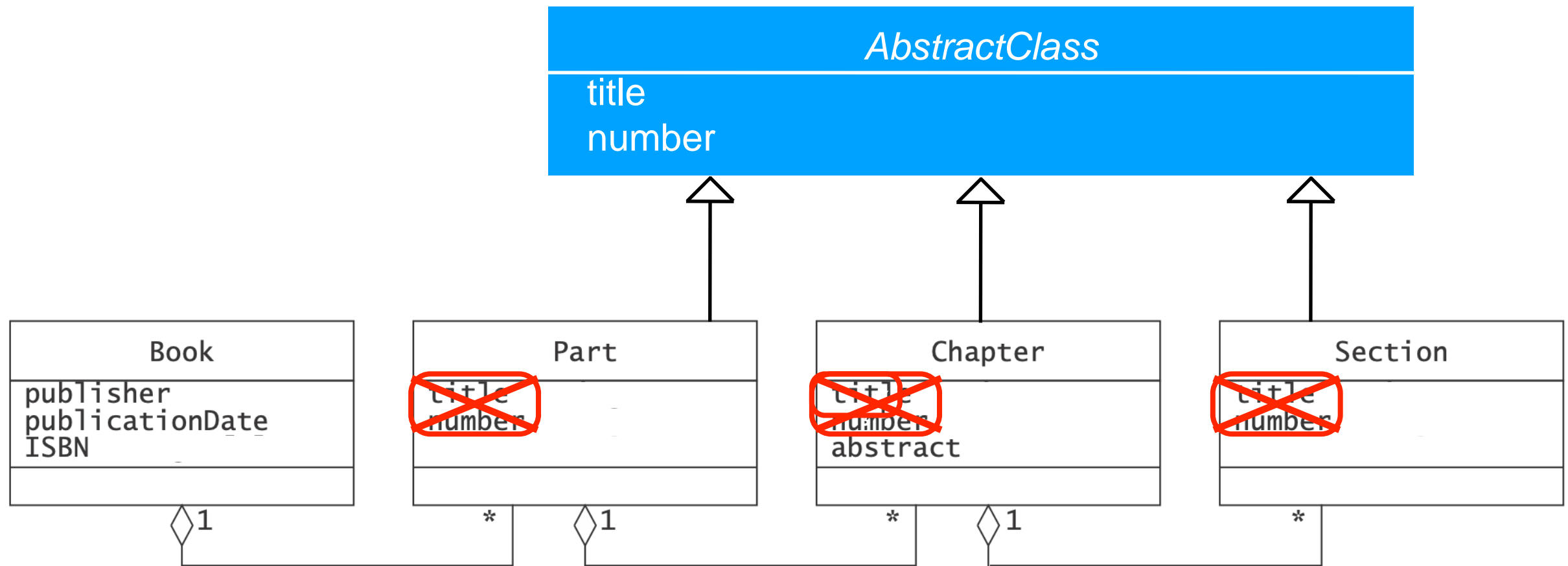
Remember: the (big white) arrow for **inheritance** points from the subclass to the superclass

Consider the class diagram of Exercise 2–9. Note that the Part, Chapter, and Section classes all include a title and a number attribute. Add an abstract class and a generalization relationship to factor out these two attributes into the abstract class.



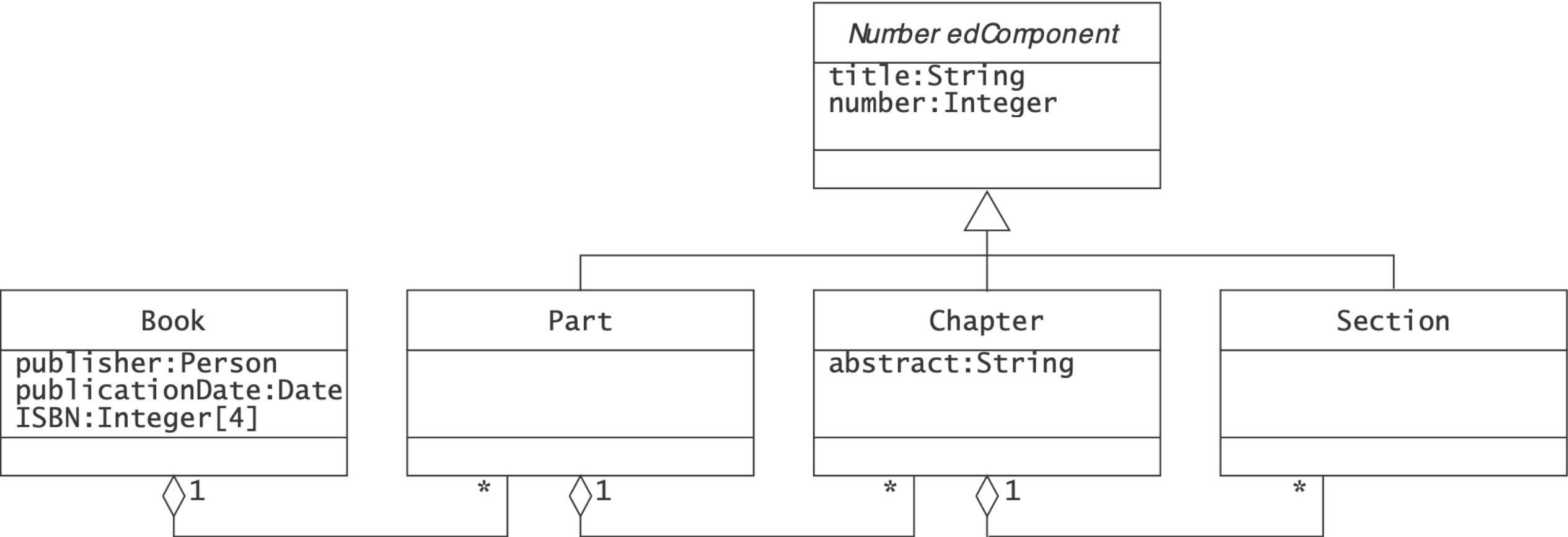
Remember: the (big white) arrow for **inheritance** points from the subclass to the superclass

Consider the class diagram of Exercise 2–9. Note that the Part, Chapter, and Section classes all include a title and a number attribute. Add an abstract class and a generalization relationship to factor out these two attributes into the abstract class.



Remember: the (big white) arrow for **inheritance** points from the subclass to the superclass

Consider the class diagram of Exercise 2–9. Note that the Part, Chapter, and Section classes all include a title and a number attribute. Add an abstract class and a generalization relationship to factor out these two attributes into the abstract class.



Draw a class diagram representing the relationship between parents and children. Take into account that a person can have both a parent and a child. Annotate associations with roles and multiplicities.

Draw a class diagram representing the relationship between parents and children. Take into account that a person can have both a parent and a child. Annotate associations with roles and multiplicities.

A common modeling mistake by novices is to draw two classes, one for the parent and one for the child.



2–12 Draw a class diagram for bibliographic references. Use the references in Appendix C, Bibliography, to test your class diagram. Your class diagram should be as detailed as possible.

The domain of bibliographic references is rich and complex. Consequently, students should deepen their understanding of the domain before they attempt to draw a class diagram (similar to the requirements analysis of a system). Figure 2-8 depicts an incomplete sample solution that could be accepted as sufficient from an instructor. The class diagram should minimally include the following concepts:

- An abstract class *Publication* (or *BibliographicReference*)
- A many to many relationship between *Author* and *Publication*
- At least three or more concrete classes refining *Publication*
- At least one aggregation relationship (e.g., between *Journal* and *Article* or *Proceedings* and *ConferencePaper*). Both ends of the aggregation should also be subclasses of *Publication*.

2–12 Draw a class diagram for bibliographic references. Use the references in Appendix C, Bibliography, to test your class diagram. Your class diagram should be as detailed as possible.

The domain of bibliographic references is rich and complex. Consequently, students should deepen their understanding of the domain before they attempt to draw a class diagram (similar to the requirements analysis of a system). Figure 2-8 depicts an incomplete sample solution that could be accepted as sufficient from an instructor. The class diagram should minimally include the following concepts:

- An abstract class *Publication* (or *BibliographicReference*)
- A many to many relationship between *Author* and *Publication*
- At least three or more concrete classes refining *Publication*
- At least one aggregation relationship (e.g., between *Journal* and *Article* or *Proceedings* and *ConferencePaper*). Both ends of the aggregation should also be subclasses of *Publication*.

There are two types of **Books**:

- (conference) **Proceedings** (consisting of **ConferencePapers**)
- **Compendium** (consisting of **CompendiumChapters**)

2–12 Draw a class diagram for bibliographic references. Use the references in Appendix C, Bibliography, to test your class diagram. Your class diagram should be as detailed as possible.

The domain of bibliographic references is rich and complex. Consequently, students should deepen their understanding of the domain before they attempt to draw a class diagram (similar to the requirements analysis of a system). Figure 2-8 depicts an incomplete sample solution that could be accepted as sufficient from an instructor. The class diagram should minimally include the following concepts:

- An abstract class *Publication* (or *BibliographicReference*)
- A many to many relationship between Author and *Publication*
- At least three or more concrete classes refining *Publication*
- At least one aggregation relationship (e.g., between *Journal* and *Article* or *Proceedings* and *ConferencePaper*). Both ends of the aggregation should also be subclasses of *Publication*.

There are two types of **Books**:

- (conference) **Proceedings** (consisting of **ConferencePapers**)
- **Compendium** (consisting of **CompendiumChapters**)

There is a many-to-many relationship between **Author** and **Publication**

2–12 Draw a class diagram for bibliographic references. Use the references in Appendix C, Bibliography, to test your class diagram. Your class diagram should be as detailed as possible.

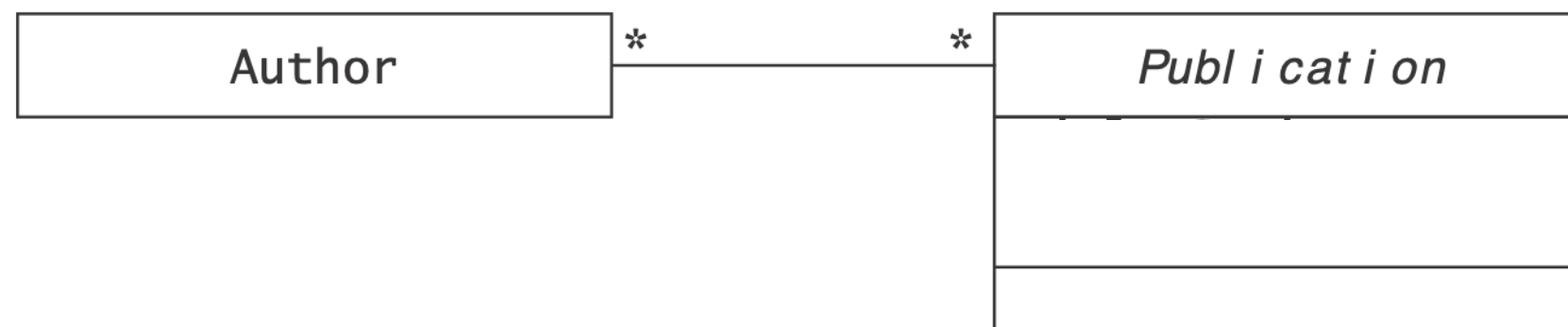
The domain of bibliographic references is rich and complex. Consequently, students should deepen their understanding of the domain before they attempt to draw a class diagram (similar to the requirements analysis of a system). Figure 2-8 depicts an incomplete sample solution that could be accepted as sufficient from an instructor. The class diagram should minimally include the following concepts:

- An abstract class *Publication* (or *BibliographicReference*)
- A many to many relationship between Author and *Publication*
- At least three or more concrete classes refining *Publication*
- At least one aggregation relationship (e.g., between Journal and Article or Proceedings and ConferencePaper). Both ends of the aggregation should also be subclasses of *Publication*.

There are two types of Books:

- (conference) **Proceedings** (consisting of **ConferencePapers**)
- **Compendium** (consisting of **CompendiumChapters**)

There is a many-to-many relationship between **Author** and **Publication**



2–12 Draw a class diagram for bibliographic references. Use the references in Appendix C, Bibliography, to test your class diagram. Your class diagram should be as detailed as possible.

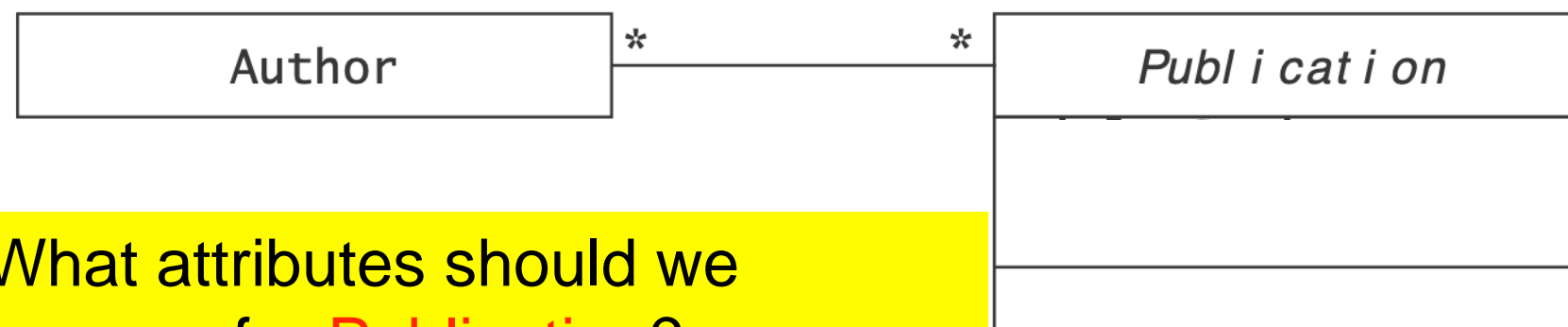
The domain of bibliographic references is rich and complex. Consequently, students should deepen their understanding of the domain before they attempt to draw a class diagram (similar to the requirements analysis of a system). Figure 2-8 depicts an incomplete sample solution that could be accepted as sufficient from an instructor. The class diagram should minimally include the following concepts:

- An abstract class *Publication* (or *BibliographicReference*)
- A many to many relationship between *Author* and *Publication*
- At least three or more concrete classes refining *Publication*
- At least one aggregation relationship (e.g., between *Journal* and *Article* or *Proceedings* and *ConferencePaper*). Both ends of the aggregation should also be subclasses of *Publication*.

There are two types of **Books**:

- (conference) **Proceedings** (consisting of **ConferencePapers**)
- **Compendium** (consisting of **CompendiumChapters**)

There is a many-to-many relationship between **Author** and **Publication**



What attributes should we assume for **Publication**?

2–12 Draw a class diagram for bibliographic references. Use the references in Appendix C, Bibliography, to test your class diagram. Your class diagram should be as detailed as possible.

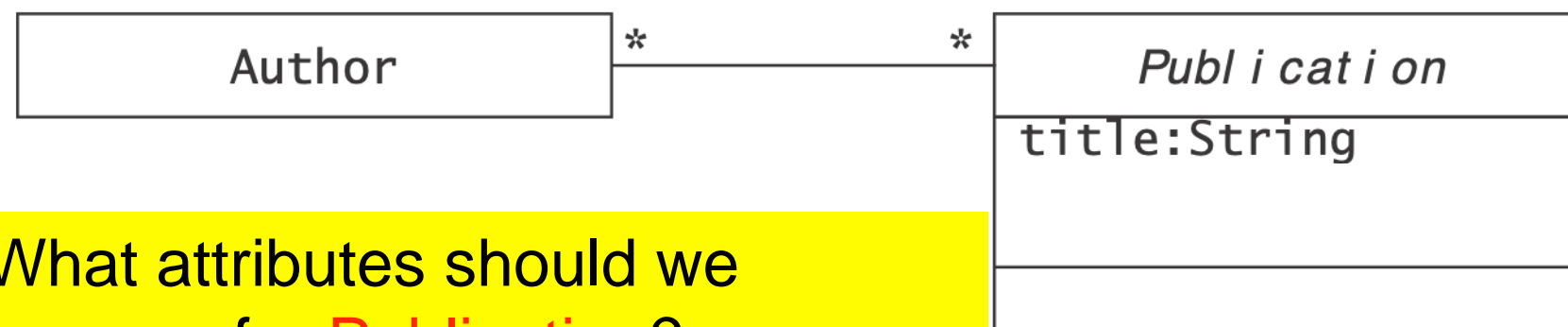
The domain of bibliographic references is rich and complex. Consequently, students should deepen their understanding of the domain before they attempt to draw a class diagram (similar to the requirements analysis of a system). Figure 2-8 depicts an incomplete sample solution that could be accepted as sufficient from an instructor. The class diagram should minimally include the following concepts:

- An abstract class *Publication* (or *BibliographicReference*)
- A many to many relationship between *Author* and *Publication*
- At least three or more concrete classes refining *Publication*
- At least one aggregation relationship (e.g., between *Journal* and *Article* or *Proceedings* and *ConferencePaper*). Both ends of the aggregation should also be subclasses of *Publication*.

There are two types of **Books**:

- (conference) **Proceedings** (consisting of **ConferencePapers**)
- **Compendium** (consisting of **CompendiumChapters**)

There is a many-to-many relationship between **Author** and **Publication**



What attributes should we assume for **Publication**?

2–12 Draw a class diagram for bibliographic references. Use the references in Appendix C, Bibliography, to test your class diagram. Your class diagram should be as detailed as possible.

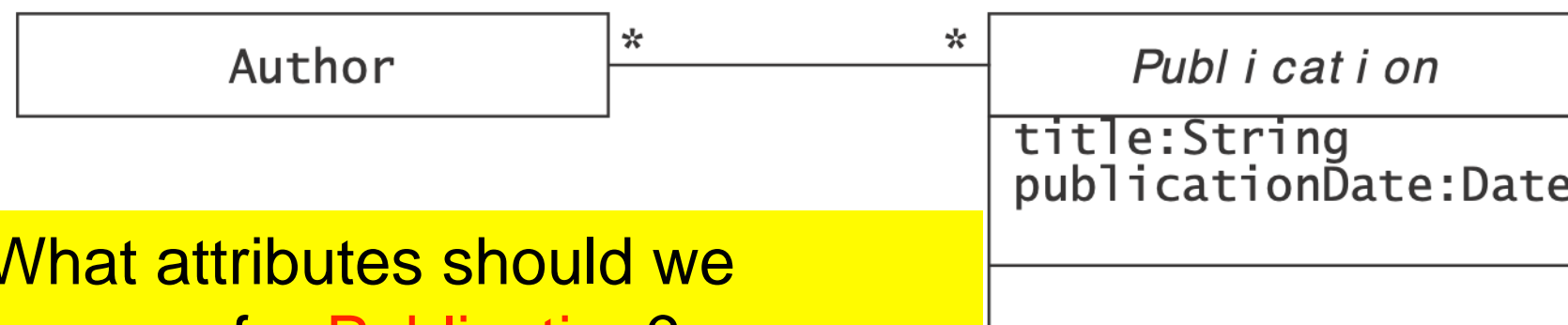
The domain of bibliographic references is rich and complex. Consequently, students should deepen their understanding of the domain before they attempt to draw a class diagram (similar to the requirements analysis of a system). Figure 2-8 depicts an incomplete sample solution that could be accepted as sufficient from an instructor. The class diagram should minimally include the following concepts:

- An abstract class *Publication* (or *BibliographicReference*)
- A many to many relationship between *Author* and *Publication*
- At least three or more concrete classes refining *Publication*
- At least one aggregation relationship (e.g., between *Journal* and *Article* or *Proceedings* and *ConferencePaper*). Both ends of the aggregation should also be subclasses of *Publication*.

There are two types of **Books**:

- (conference) **Proceedings** (consisting of **ConferencePapers**)
- **Compendium** (consisting of **CompendiumChapters**)

There is a many-to-many relationship between **Author** and **Publication**



What attributes should we assume for **Publication**?

2–12 Draw a class diagram for bibliographic references. Use the references in Appendix C, Bibliography, to test your class diagram. Your class diagram should be as detailed as possible.

The domain of bibliographic references is rich and complex. Consequently, students should deepen their understanding of the domain before they attempt to draw a class diagram (similar to the requirements analysis of a system). Figure 2-8 depicts an incomplete sample solution that could be accepted as sufficient from an instructor. The class diagram should minimally include the following concepts:

- An abstract class *Publication* (or *BibliographicReference*)
 - A many to many relationship between *Author* and *Publication*
 - At least three or more concrete classes refining *Publication*
 - At least one aggregation relationship (e.g., between Journal and Article or Proceedings and ConferencePaper). Both ends of the aggregation should also be subclasses of *Publication*.
- a **Journal** consists of **Articles**
 - a **Proceedings** (Book) consists of **ConferencePapers**

There are two types of **Books**:

- (conference) **Proceedings** (consisting of **ConferencePapers**)
- **Compendium** (consisting of **CompendiumChapters**)

2–12 Draw a class diagram for bibliographic references. Use the references in Appendix C, Bibliography, to test your class diagram. Your class diagram should be as detailed as possible.

The domain of bibliographic references is rich and complex. Consequently, students should deepened their understanding of the domain before they attempt to draw a class diagram (similar to the requirements analysis of a system). Figure 2-8 depicts an incomplete sample solution that could be accepted as sufficient from an instructor. The class diagram should minimally include the following concepts:

- An abstract class *Publication* (or *BibliographicReference*)
- A many to many relationship between *Author* and *Publication*
- At least three or more concrete classes refining *Publication*
- At least one aggregation relationship (e.g., between Journal and Article or Proceedings and ConferencePaper). Both ends of the aggregation should also be subclasses of *Publication*.

— a **Journal** **consists of** **Articles**

— a **Proceedings** (Book) **consists of** **ConferencePapers**

how do we draw **consist of**?

 weak (Aggregation)

 strong (Composition)

There are two types of **Books**:

- (conference) **Proceedings** (consisting of **ConferencePapers**)
- **Compendium** (consisting of **CompendiumChapters**)

2–12 Draw a class diagram for bibliographic references. Use the references in Appendix C, Bibliography, to test your class diagram. Your class diagram should be as detailed as possible.

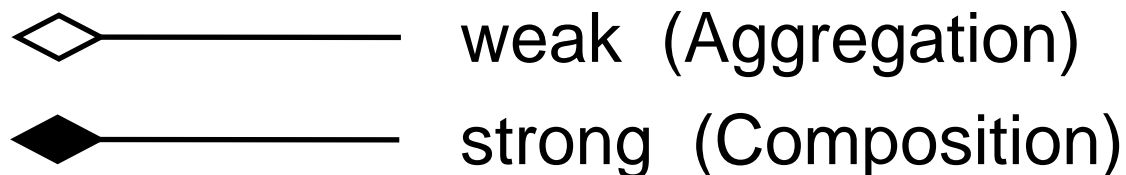
The domain of bibliographic references is rich and complex. Consequently, students should deepen their understanding of the domain before they attempt to draw a class diagram (similar to the requirements analysis of a system). Figure 2-8 depicts an incomplete sample solution that could be accepted as sufficient from an instructor. The class diagram should minimally include the following concepts:

- An abstract class *Publication* (or *BibliographicReference*)
- A many to many relationship between *Author* and *Publication*
- At least three or more concrete classes refining *Publication*
- At least one aggregation relationship (e.g., between Journal and Article or Proceedings and ConferencePaper). Both ends of the aggregation should also be subclasses of *Publication*.

— a **Journal** consists of **Articles**

— a **Proceedings** (Book) consists of **ConferencePapers**

how do we draw **consist of**?



Which one to choose?
And why?

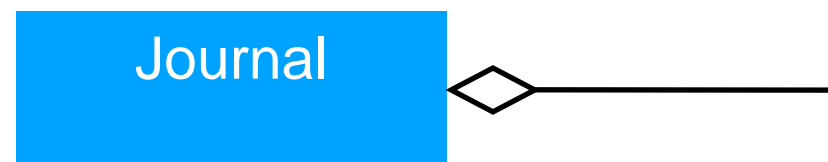
There are two types of **Books**:

- (conference) **Proceedings** (consisting of **ConferencePapers**)
- **Compendium** (consisting of **CompendiumChapters**)

2–12 Draw a class diagram for bibliographic references. Use the references in Appendix C, Bibliography, to test your class diagram. Your class diagram should be as detailed as possible.

The domain of bibliographic references is rich and complex. Consequently, students should deepen their understanding of the domain before they attempt to draw a class diagram (similar to the requirements analysis of a system). Figure 2-8 depicts an incomplete sample solution that could be accepted as sufficient from an instructor. The class diagram should minimally include the following concepts:

- An abstract class *Publication* (or *BibliographicReference*)
 - A many to many relationship between *Author* and *Publication*
 - At least three or more concrete classes refining *Publication*
 - At least one aggregation relationship (e.g., between Journal and Article or Proceedings and ConferencePaper). Both ends of the aggregation should also be subclasses of *Publication*.
- a **Journal** consists of **Articles**
 - a **Proceedings** (Book) consists of **ConferencePapers**



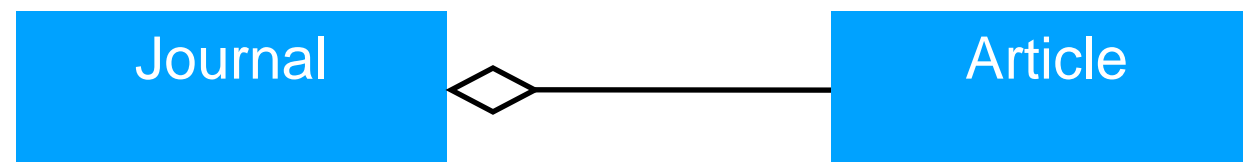
There are two types of **Books**:

- (conference) **Proceedings** (consisting of **ConferencePapers**)
- **Compendium** (consisting of **CompendiumChapters**)

2–12 Draw a class diagram for bibliographic references. Use the references in Appendix C, Bibliography, to test your class diagram. Your class diagram should be as detailed as possible.

The domain of bibliographic references is rich and complex. Consequently, students should deepen their understanding of the domain before they attempt to draw a class diagram (similar to the requirements analysis of a system). Figure 2-8 depicts an incomplete sample solution that could be accepted as sufficient from an instructor. The class diagram should minimally include the following concepts:

- An abstract class *Publication* (or *BibliographicReference*)
 - A many to many relationship between *Author* and *Publication*
 - At least three or more concrete classes refining *Publication*
 - At least one aggregation relationship (e.g., between Journal and Article or Proceedings and ConferencePaper). Both ends of the aggregation should also be subclasses of *Publication*.
- a **Journal** consists of **Articles**
 - a **Proceedings** (Book) consists of **ConferencePapers**



There are two types of **Books**:

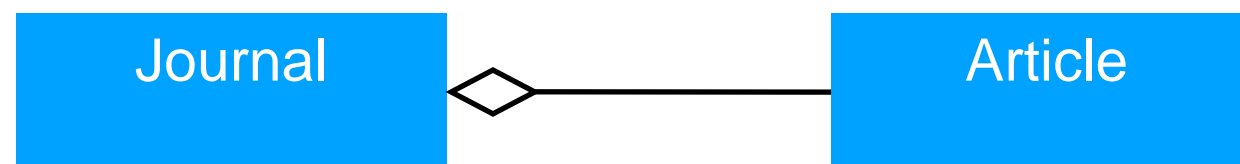
- (conference) **Proceedings** (consisting of **ConferencePapers**)
- **Compendium** (consisting of **CompendiumChapters**)

2–12 Draw a class diagram for bibliographic references. Use the references in Appendix C, Bibliography, to test your class diagram. Your class diagram should be as detailed as possible.

The domain of bibliographic references is rich and complex. Consequently, students should deepen their understanding of the domain before they attempt to draw a class diagram (similar to the requirements analysis of a system). Figure 2-8 depicts an incomplete sample solution that could be accepted as sufficient from an instructor. The class diagram should minimally include the following concepts:

- An abstract class *Publication* (or *BibliographicReference*)
- A many to many relationship between *Author* and *Publication*
- At least three or more concrete classes refining *Publication*
- At least one aggregation relationship (e.g., between Journal and Article or Proceedings and ConferencePaper). Both ends of the aggregation should also be subclasses of *Publication*.

- a **Journal** consists of **Articles**
- a **Proceedings** (Book) consists of **ConferencePapers**



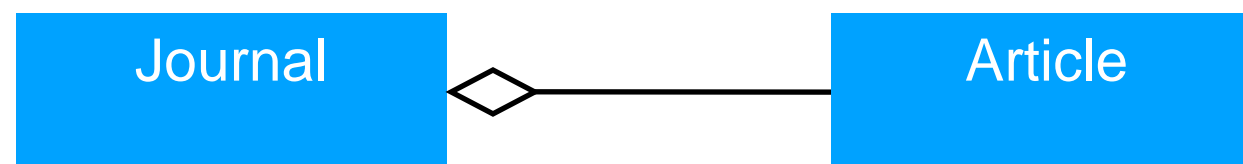
Poll

- many-to-many?
- one-to-many?
- many-to-one?

There are two types of **Books**:

- (conference) **Proceedings** (consisting of **ConferencePapers**)
- **Compendium** (consisting of **CompendiumChapters**)

- one
- many
- a **Journal** consists of **Articles**
 - a **Proceedings** (Book) consists of **ConferencePapers**

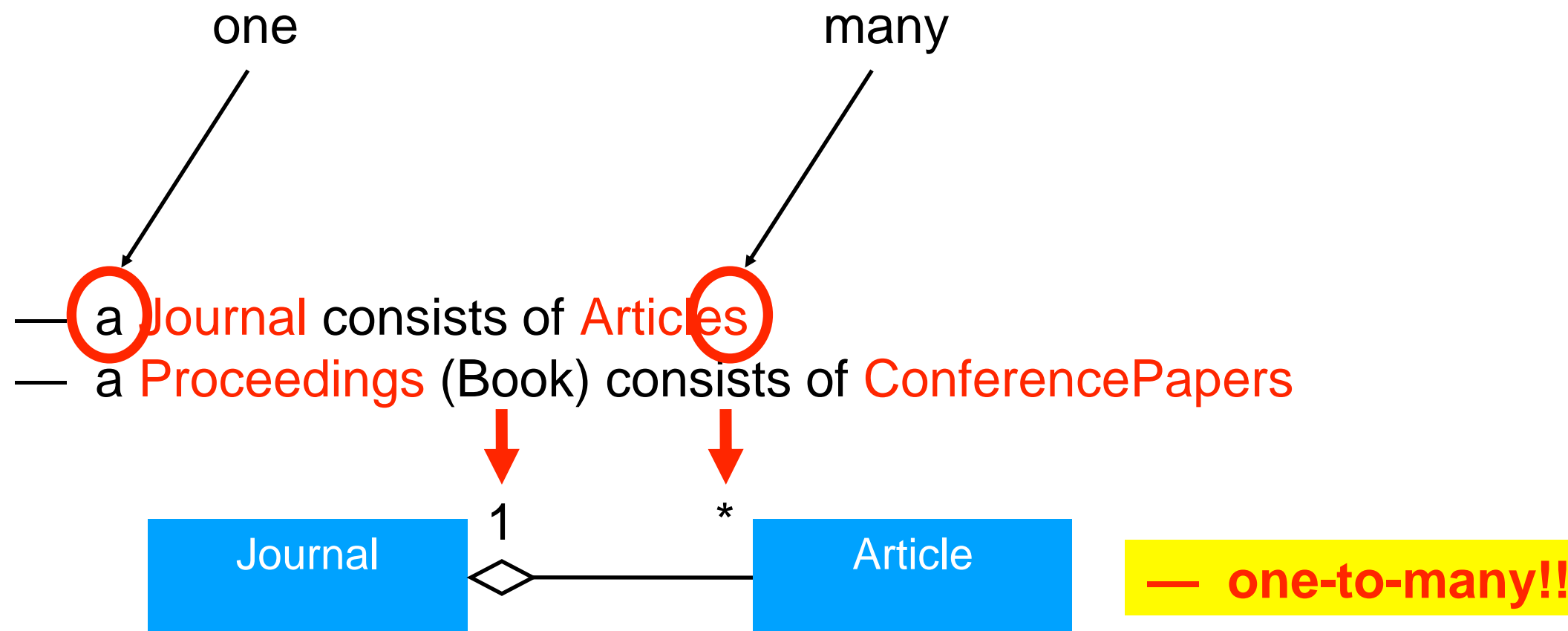


Poll

- many-to-many?
- **one-to-many!!**
- many-to-one?

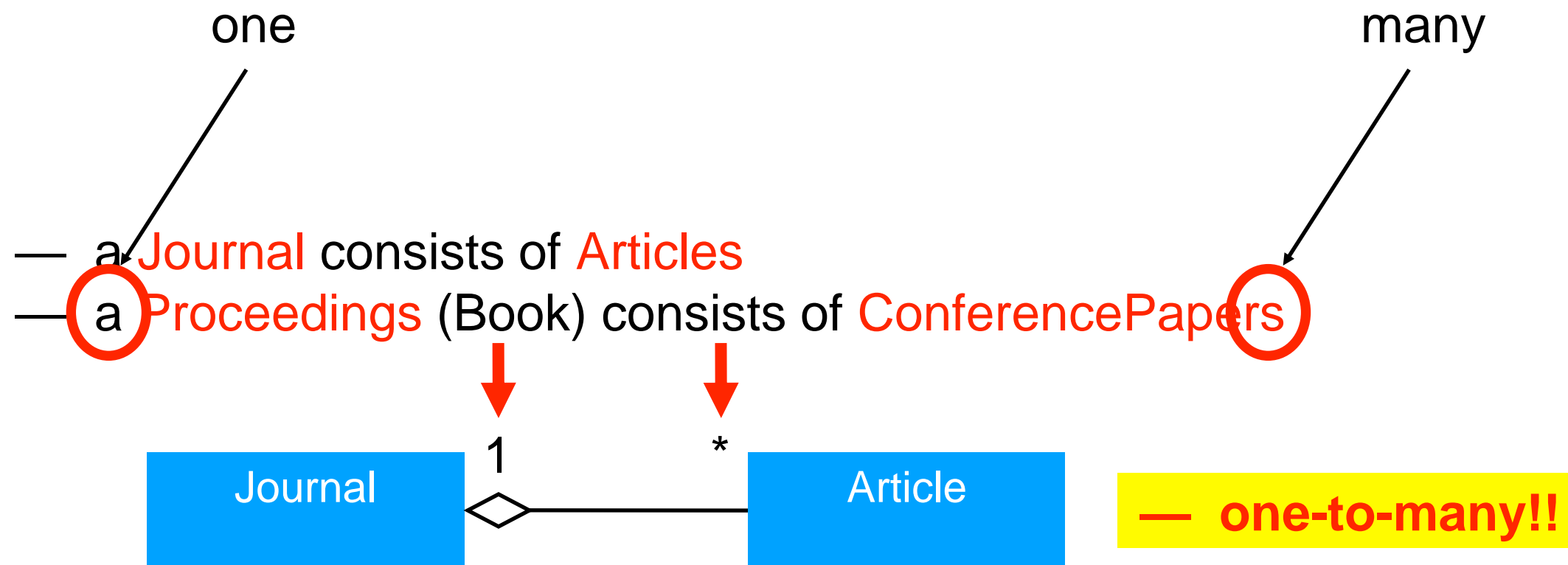
There are two types of **Books**:

- (conference) **Proceedings** (consisting of **ConferencePapers**)
- **Compendium** (consisting of **CompendiumChapters**)



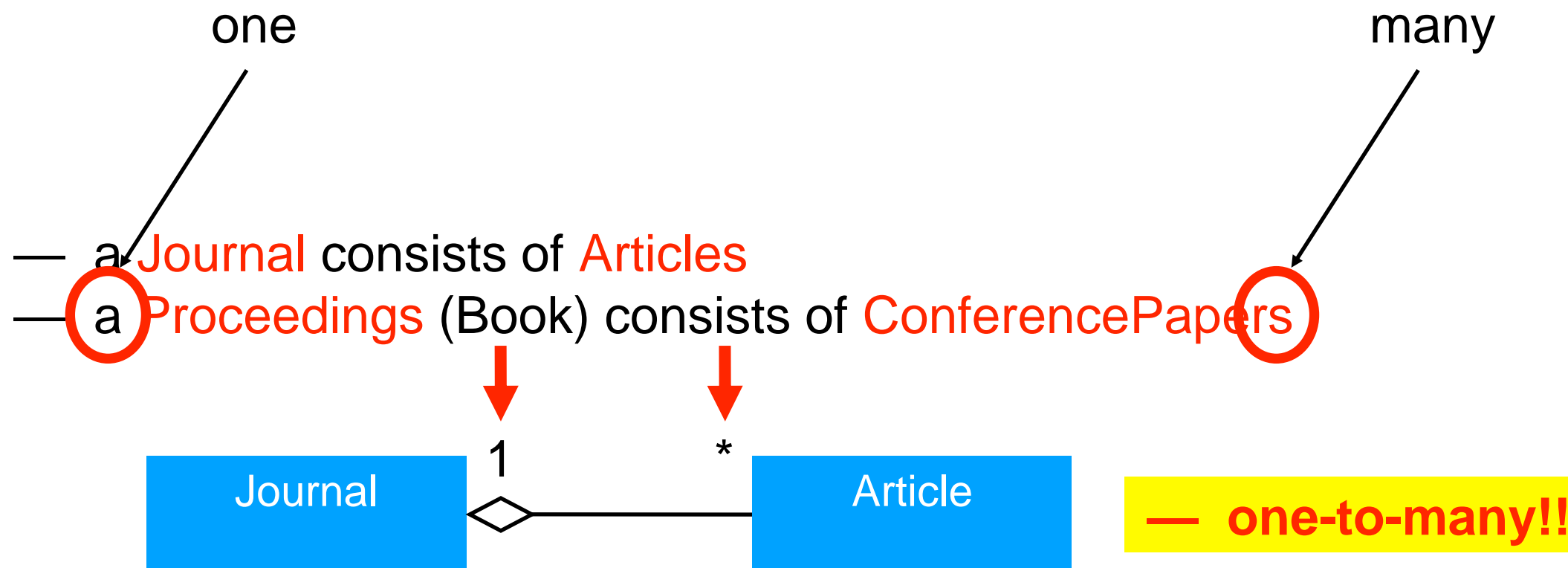
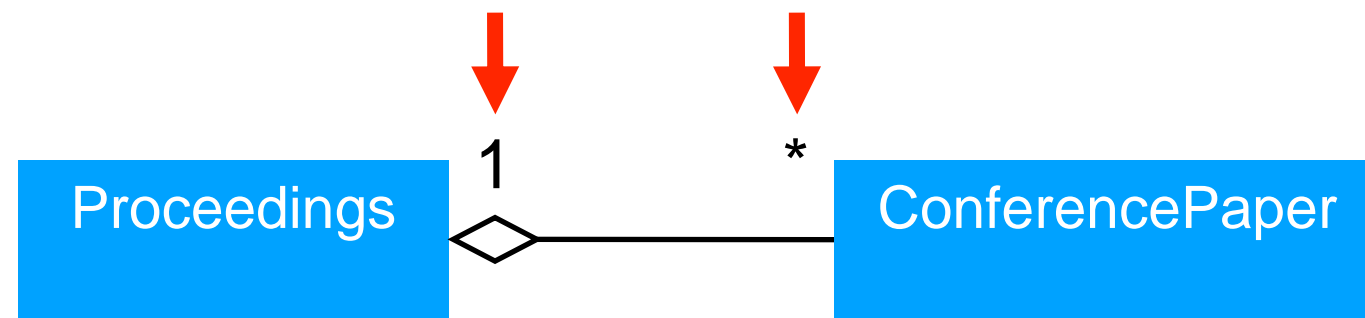
There are two types of Books:

- (conference) Proceedings (consisting of ConferencePapers)
- Compendium (consisting of CompendiumChapters)



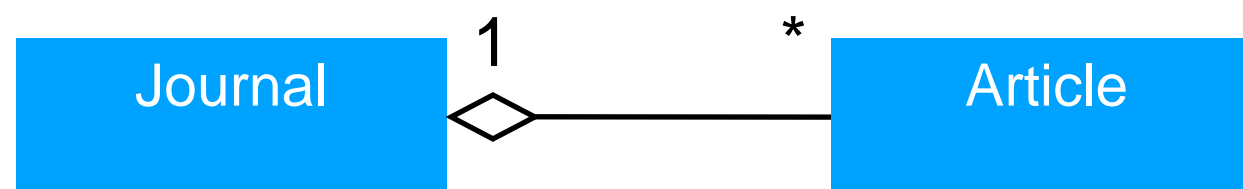
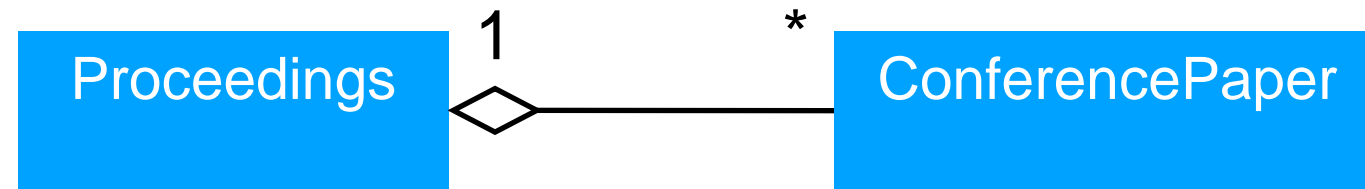
There are two types of Books:

- (conference) Proceedings (consisting of ConferencePapers)
- Compendium (consisting of CompendiumChapters)



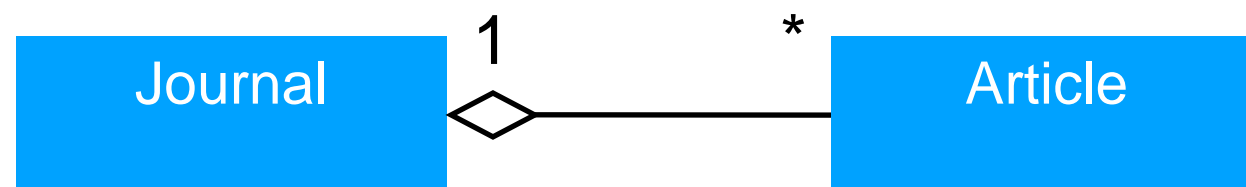
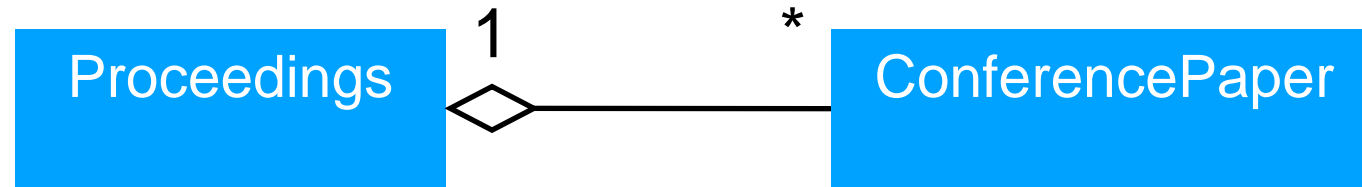
There are two types of **Books**:

- (conference) **Proceedings** (consisting of **ConferencePapers**)
- **Compendium** (consisting of **CompendiumChapters**)



There are two types of **Books**:

- (conference) **Proceedings** (consisting of **ConferencePapers**)
- **Compendium** (consisting of **CompendiumChapters**)

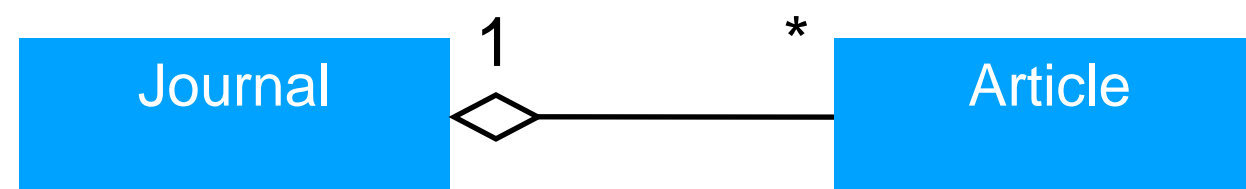
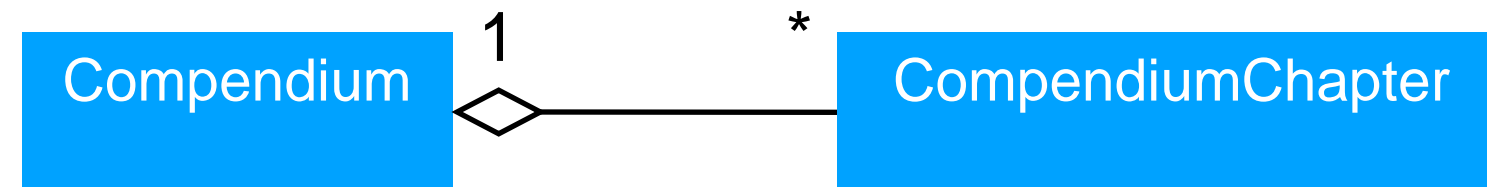
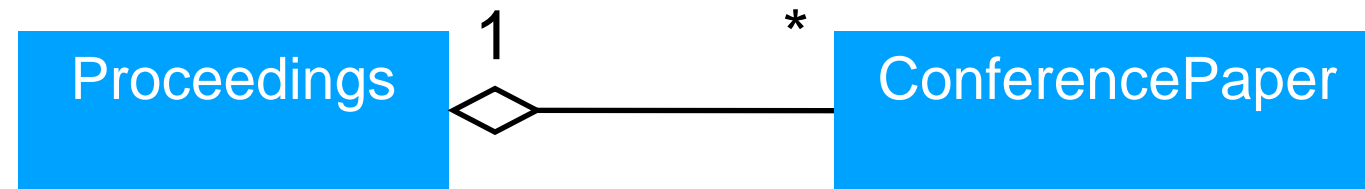


There are two types of **Books**:

- (conference) **Proceedings** (consisting of **ConferencePapers**)
- **Compendium** (consisting of **CompendiumChapters**)

many

one

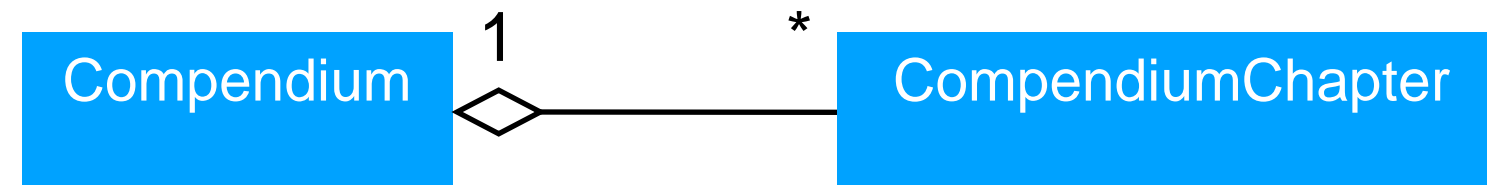
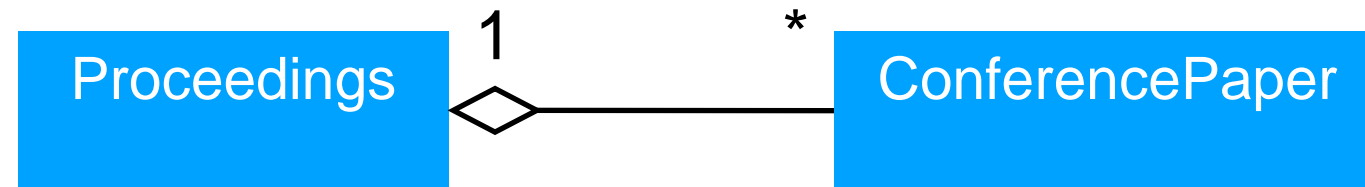


There are two types of **Books**:

- (conference) **Proceedings** (consisting of **ConferencePapers**)
- **Compendium** (consisting of **CompendiumChapters**)

many

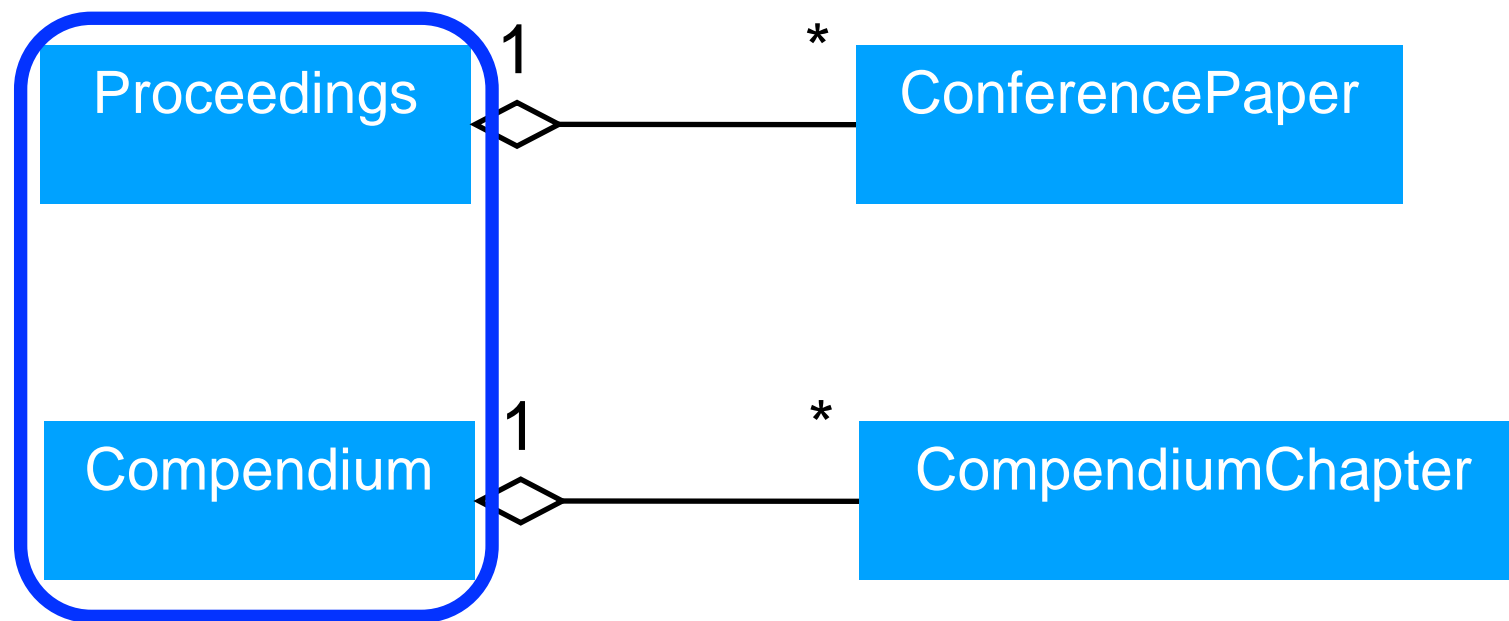
one



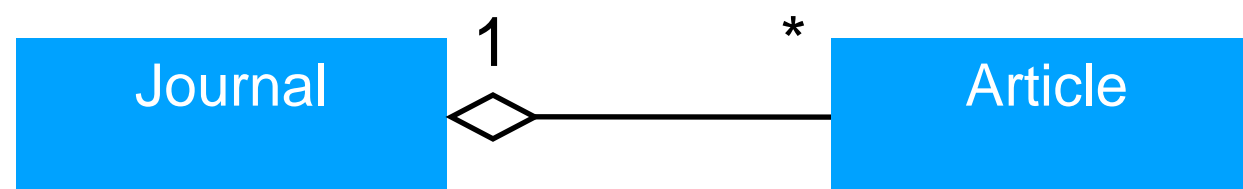
“two types of ...”
— specialization
— is subclass of

There are two types of **Books**:

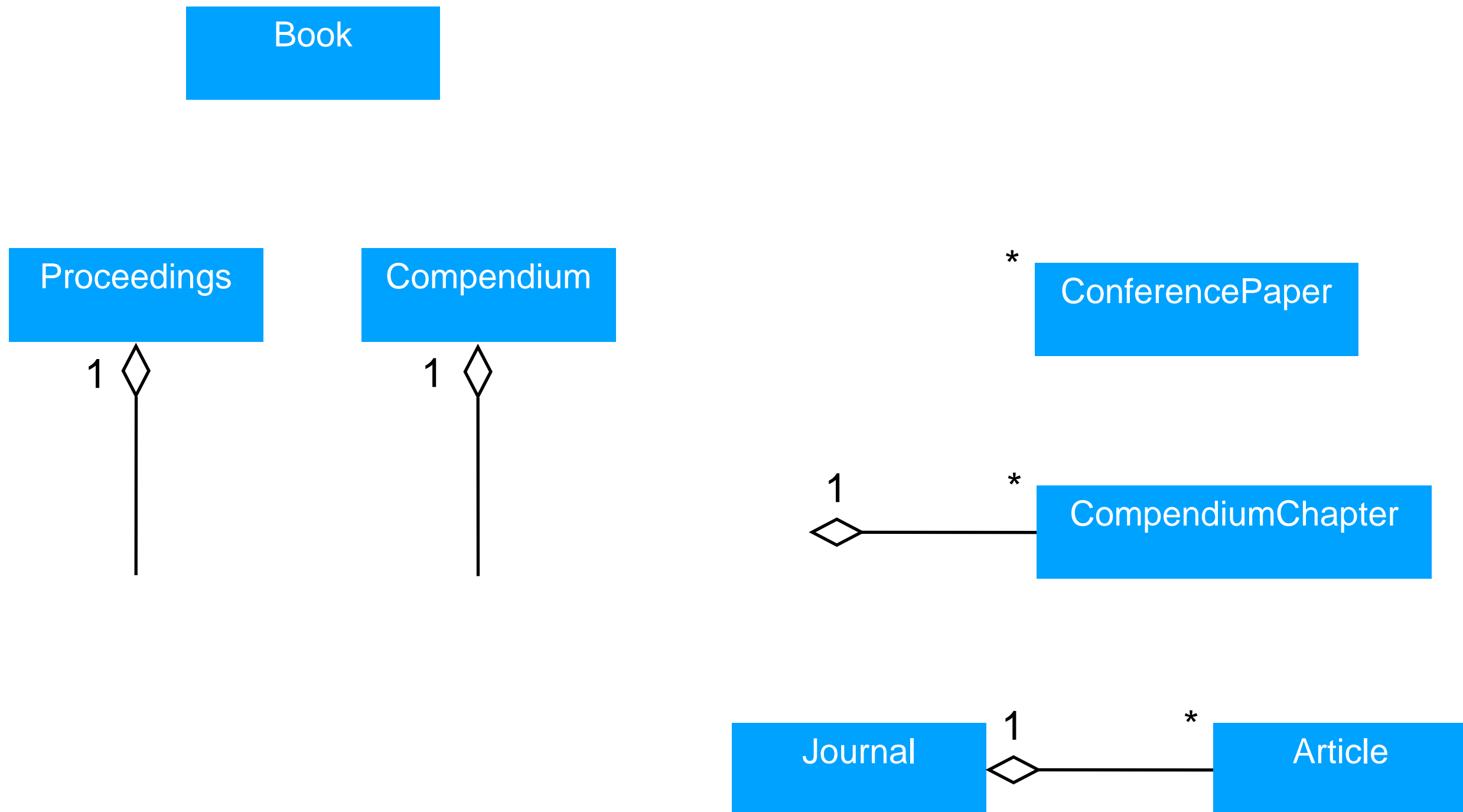
- (conference) **Proceedings** (consisting of **ConferencePapers**)
- **Compendium** (consisting of **CompendiumChapters**)

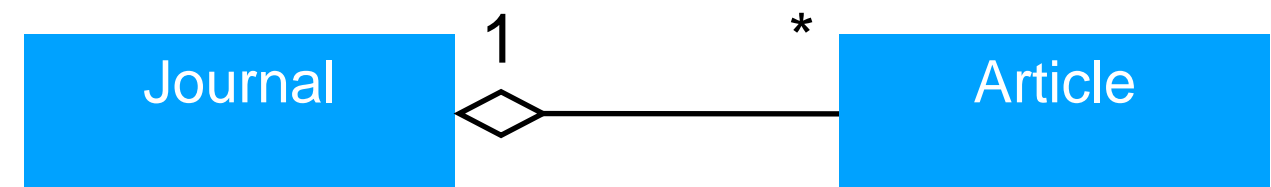
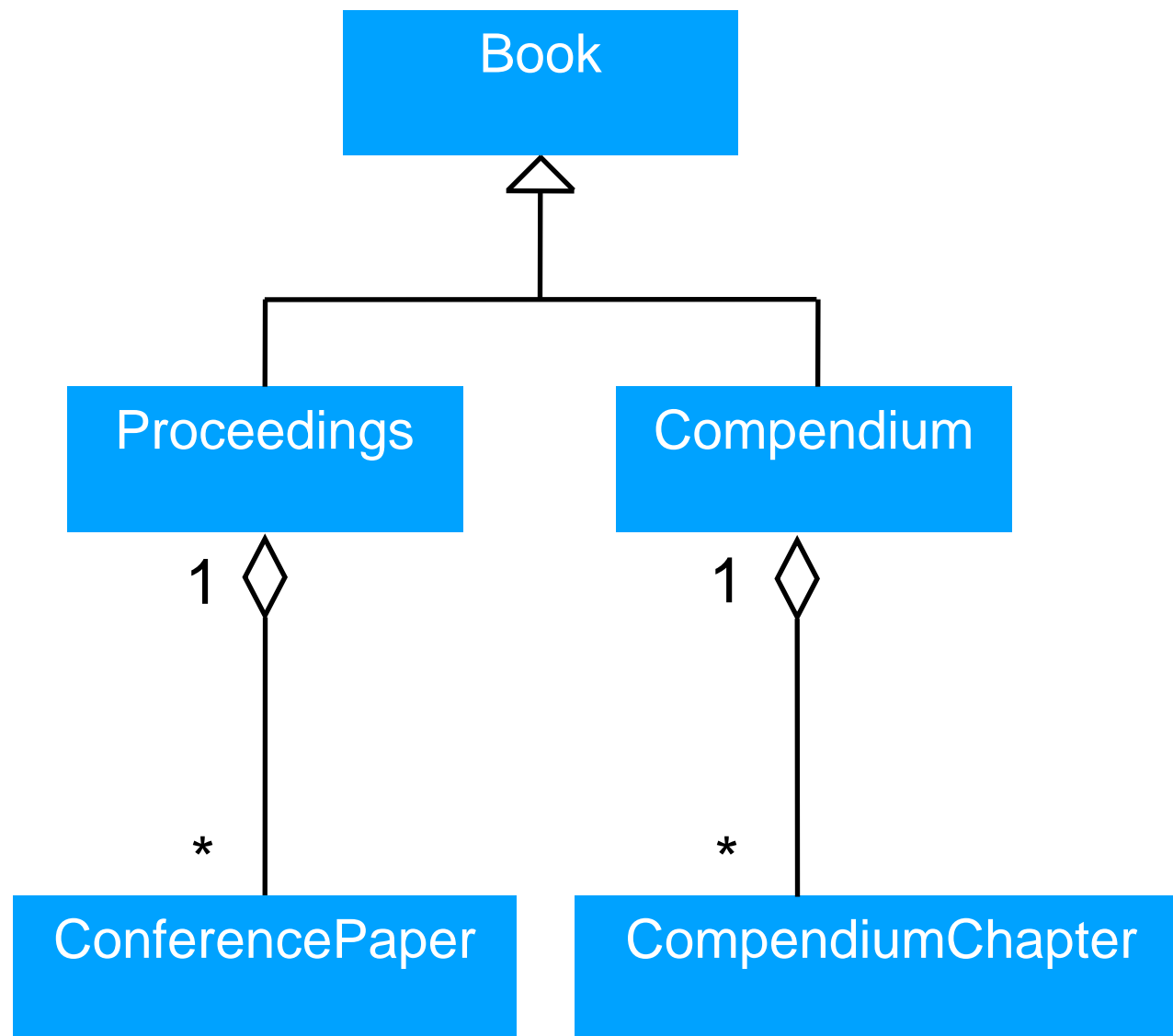


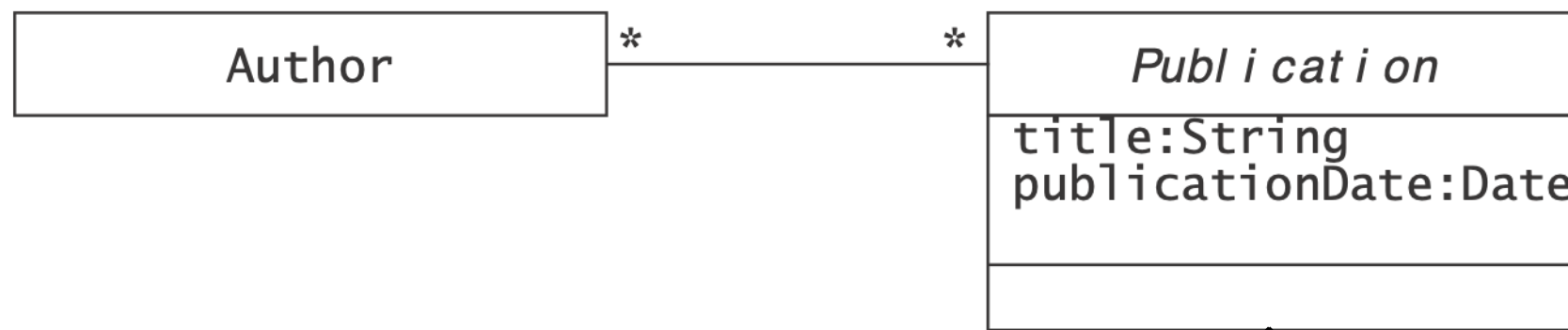
these are **subclasses**
of **class Book**



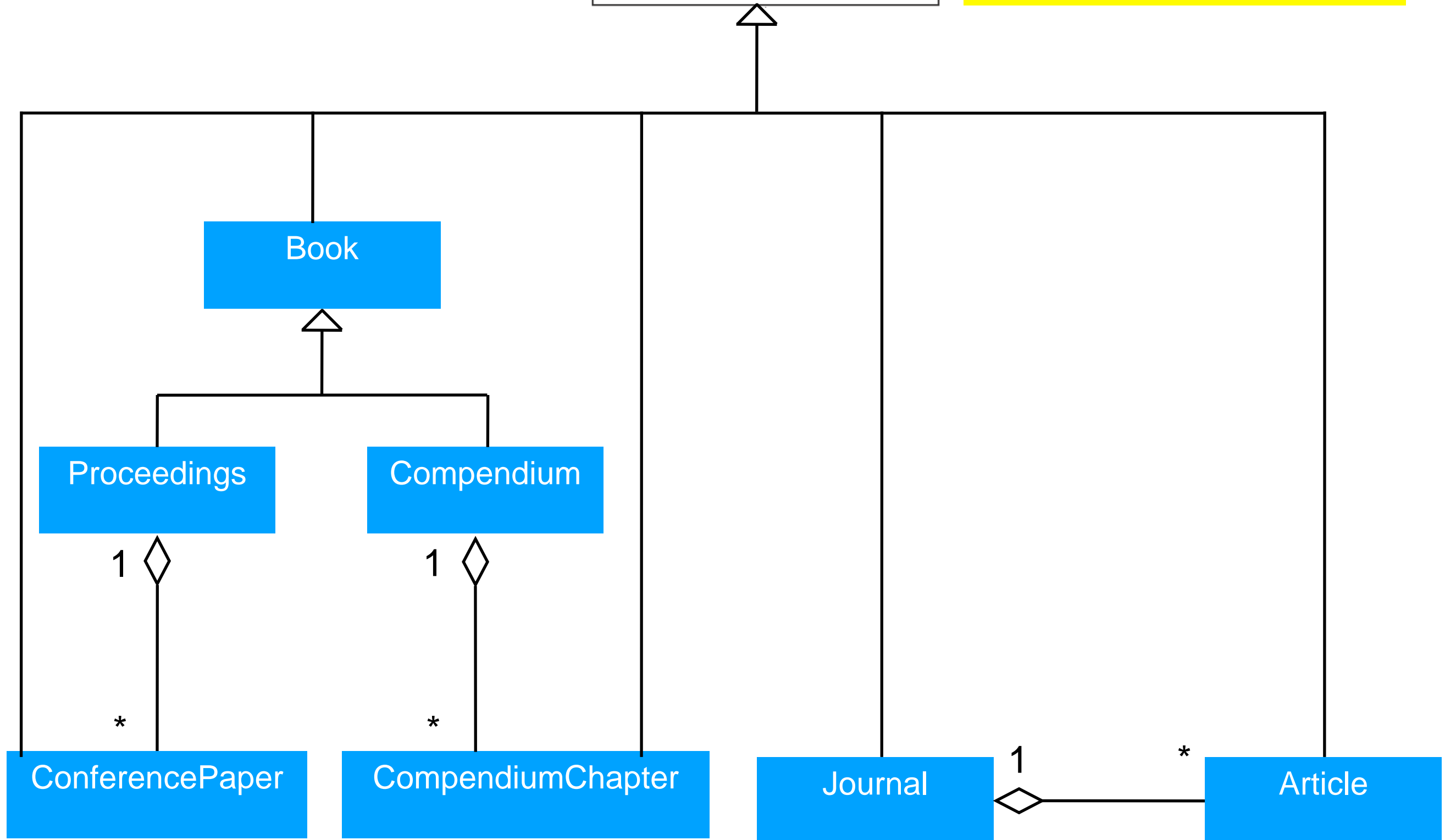
Remember: the (big white) arrow for **inheritance** points from the subclass to the superclass







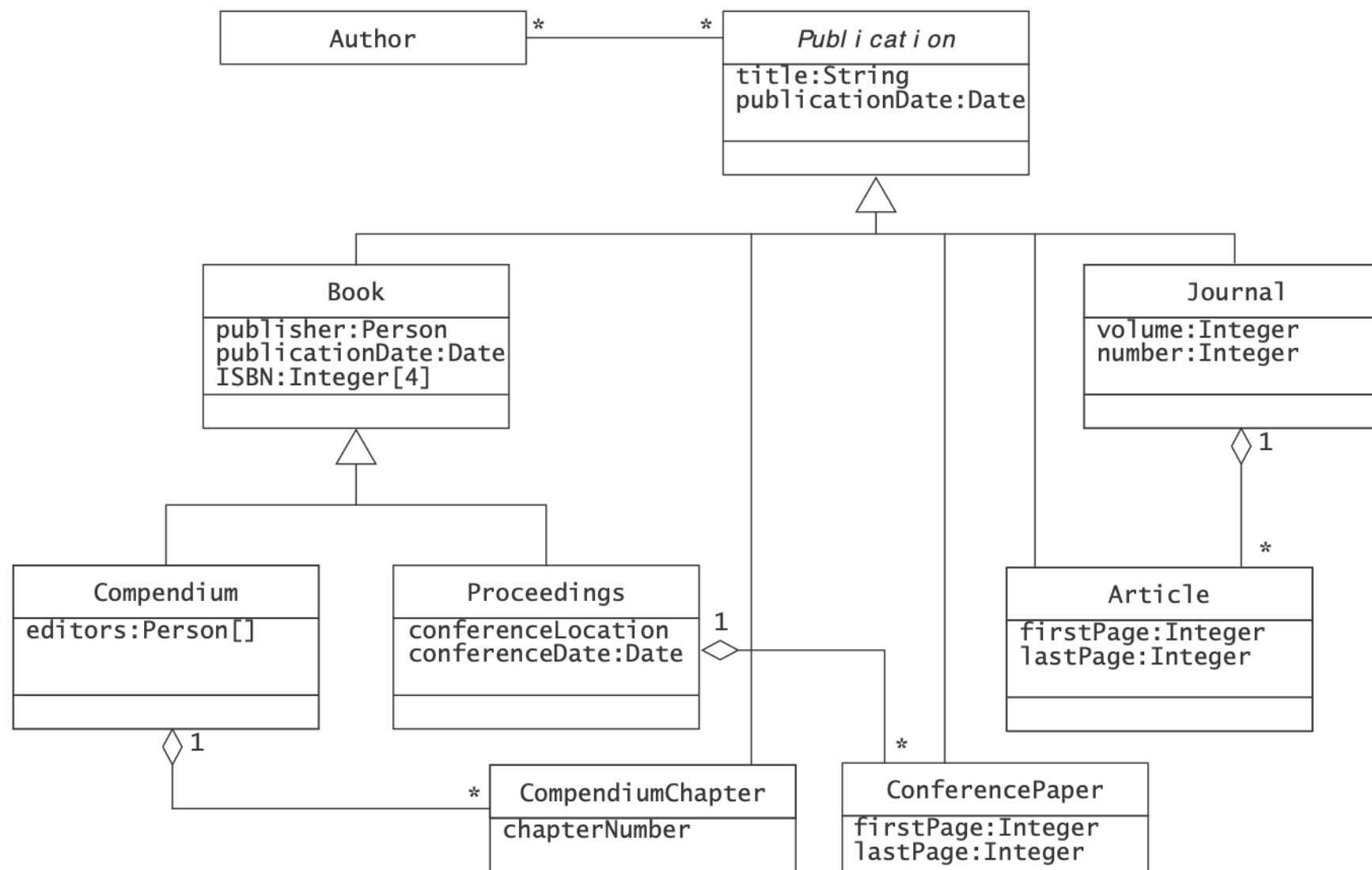
This is our *abstract class* of **any publication!**



2–12 Draw a class diagram for bibliographic references. Use the references in Appendix C, Bibliography, to test your class diagram. Your class diagram should be as detailed as possible.

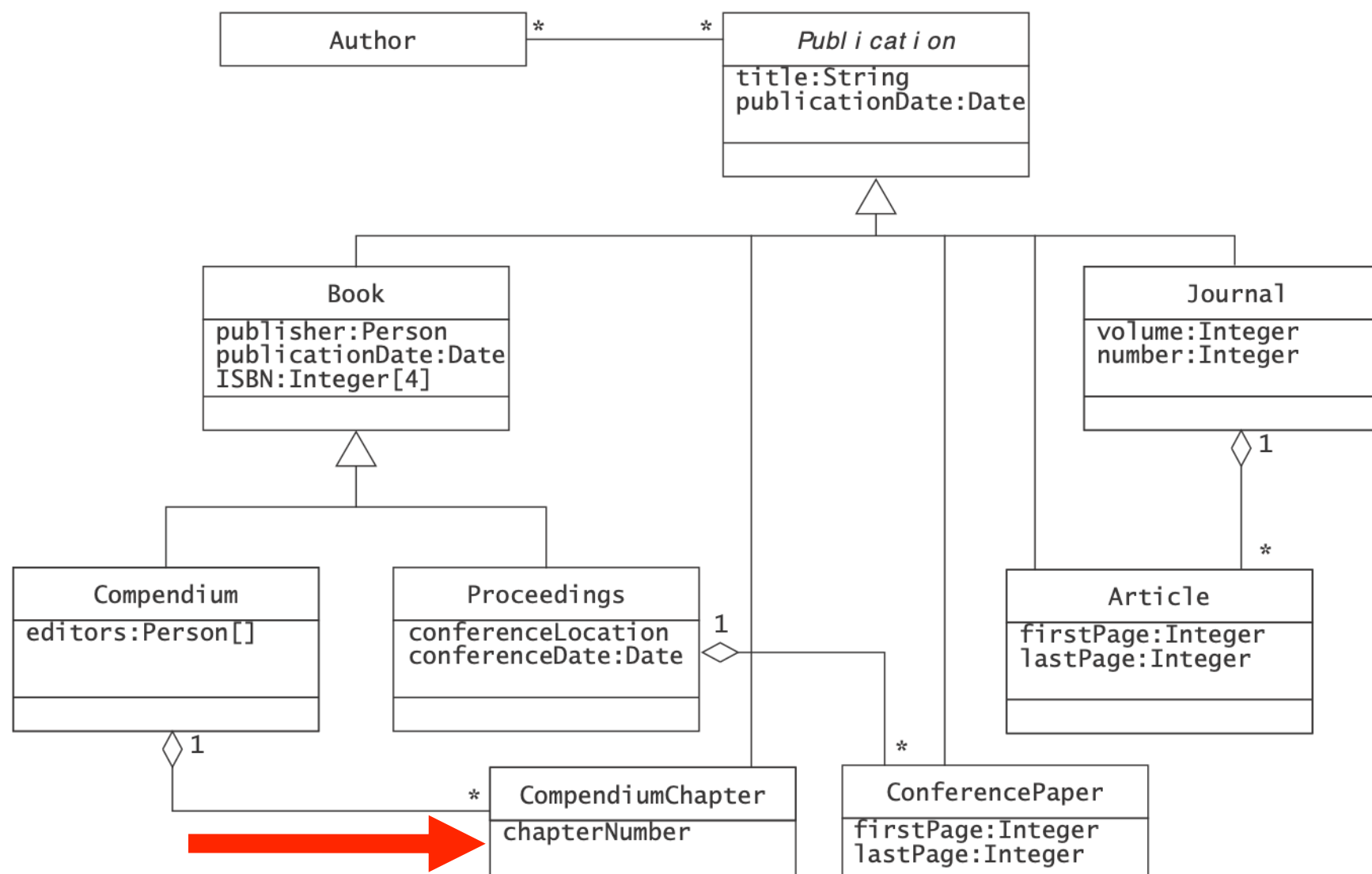
The domain of bibliographic references is rich and complex. Consequently, students should deepened their understanding of the domain before they attempt to draw a class diagram (similar to the requirements analysis of a system). Figure 2-8 depicts an incomplete sample solution that could be accepted as sufficient from an instructor. The class diagram should minimally include the following concepts:

- An abstract class *Publication* (or *BibliographicReference*)
- A many to many relationship between *Author* and *Publication*
- At least three or more concrete classes refining *Publication*
- At least one aggregation relationship (e.g., between *Journal* and *Article* or *Proceedings* and *ConferencePaper*). Both ends of the aggregation should also be subclasses of *Publication*.



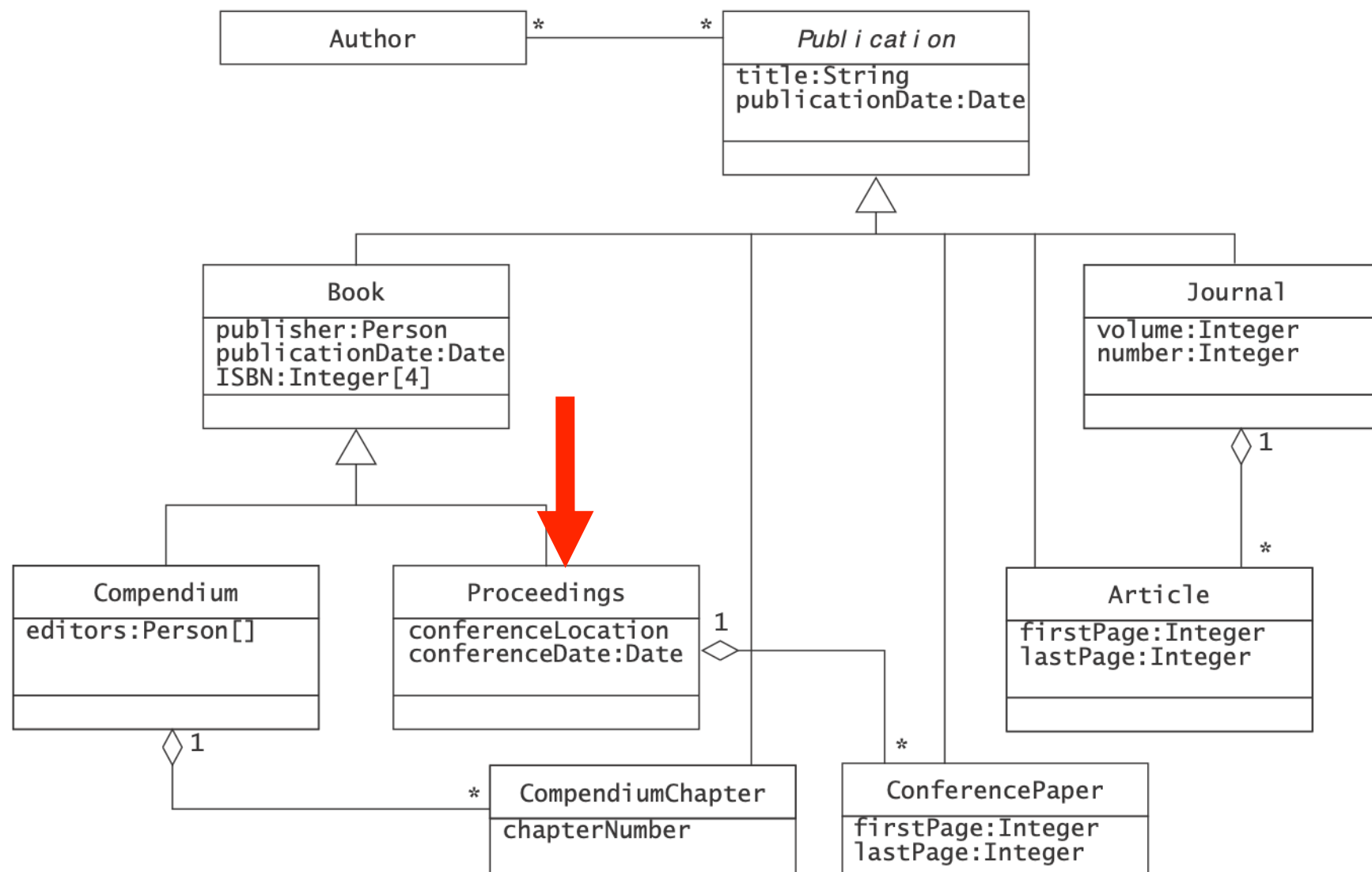
Peculiarities:

— why does `CompendiumChapter` not have `firstPage` / `lastPage`?



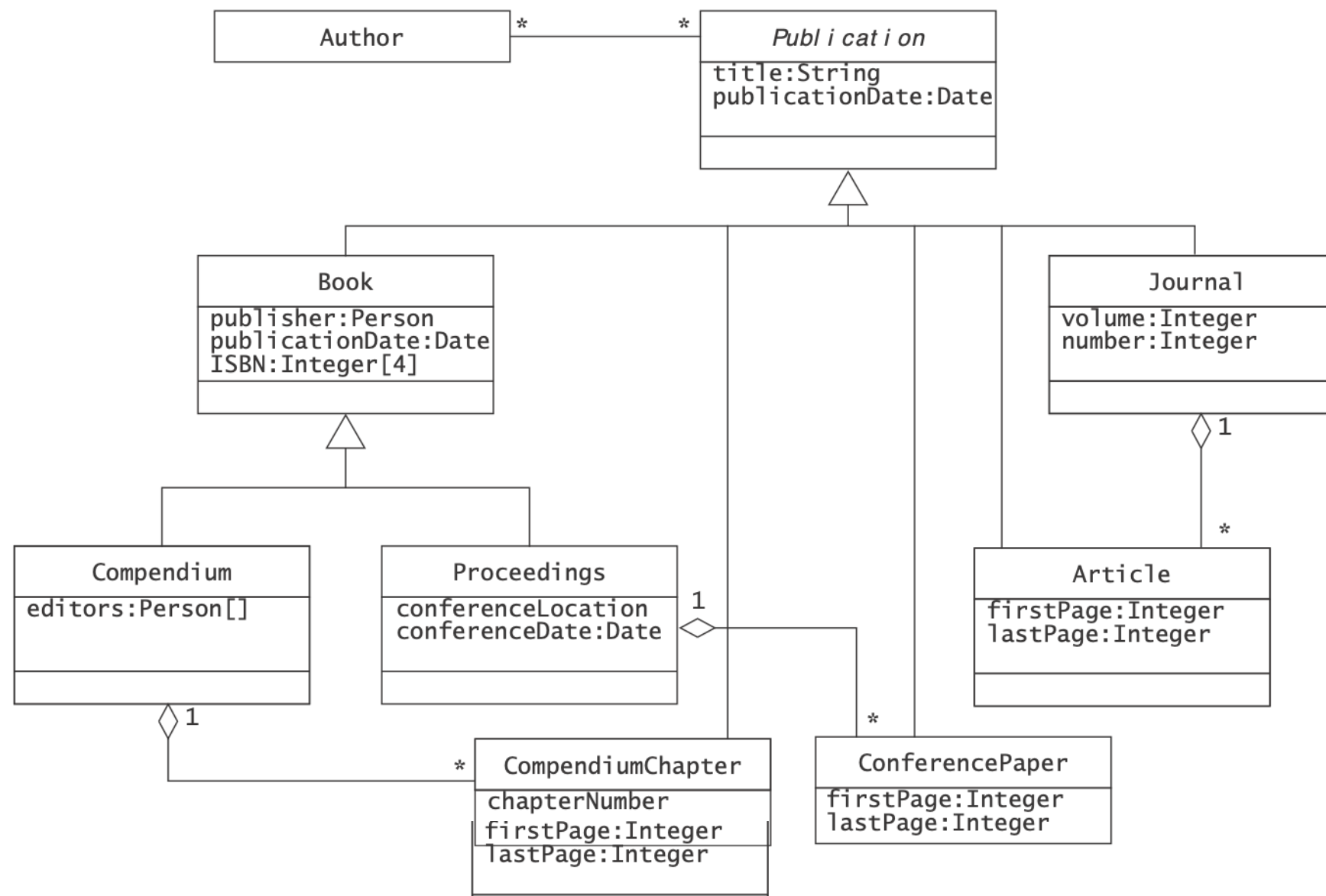
Peculiarities:

- why does `CompendiumChapter` not have `firstPage` / `lastPage`?
- why does `Proceedings` not have `editors`?



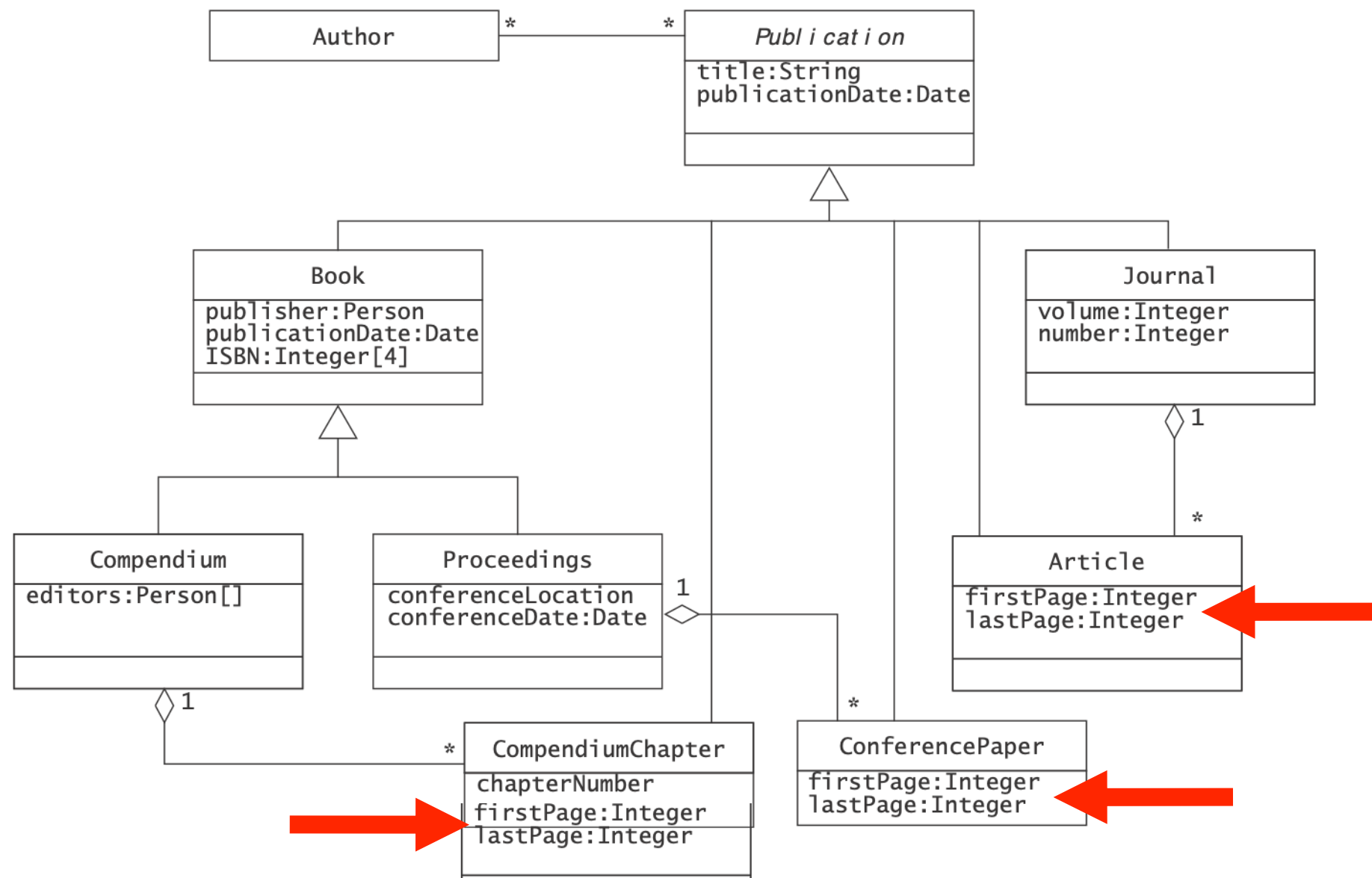
Peculiarities:

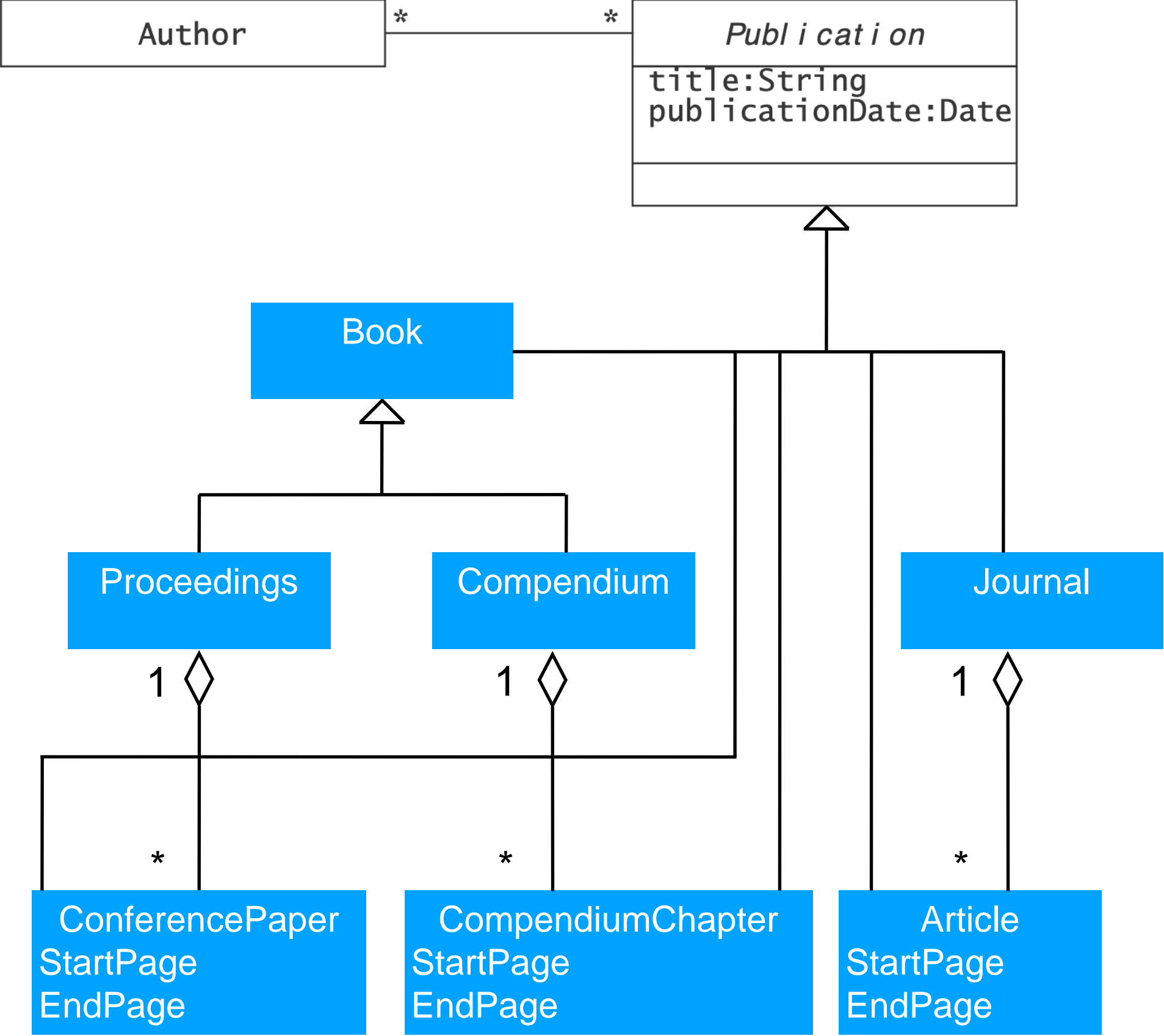
- why does CompendiumChapter not have firstPage / lastPage?
- why does **Proceedings** not have **editors**?

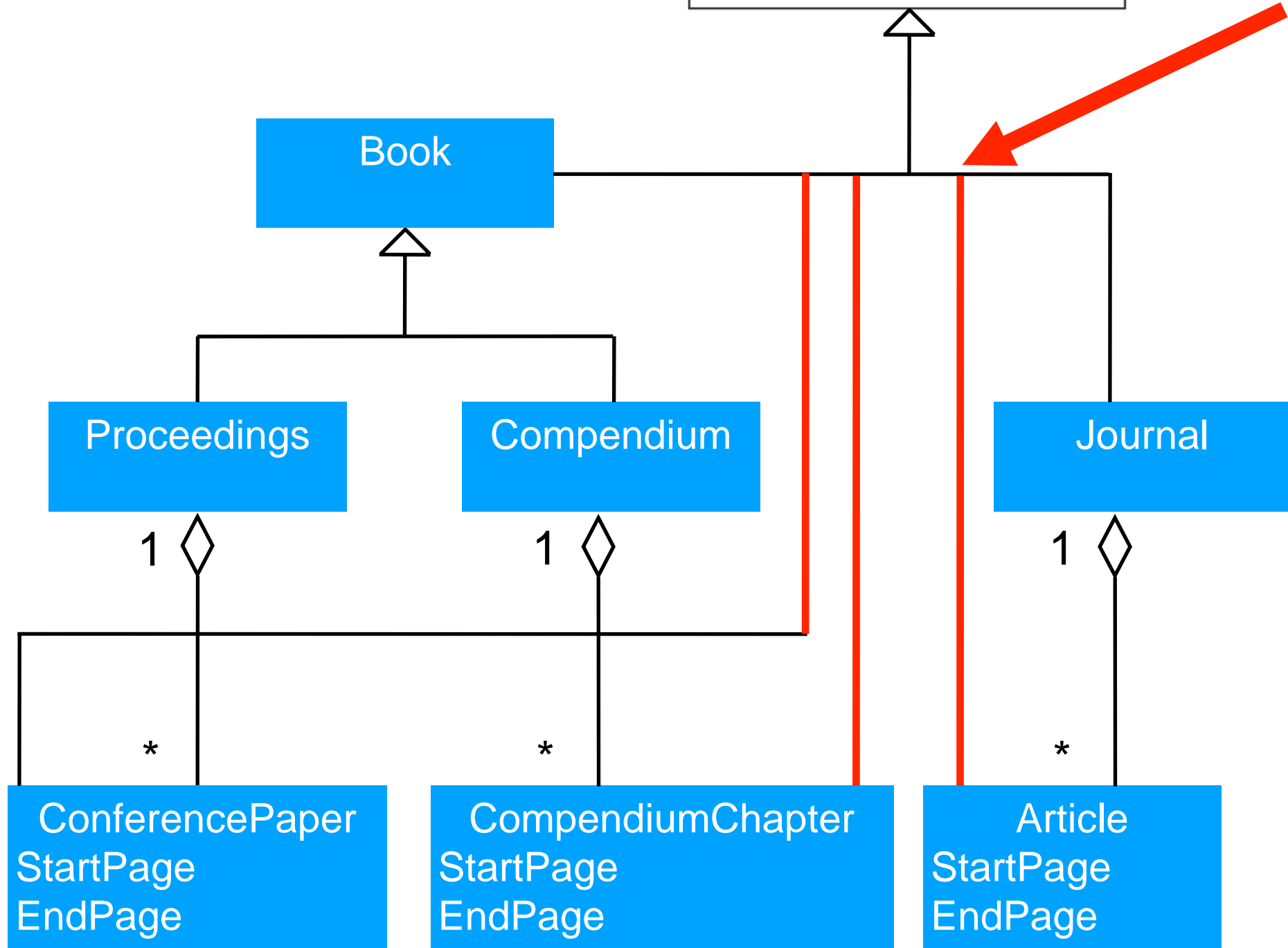
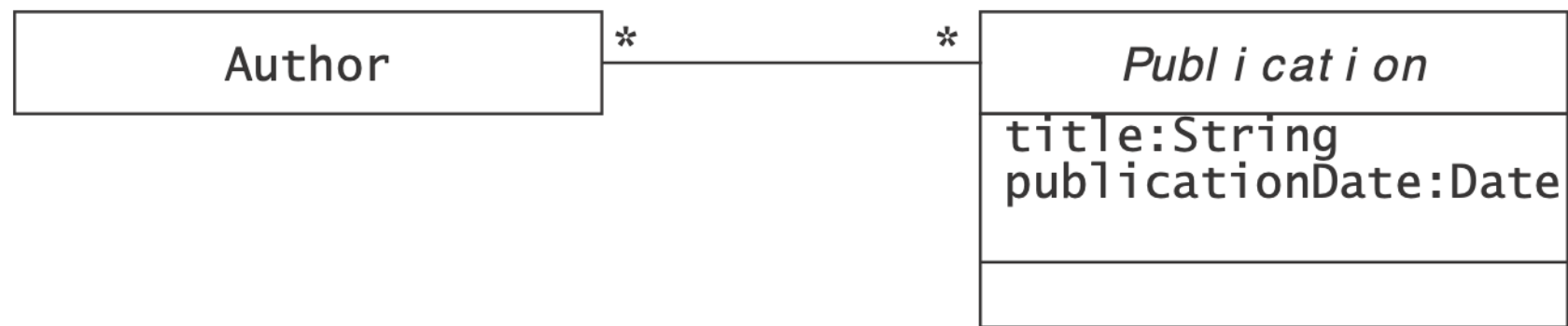


Question:

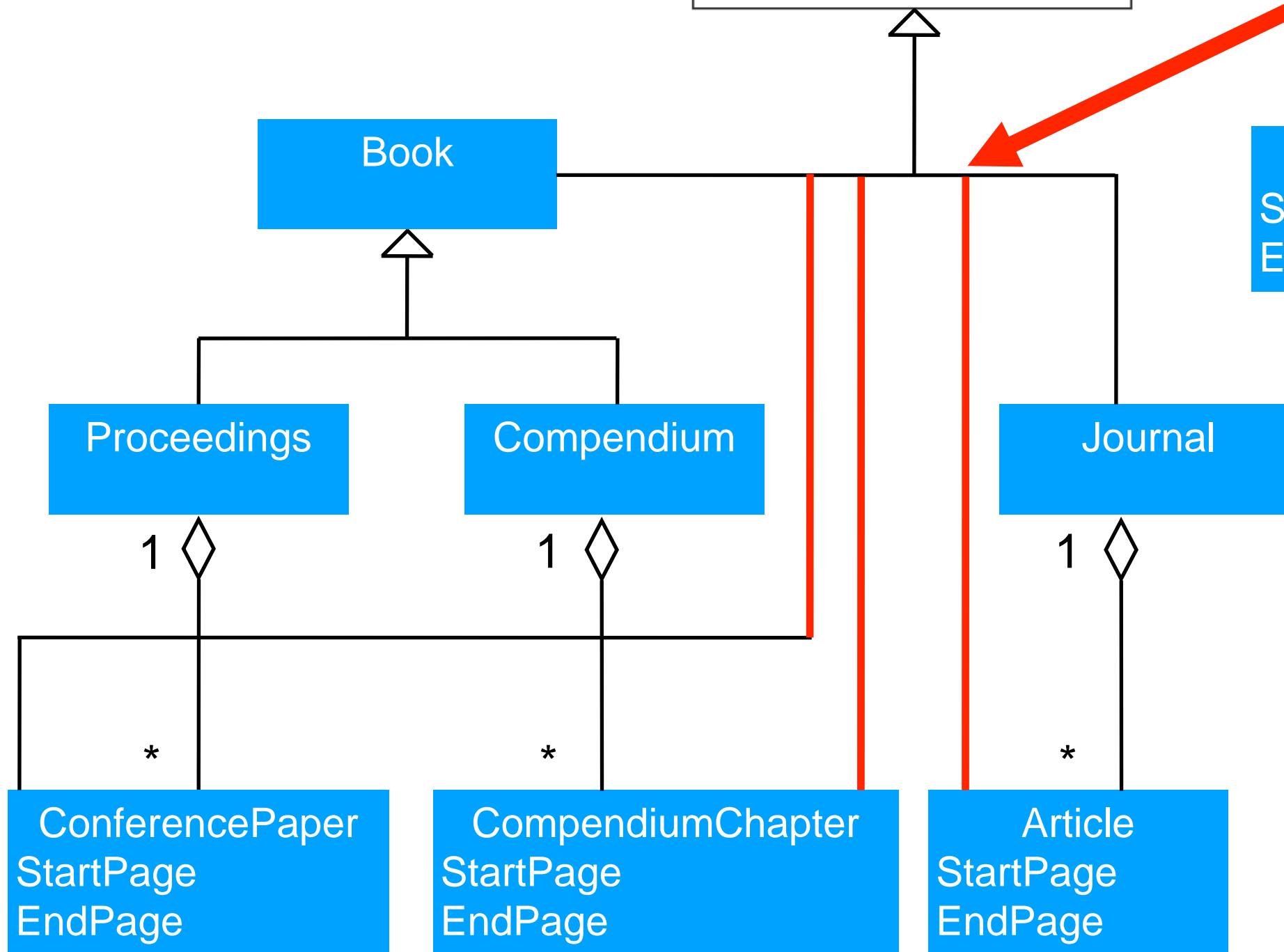
- how would you “abstract away” the common firstPage/lastPage attributes of CompendiumChapter, ConferencePaper and Article





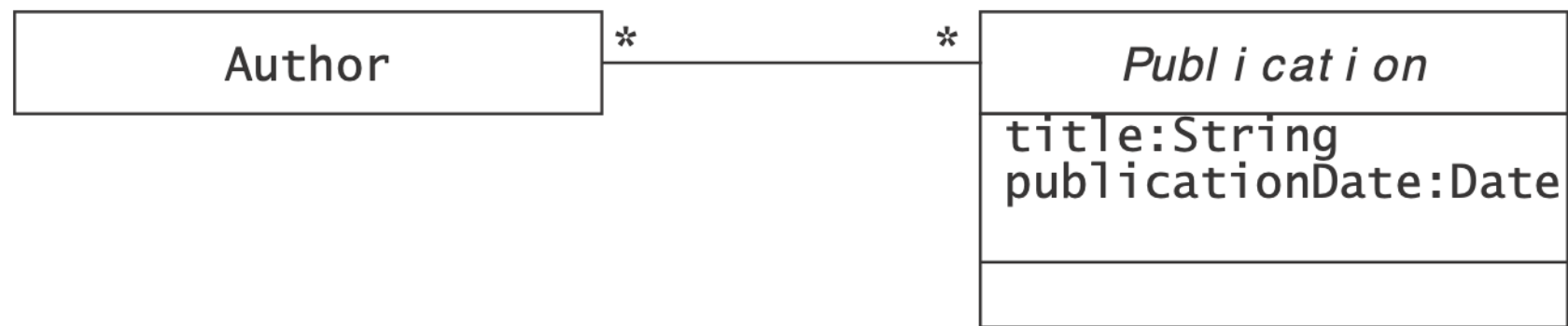


These three should go to a new abstract class!

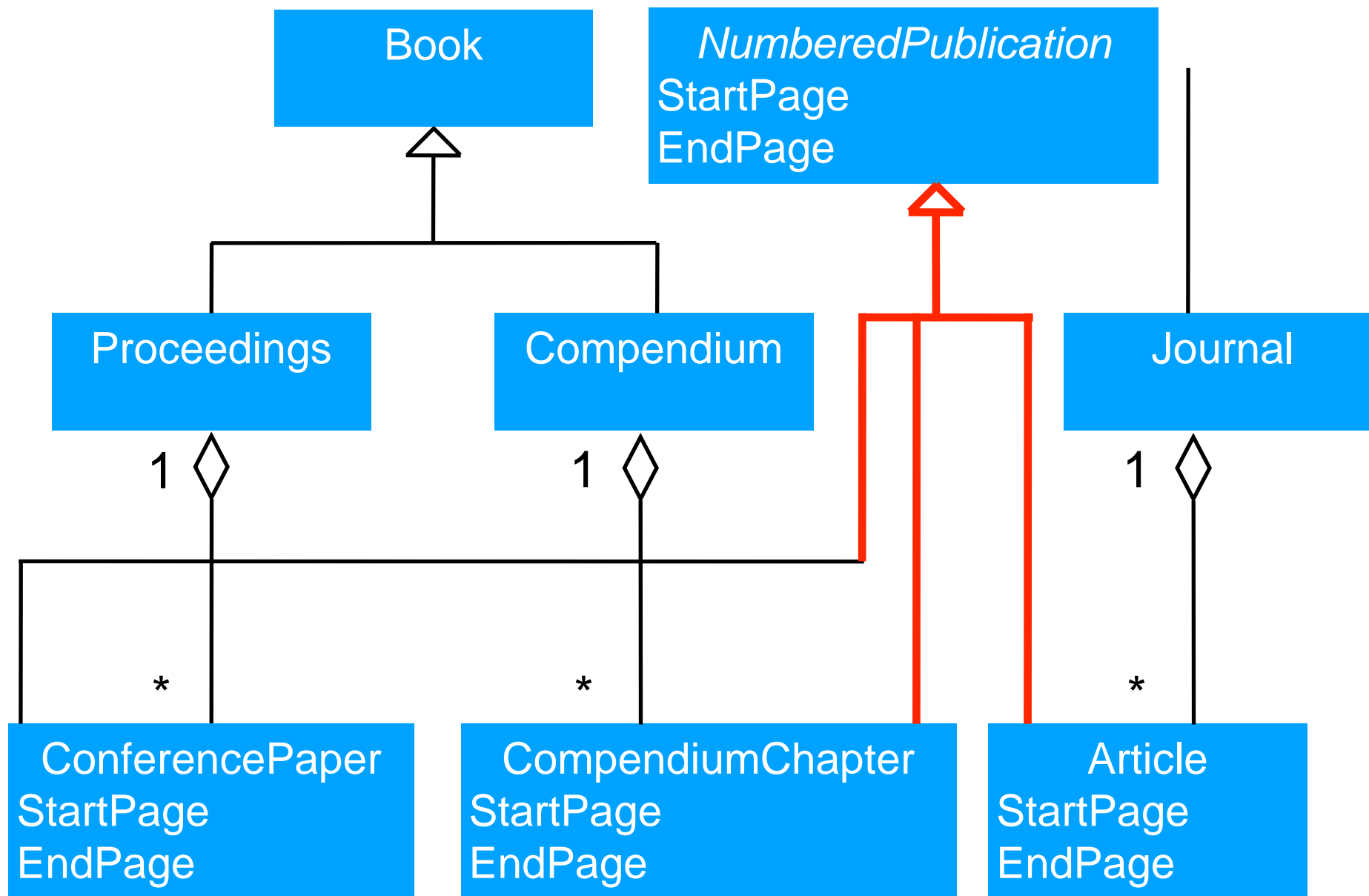


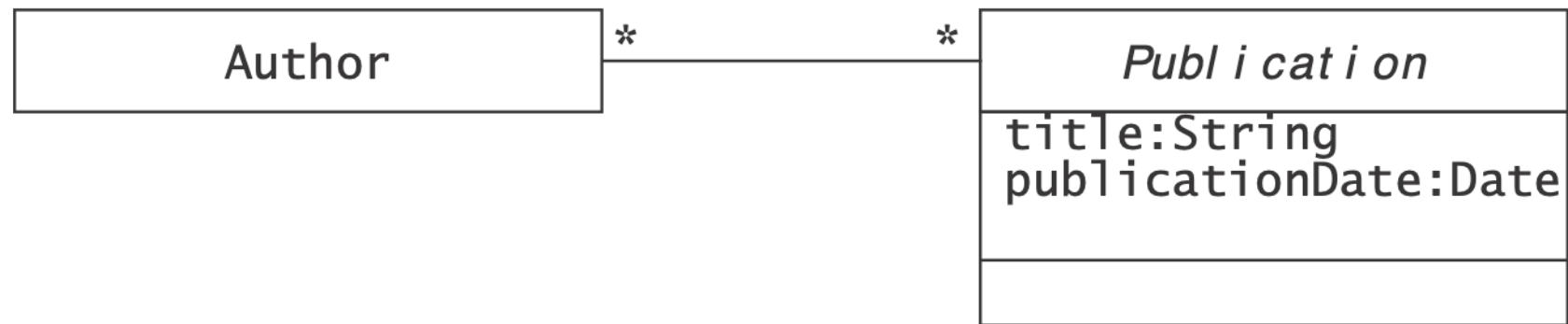
These three should go to a **new abstract class!**

NumberedPublication
StartPage
EndPage

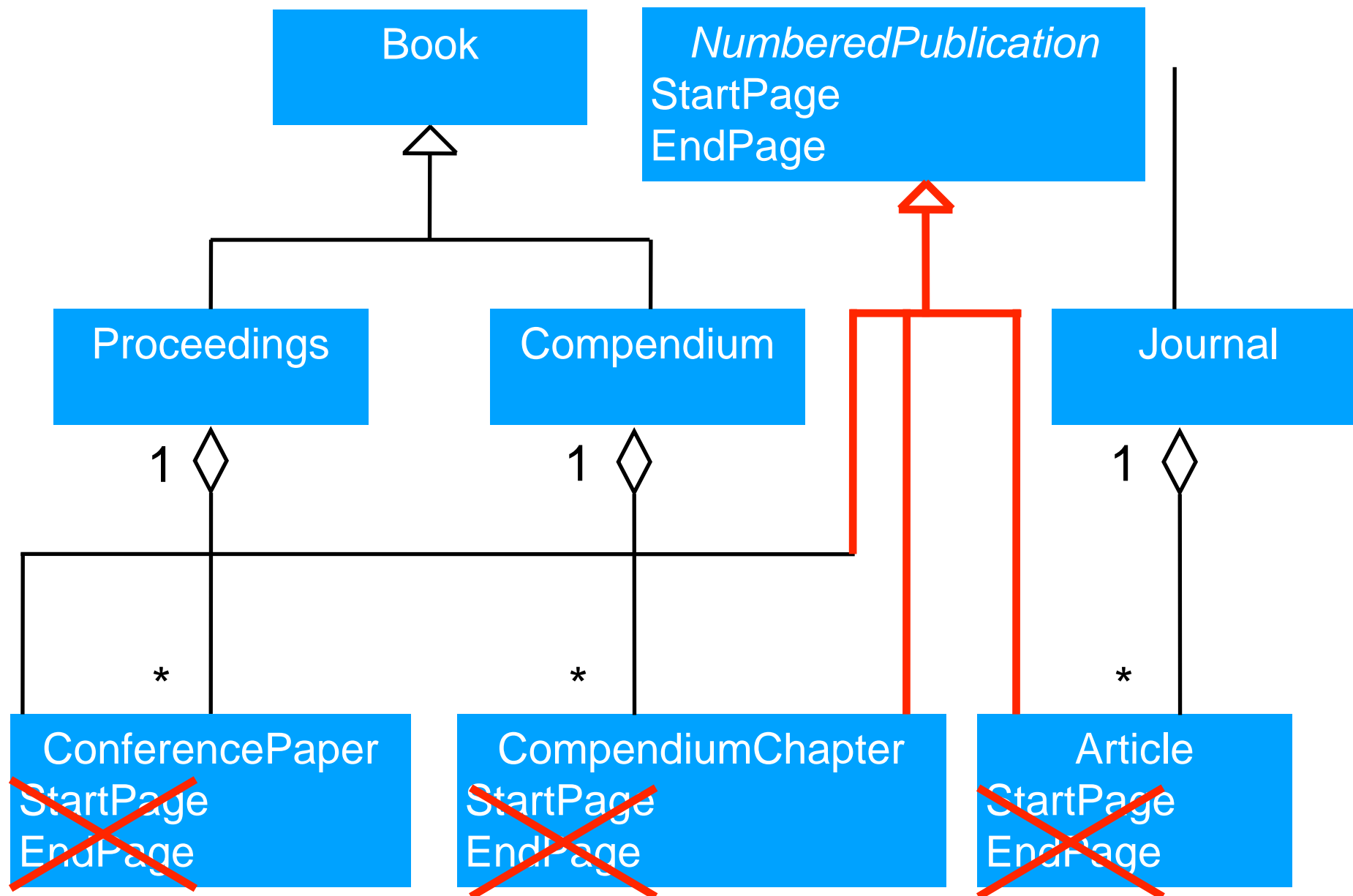


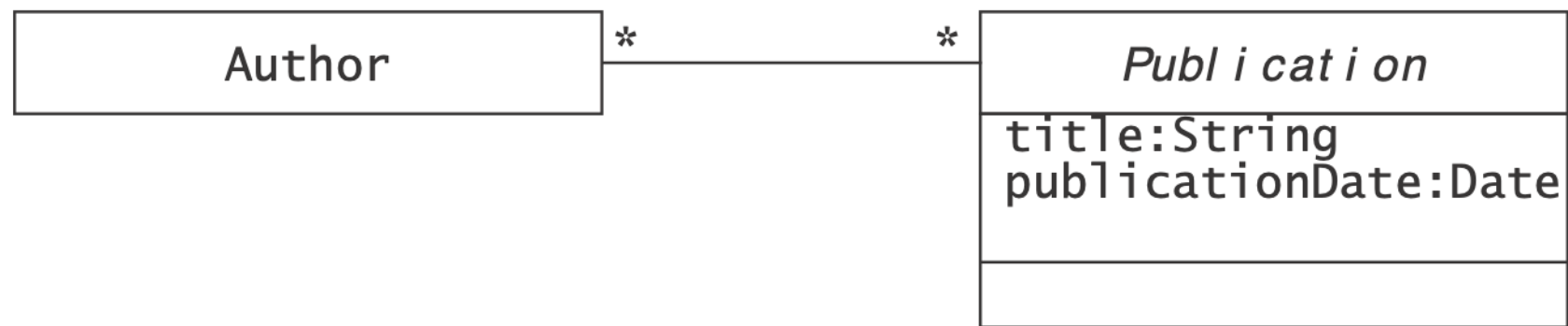
These three should go to a **new abstract class**!



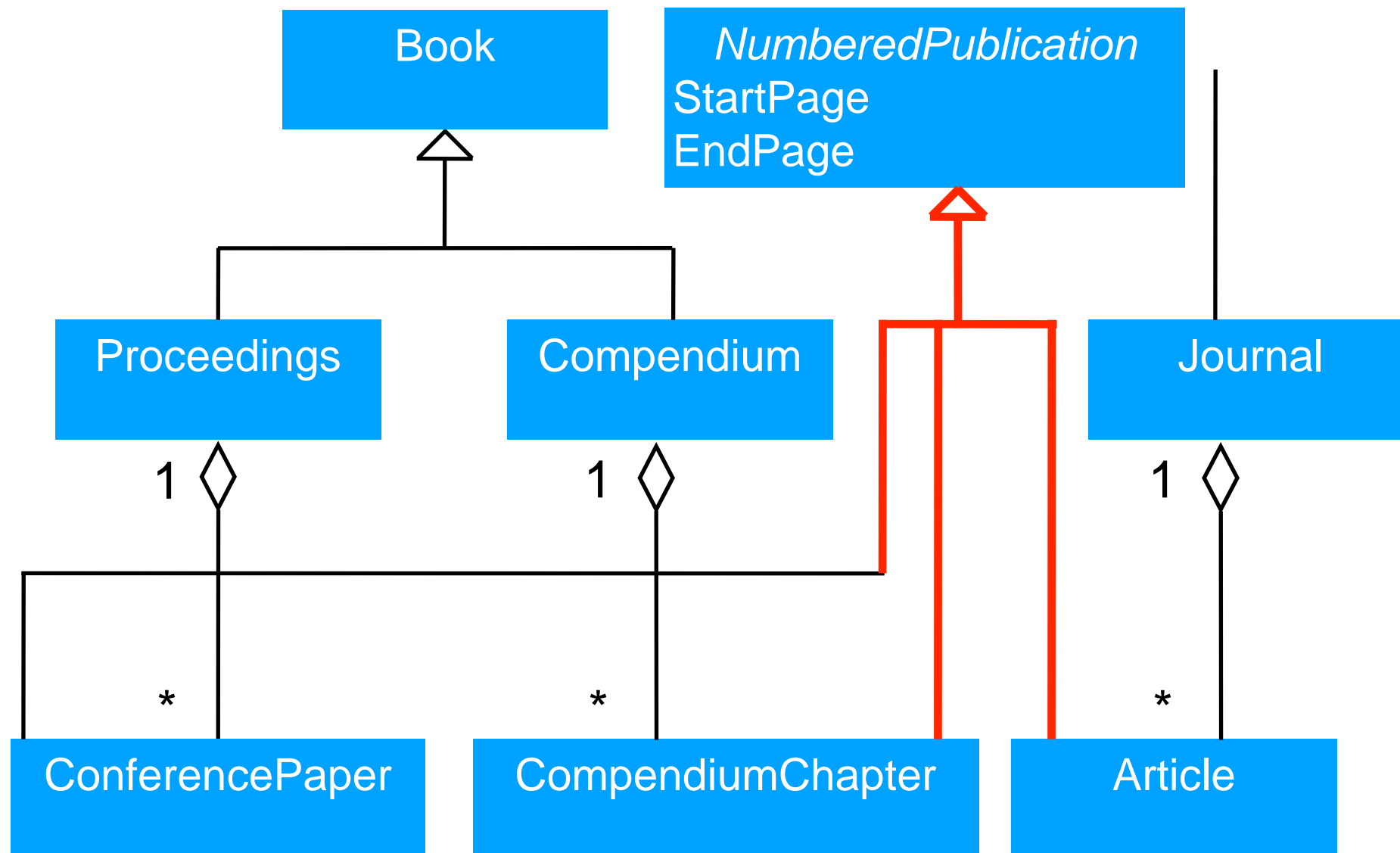


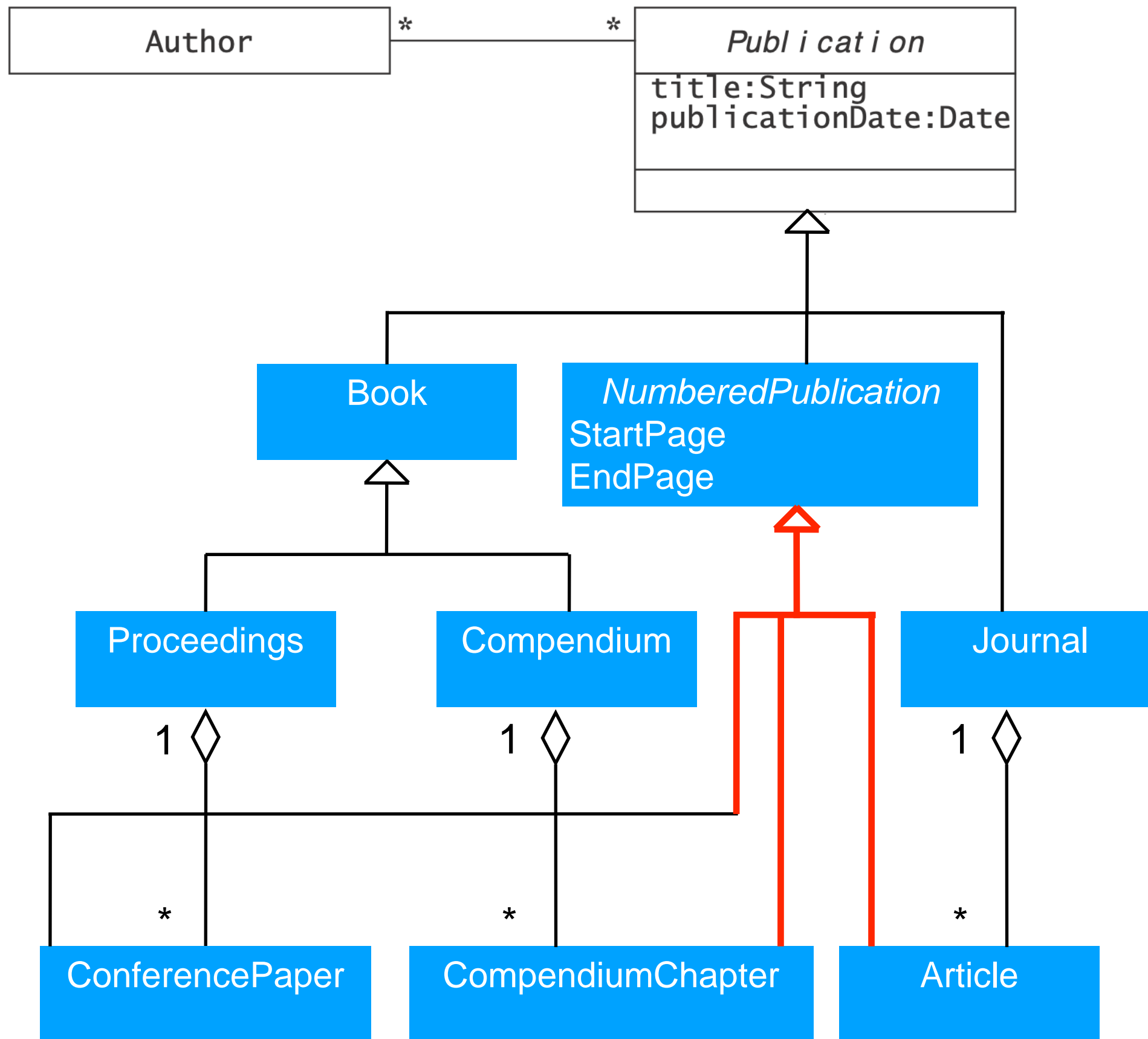
These three should go to a **new abstract class**!





These three should go to a **new abstract class**!

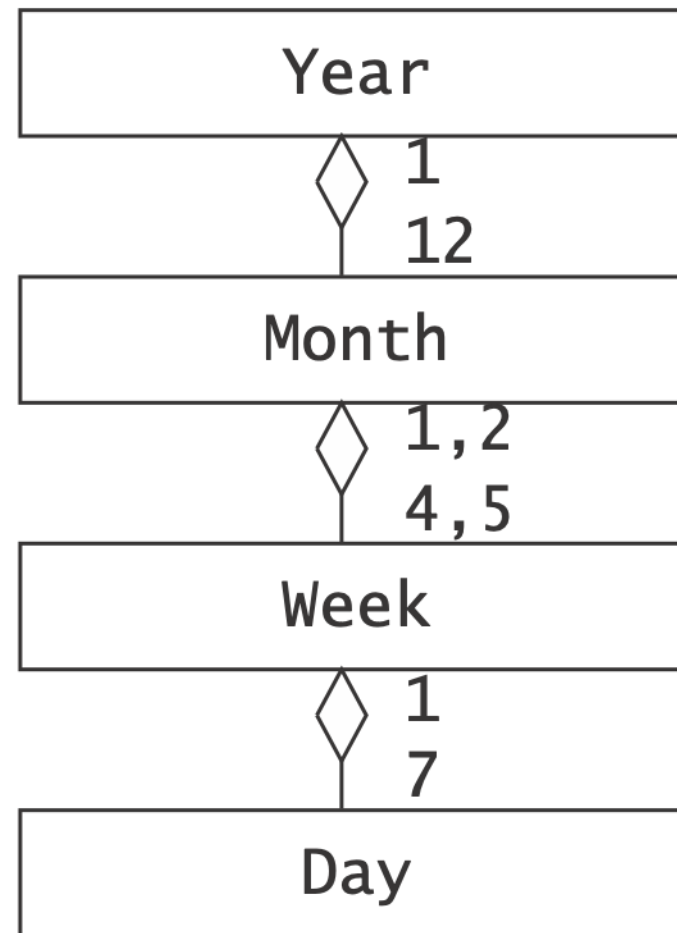




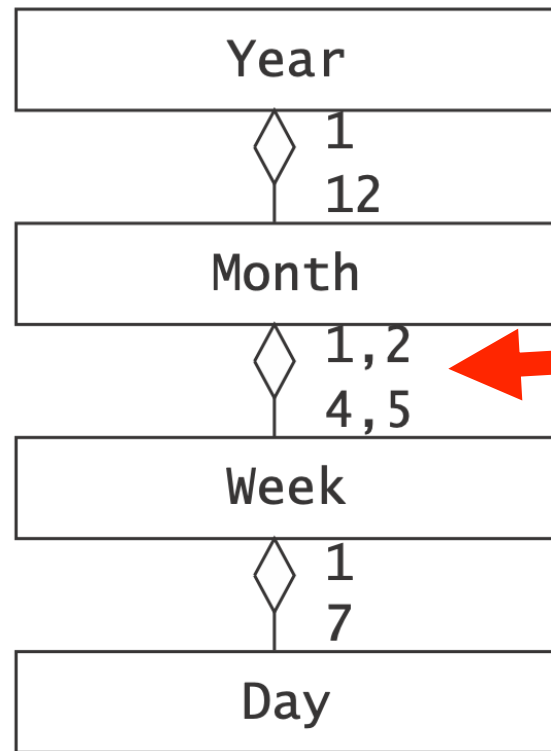
These three should
go to a **new**
abstract class!

5–6 Consider the object model in Figure 5-32 in the book (adapted from [Jackson, 1995]):
Given your knowledge of the Gregorian calendar, list all the problems with this model. Modify it to correct each of them.

- 5–6 Consider the object model in Figure 5-32 in the book (adapted from [Jackson, 1995]):
Given your knowledge of the Gregorian calendar, list all the problems with this model. Modify it to correct each of them.

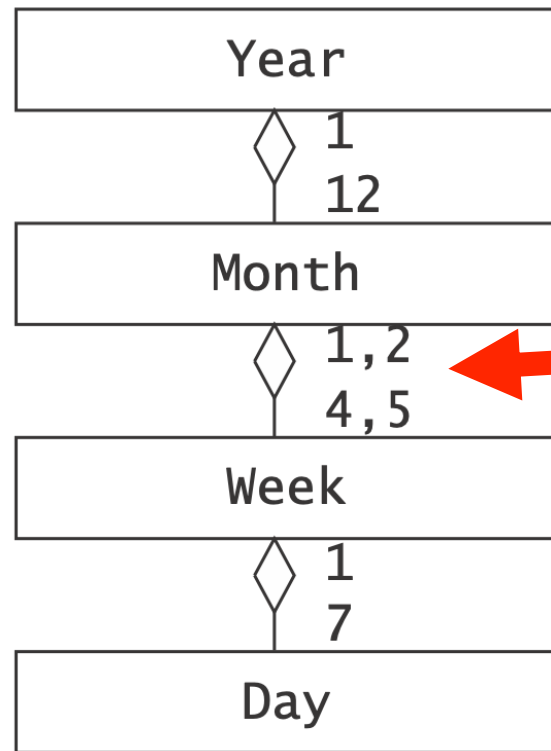


5–6 Consider the object model in Figure 5-32 in the book (adapted from [Jackson, 1995]):
Given your knowledge of the Gregorian calendar, list all the problems with this model. Modify it to correct each of them.



Say in words what
these numbers mean

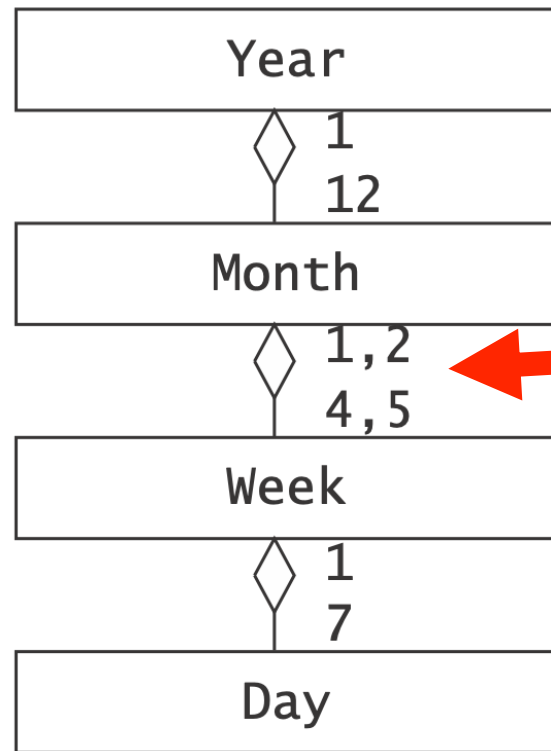
5–6 Consider the object model in Figure 5-32 in the book (adapted from [Jackson, 1995]):
Given your knowledge of the Gregorian calendar, list all the problems with this model. Modify it to correct each of them.



Say in words what
these numbers mean

— a month consists of 4 or of 5 weeks

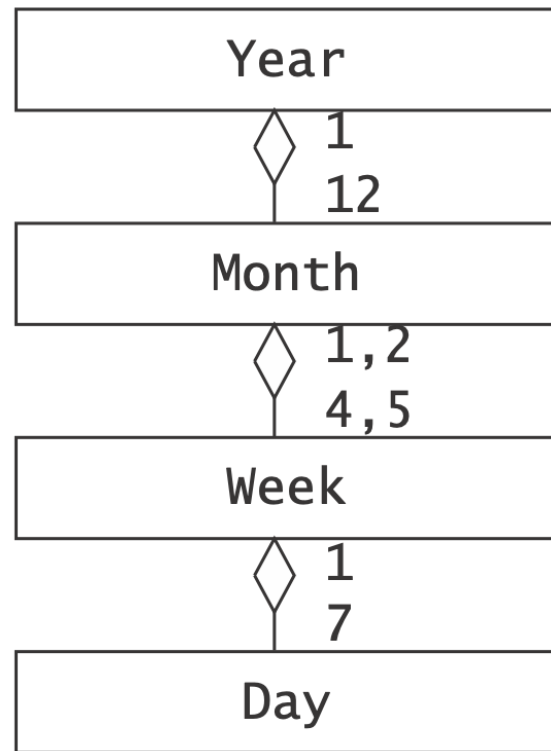
5–6 Consider the object model in Figure 5-32 in the book (adapted from [Jackson, 1995]):
Given your knowledge of the Gregorian calendar, list all the problems with this model. Modify it to correct each of them.



Say in words what these numbers mean

- a month consists of 4 or of 5 weeks
- one week belongs to 1 or to 2 months

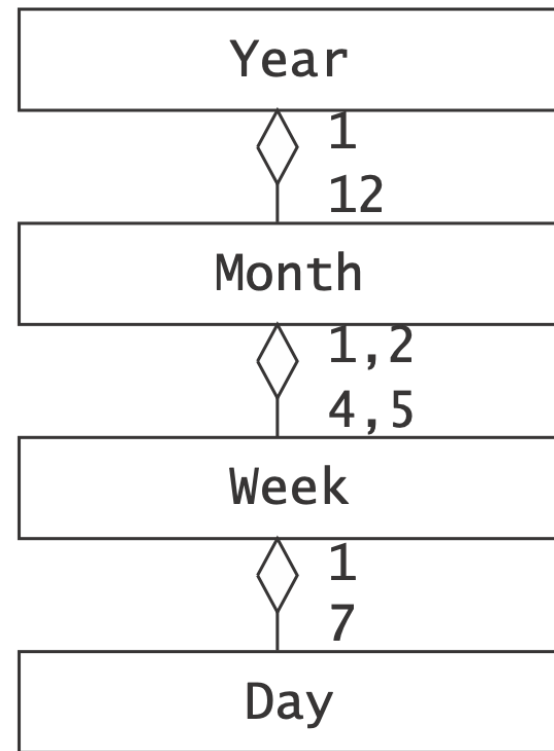
5–6 Consider the object model in Figure 5-32 in the book (adapted from [Jackson, 1995]):
Given your knowledge of the Gregorian calendar, list all the problems with this model. Modify it to correct each of them.



Problems

- fact that some months have 31 days not modeled
- fact that some months have 30 days not modeled

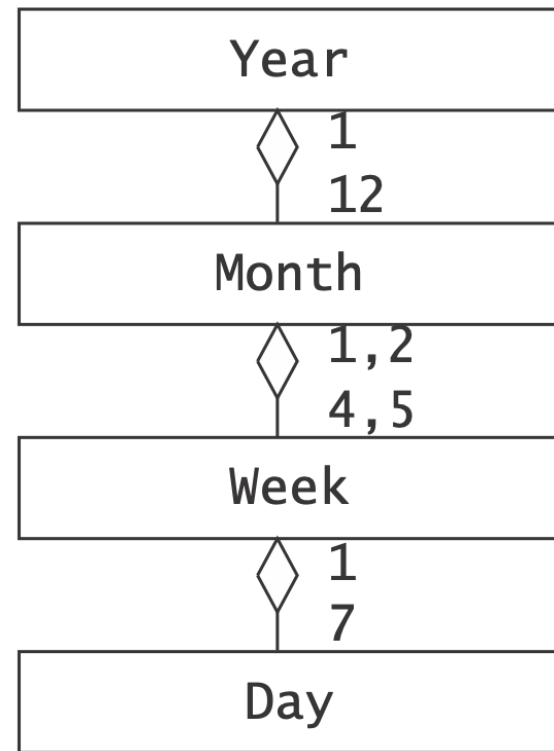
5–6 Consider the object model in Figure 5-32 in the book (adapted from [Jackson, 1995]):
Given your knowledge of the Gregorian calendar, list all the problems with this model. Modify it to correct each of them.



Problems

- fact that some months have 31 days not modeled
- fact that some months have 30 days not modeled
- fact that February has 28 (or 29) days not modeled

5–6 Consider the object model in Figure 5-32 in the book (adapted from [Jackson, 1995]):
Given your knowledge of the Gregorian calendar, list all the problems with this model. Modify it to correct each of them.

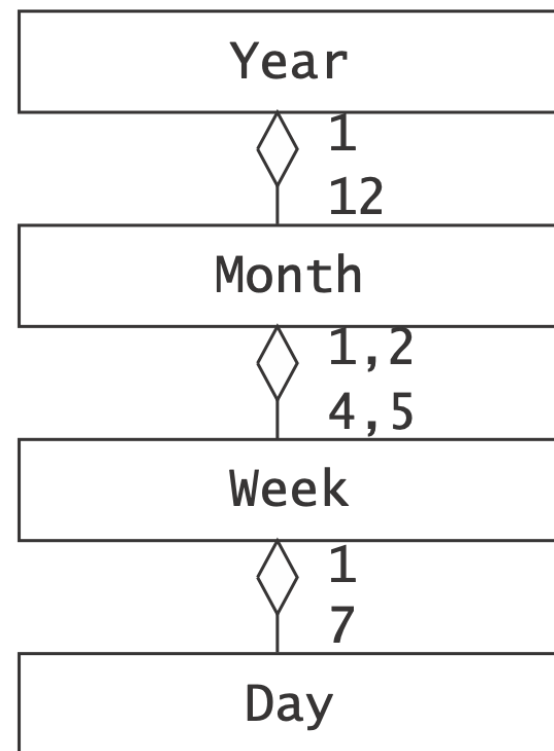


Problems

- fact that some months have 31 days not modeled
- fact that some months have 30 days not modeled
- fact that February has 28 (or 29) days not modeled

Let's model each of these three aspects!

5–6 Consider the object model in Figure 5-32 in the book (adapted from [Jackson, 1995]):
Given your knowledge of the Gregorian calendar, list all the problems with this model. Modify it to correct each of them.



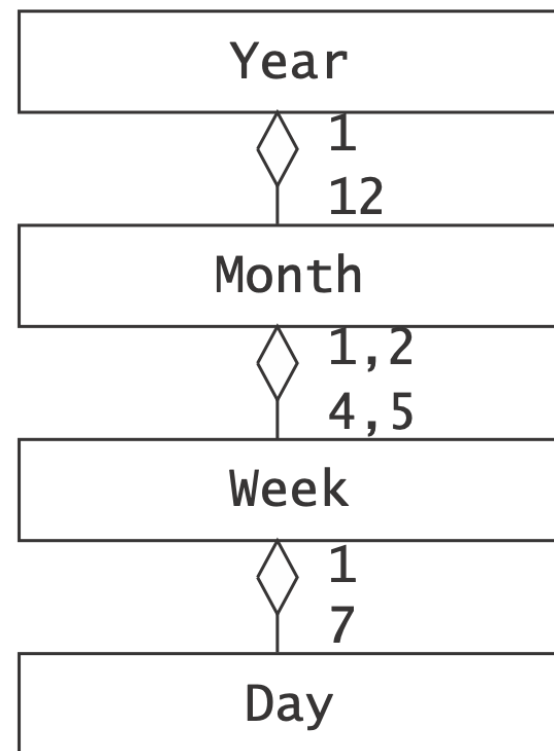
Problems

- fact that some months have **31** days not modeled
- fact that some months have **30** days not modeled
- fact that February has **28** (or **29**) days not modeled

Let's model each of these three aspects!

Four new subclasses of the abstract Class "Month"!

5–6 Consider the object model in Figure 5-32 in the book (adapted from [Jackson, 1995]):
Given your knowledge of the Gregorian calendar, list all the problems with this model. Modify it to correct each of them.



Problems

- fact that some months have **31** days not modeled
- fact that some months have **30** days not modeled
- fact that February has **28** (or **29**) days not modeled

Let's model each of these three aspects!

Four new subclasses of the abstract Class "Month"!

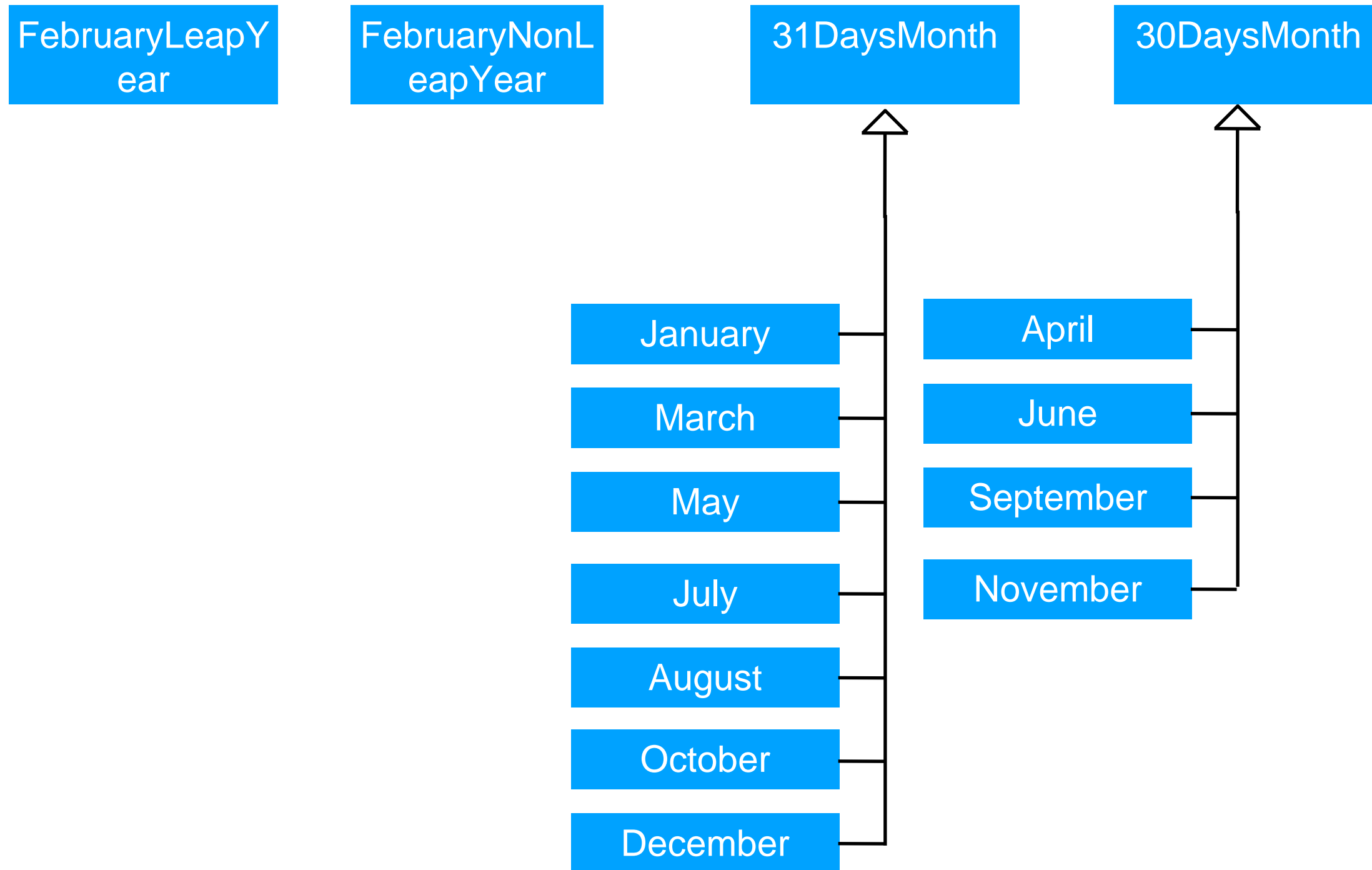
FebruaryNonLeap

FebruaryLeap

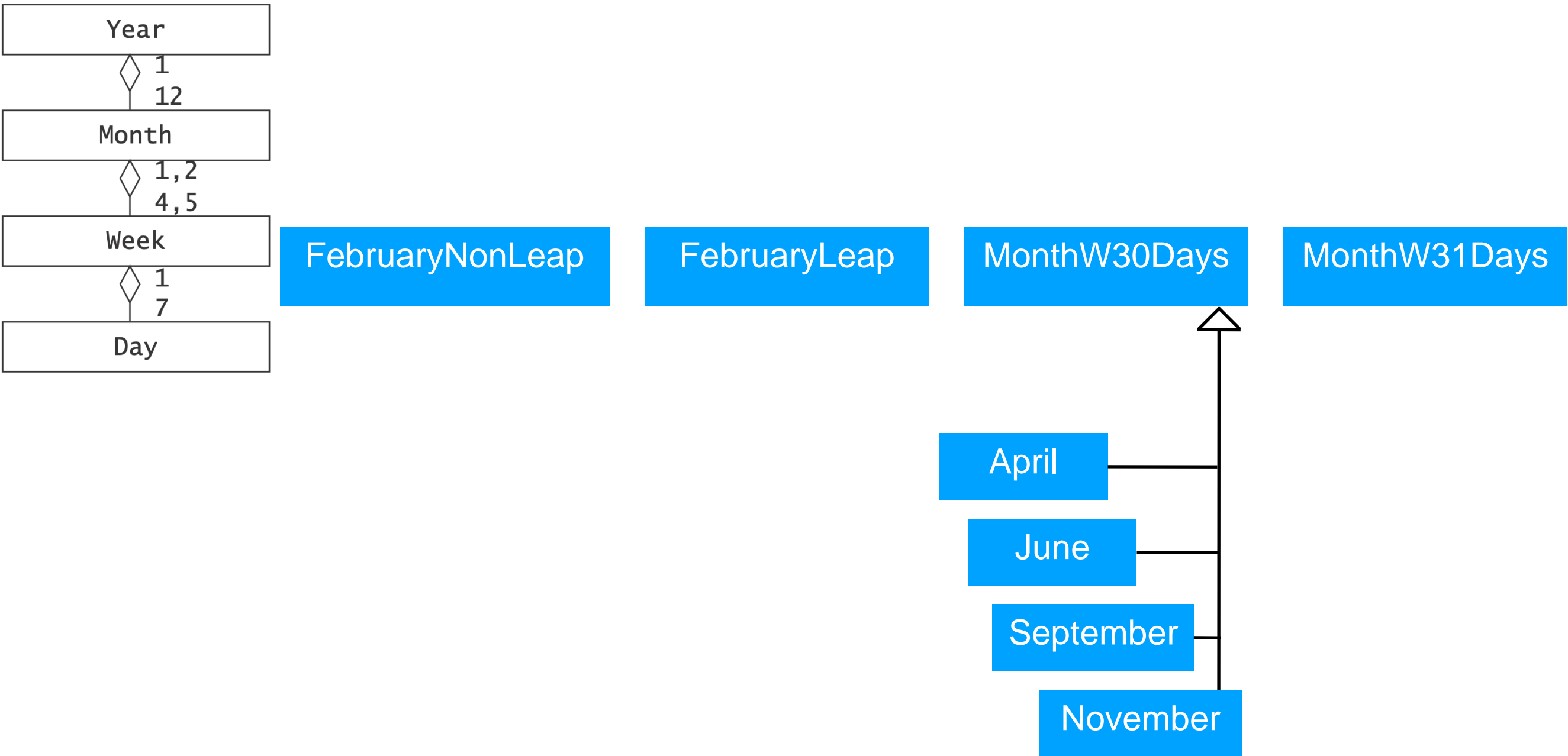
MonthW30Days

MonthW31Days

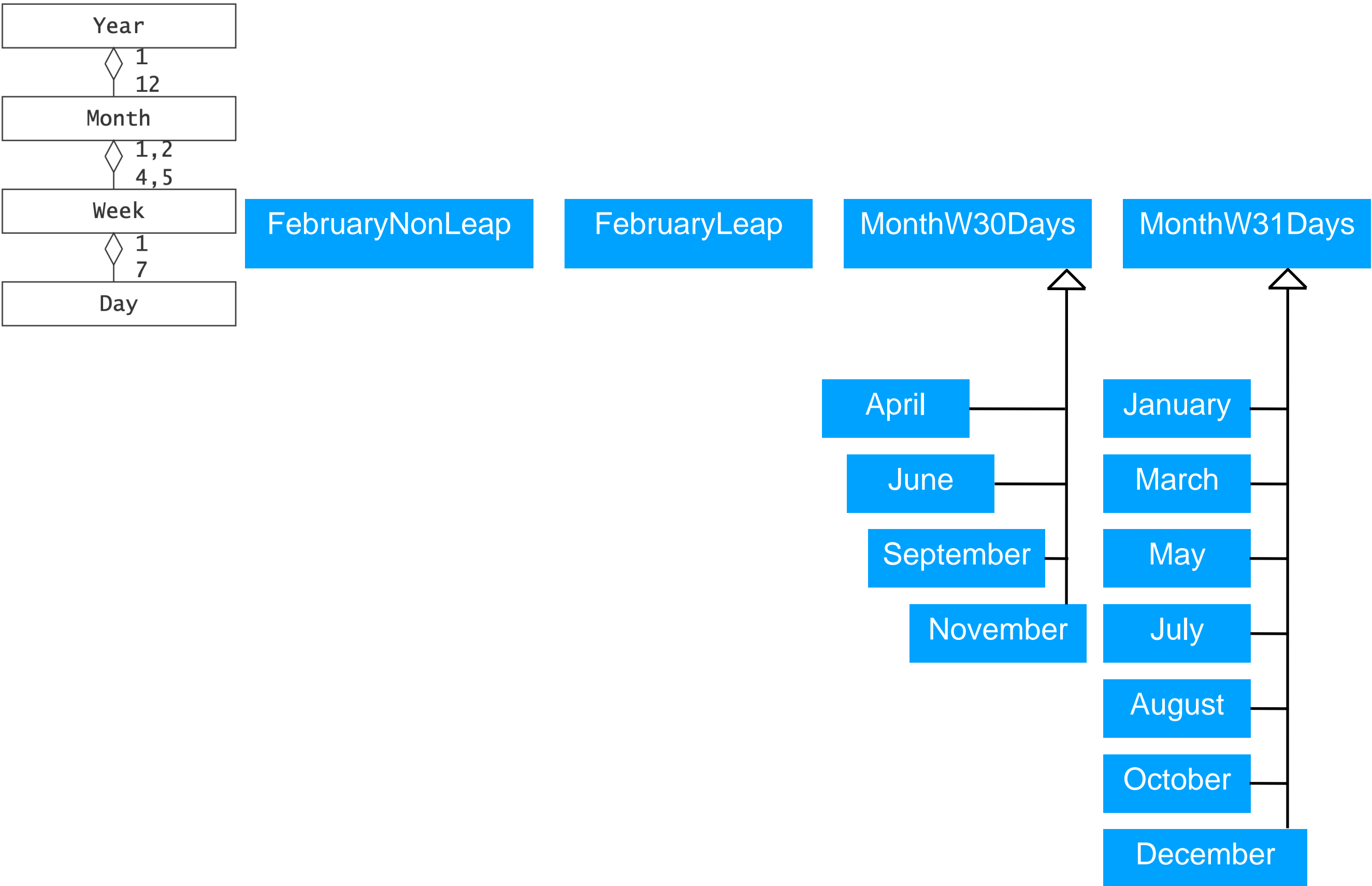
5–6 Consider the object model in Figure 5-32 in the book (adapted from [Jackson, 1995]):
Given your knowledge of the Gregorian calendar, list all the problems with this model. Modify it to correct each of them.



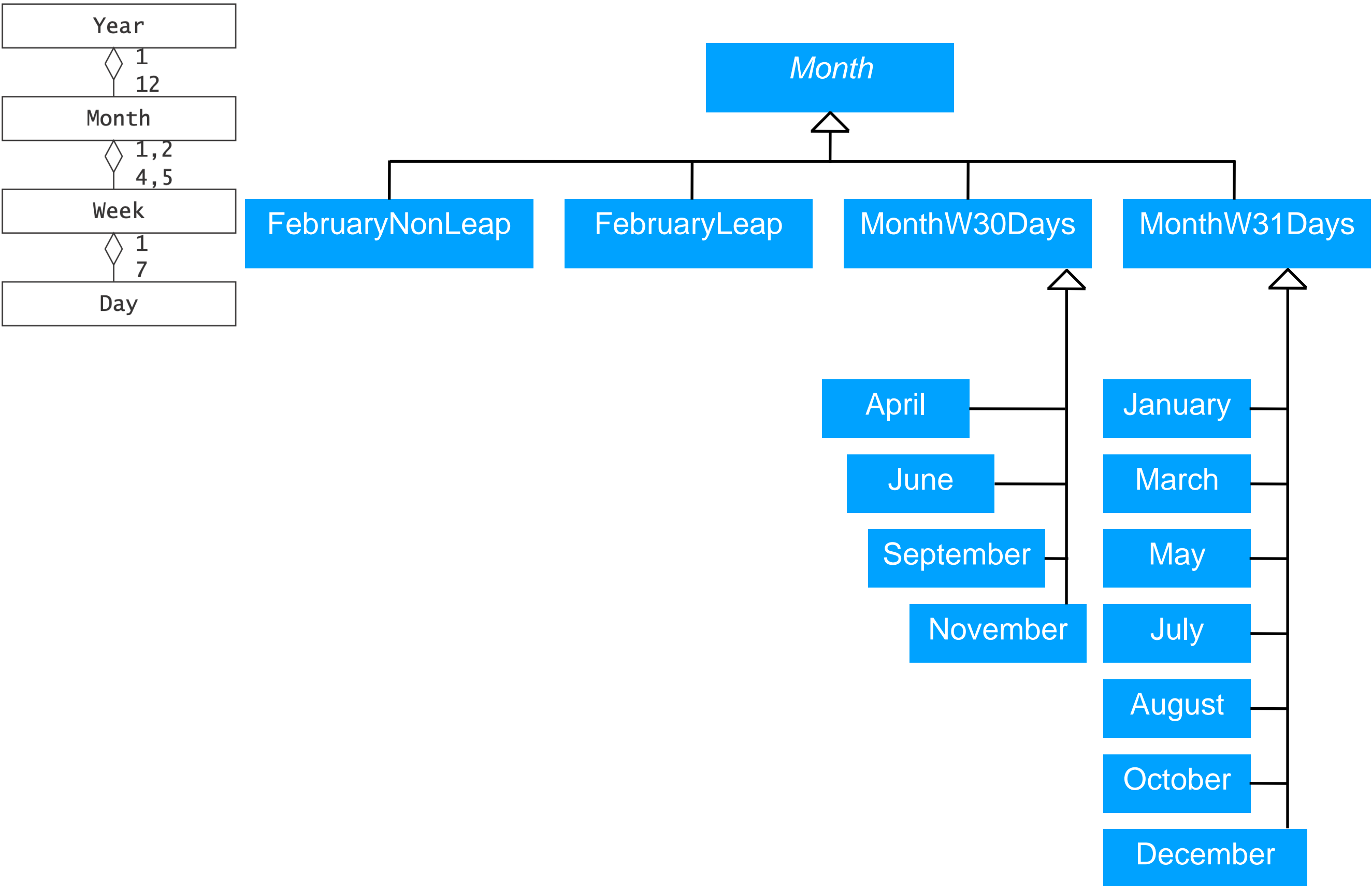
5–6 Consider the object model in Figure 5-32 in the book (adapted from [Jackson, 1995]):
Given your knowledge of the Gregorian calendar, list all the problems with this model. Modify it to correct each of them.

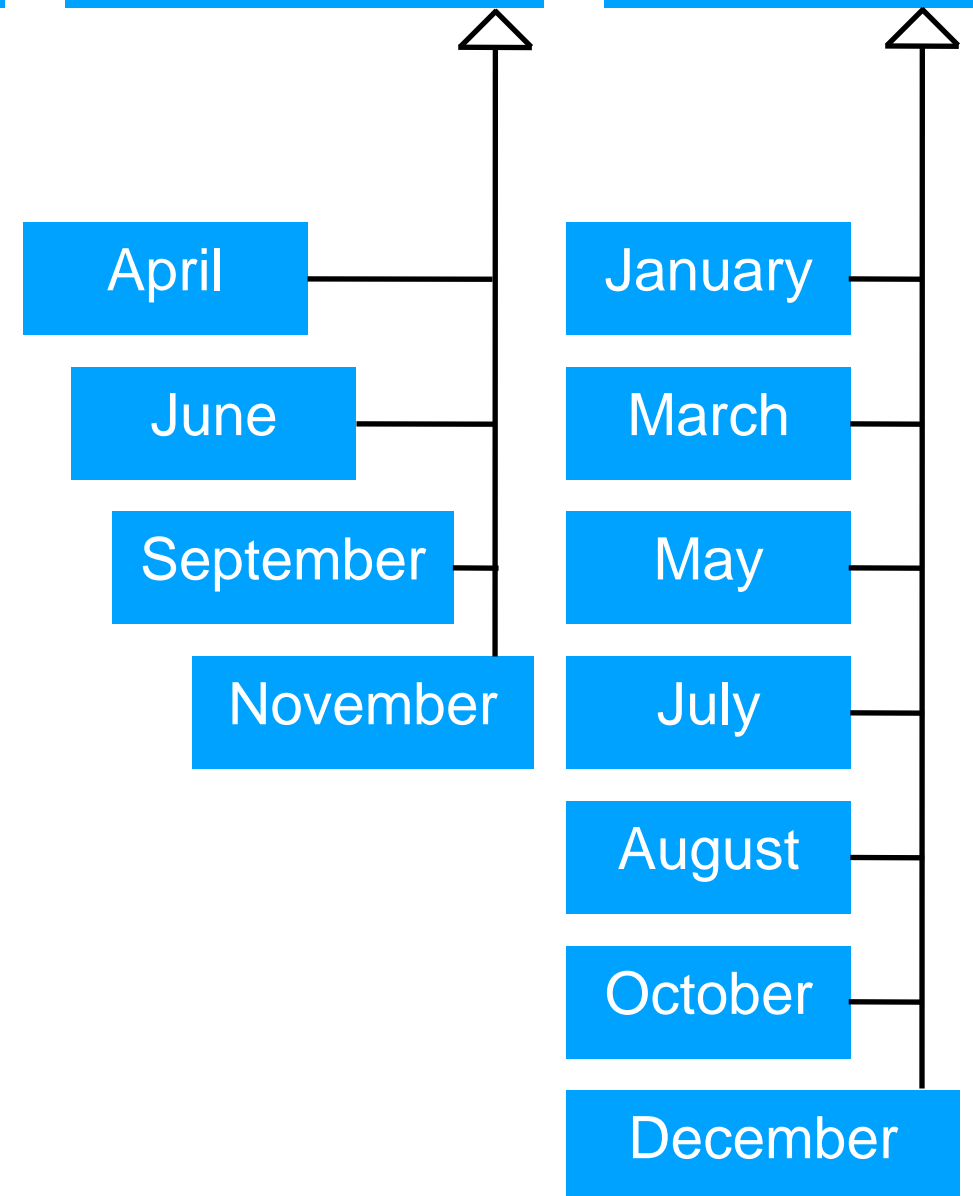
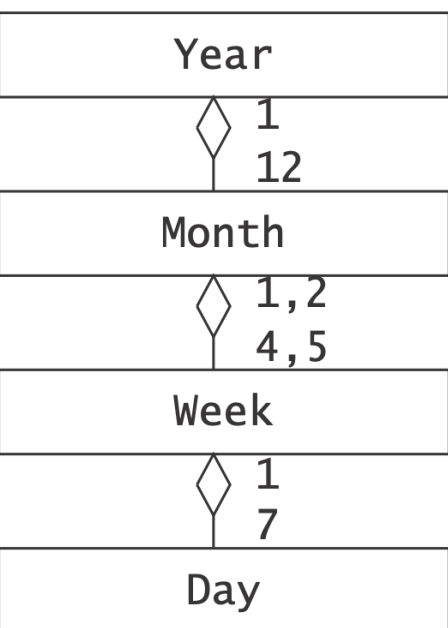
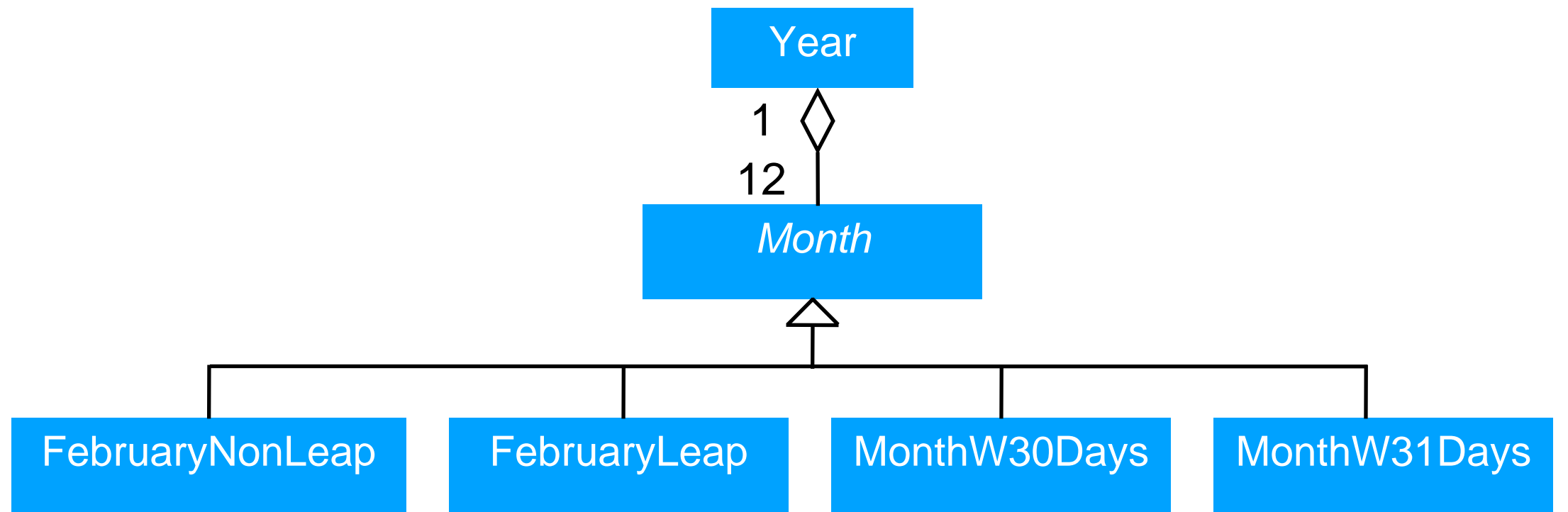


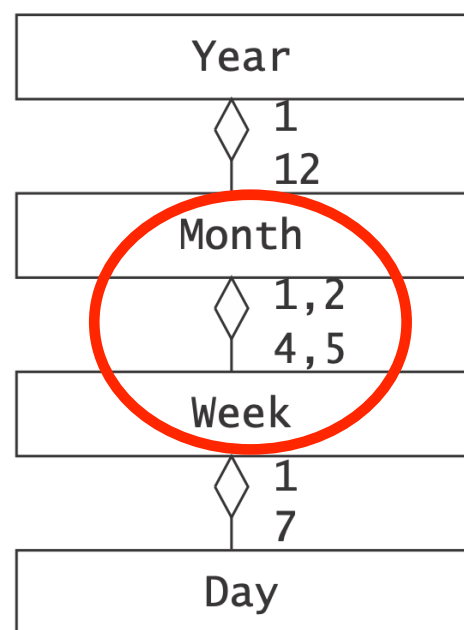
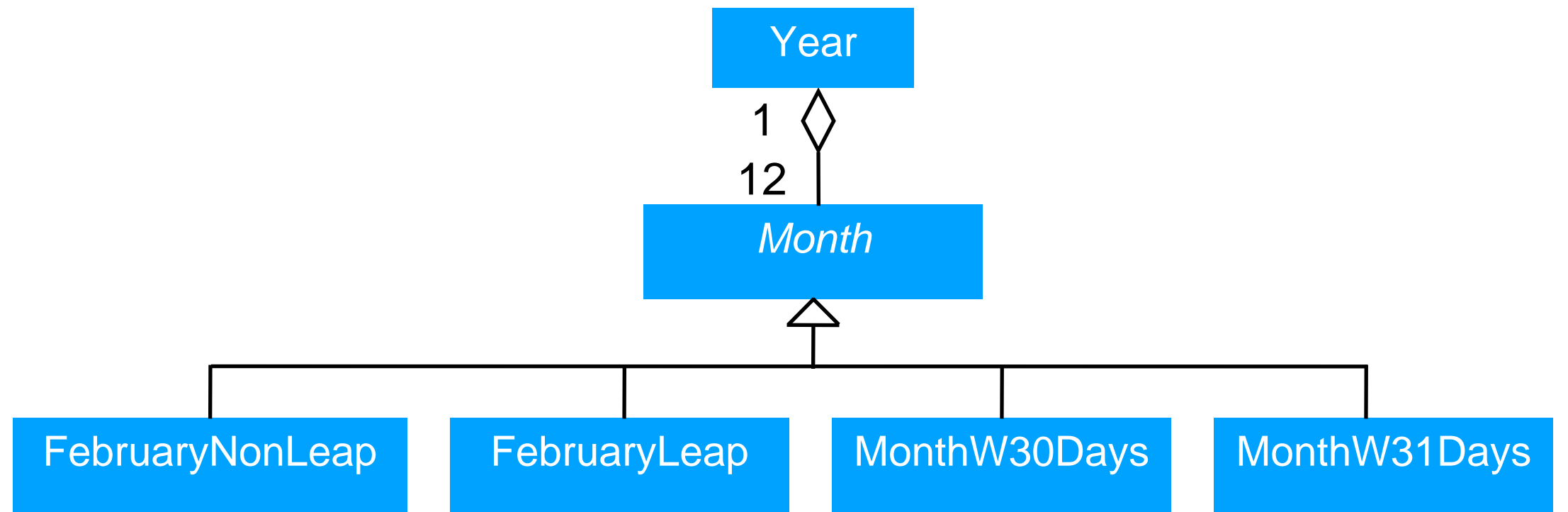
5–6 Consider the object model in Figure 5-32 in the book (adapted from [Jackson, 1995]):
Given your knowledge of the Gregorian calendar, list all the problems with this model. Modify it to correct each of them.



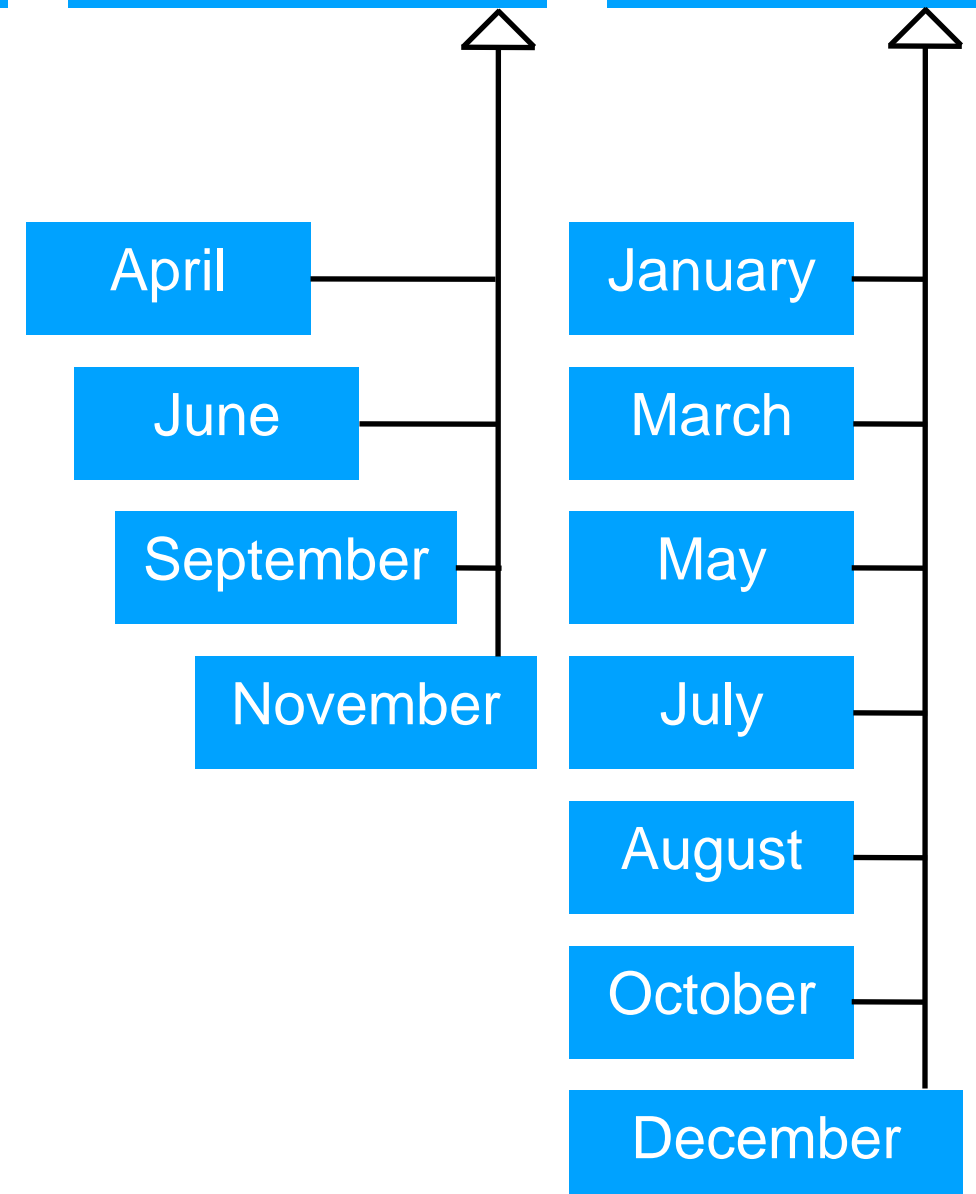
5–6 Consider the object model in Figure 5-32 in the book (adapted from [Jackson, 1995]):
Given your knowledge of the Gregorian calendar, list all the problems with this model. Modify it to correct each of them.



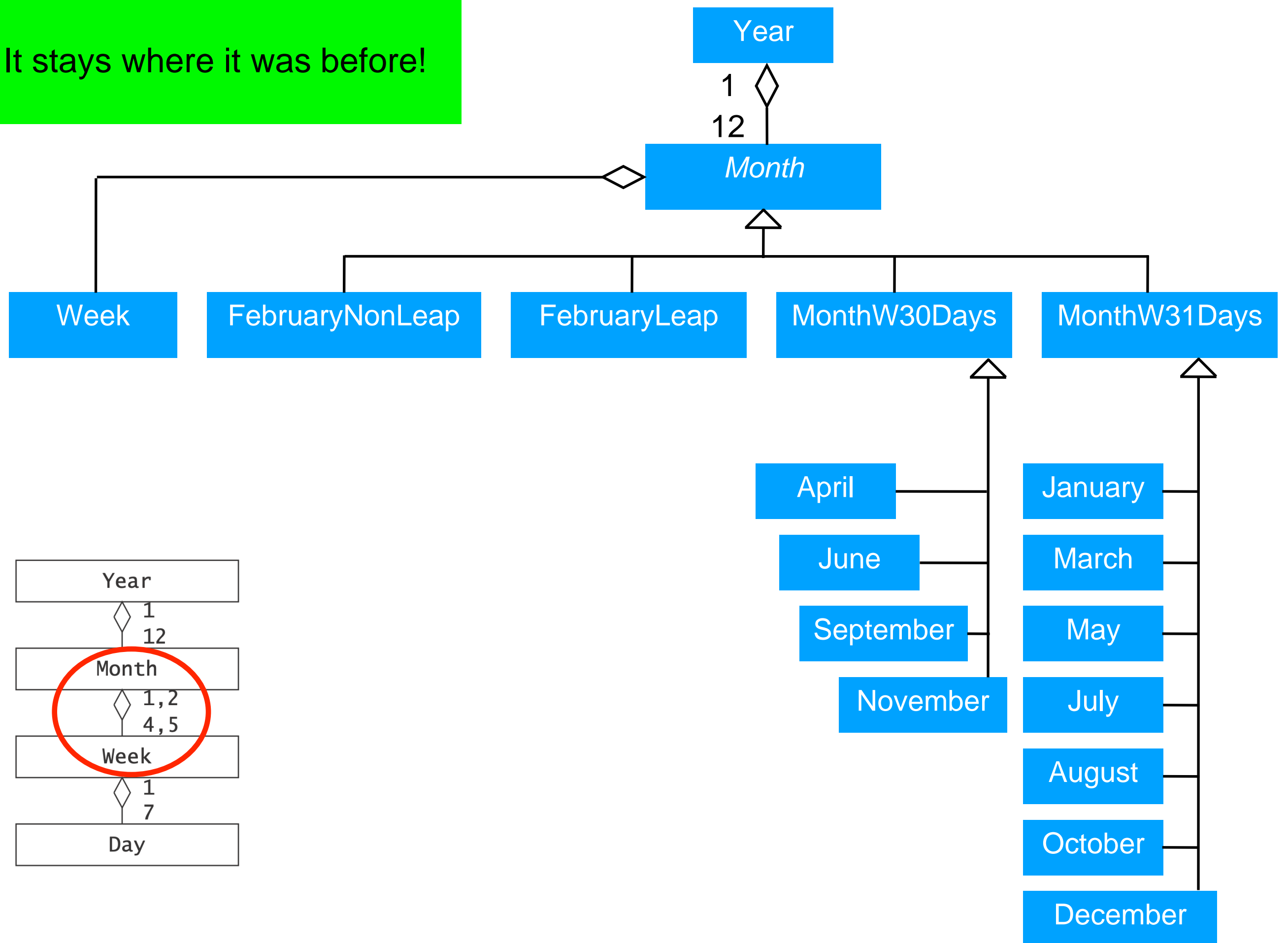




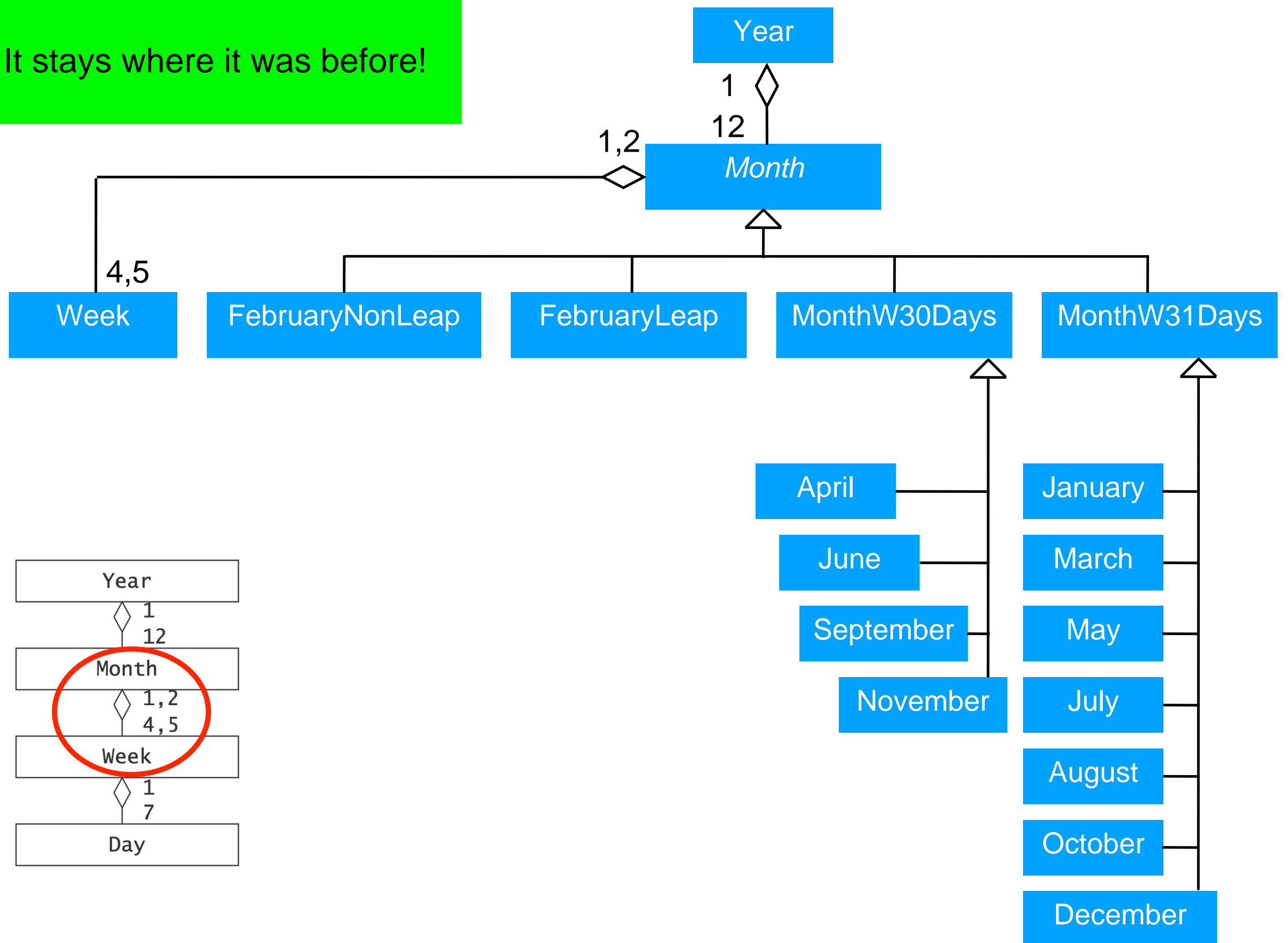
Where do we put the
“**consists-of-Weeks**”
relationship (arrow)??



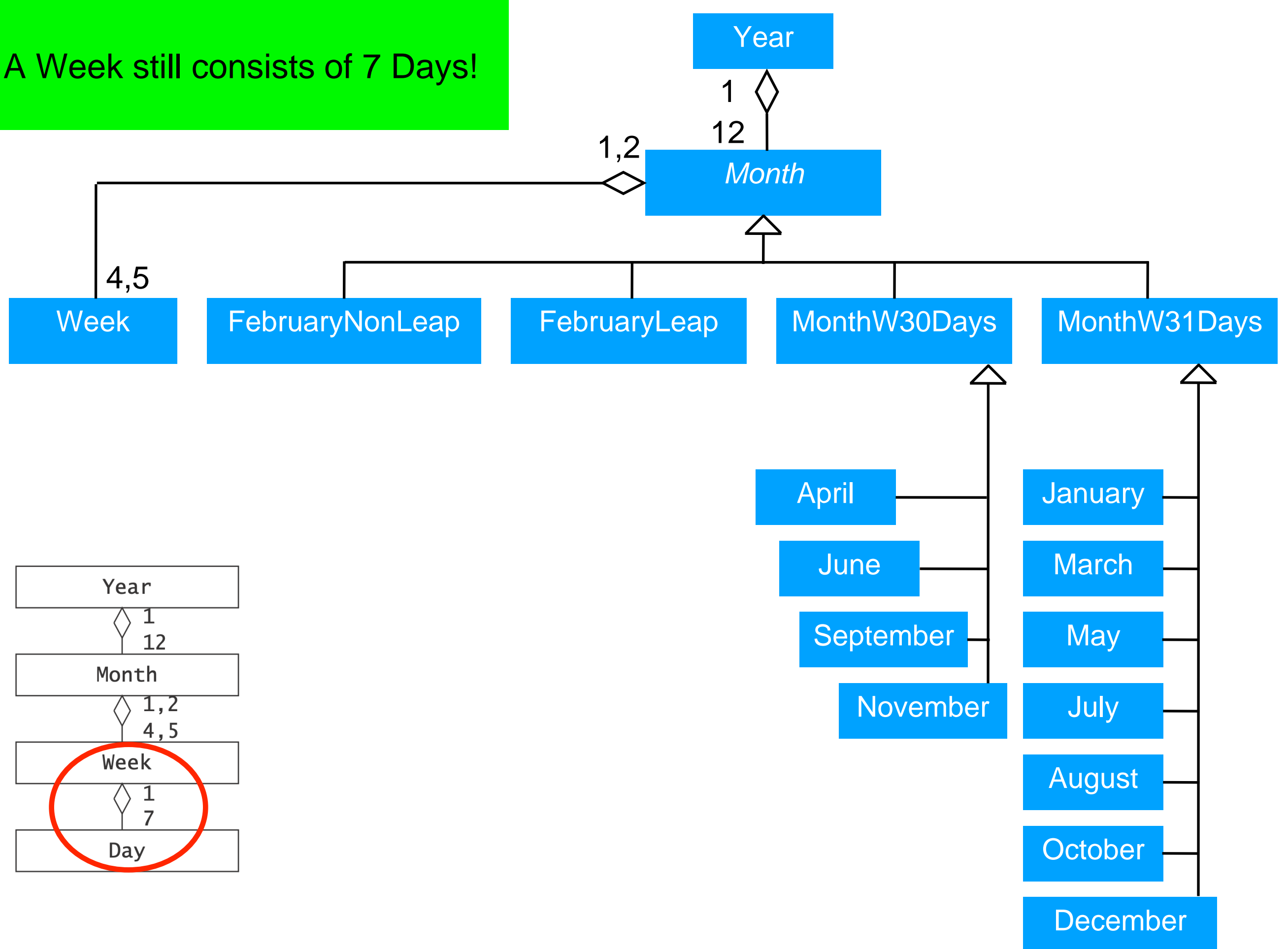
It stays where it was before!

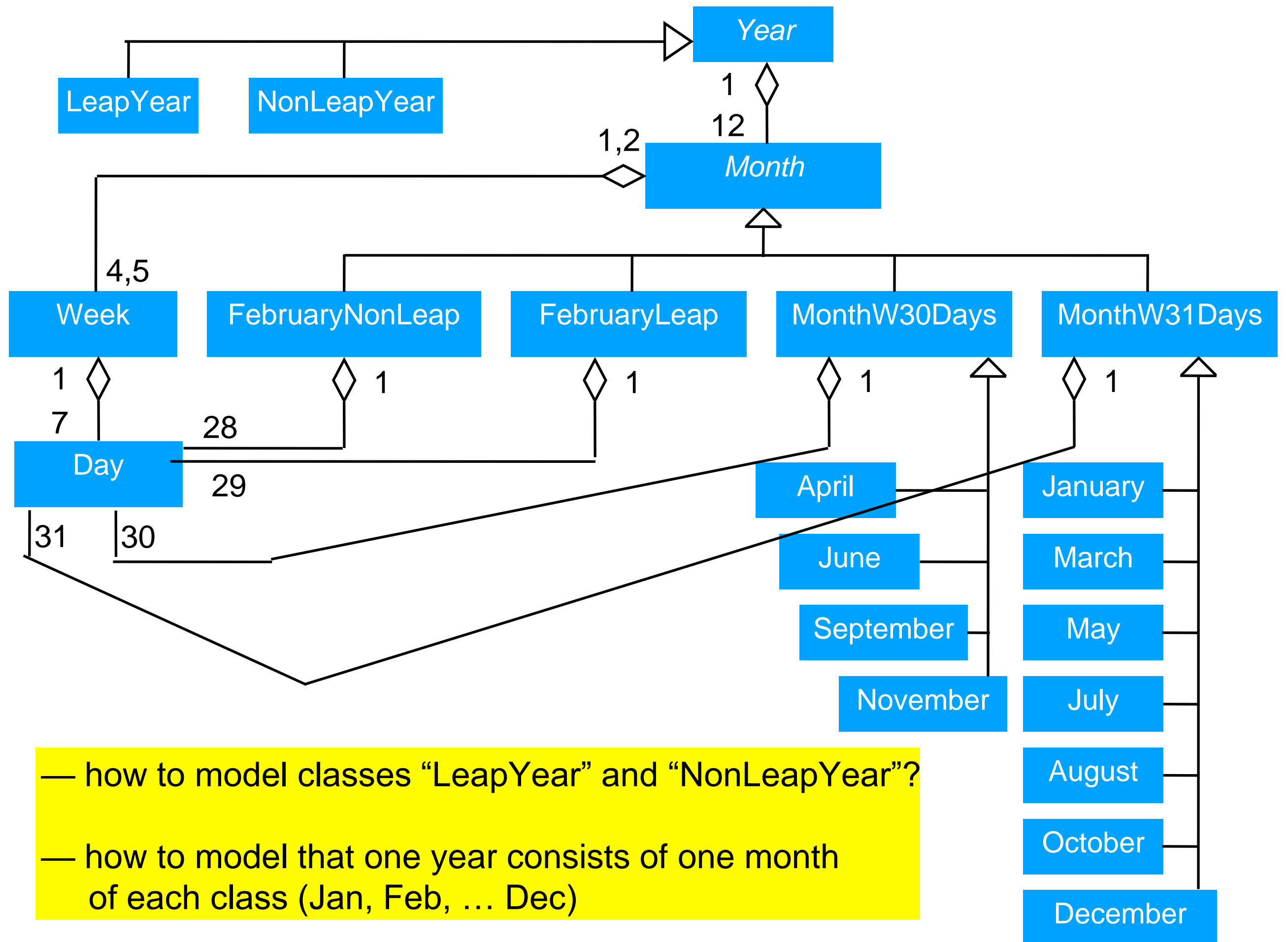


It stays where it was before!

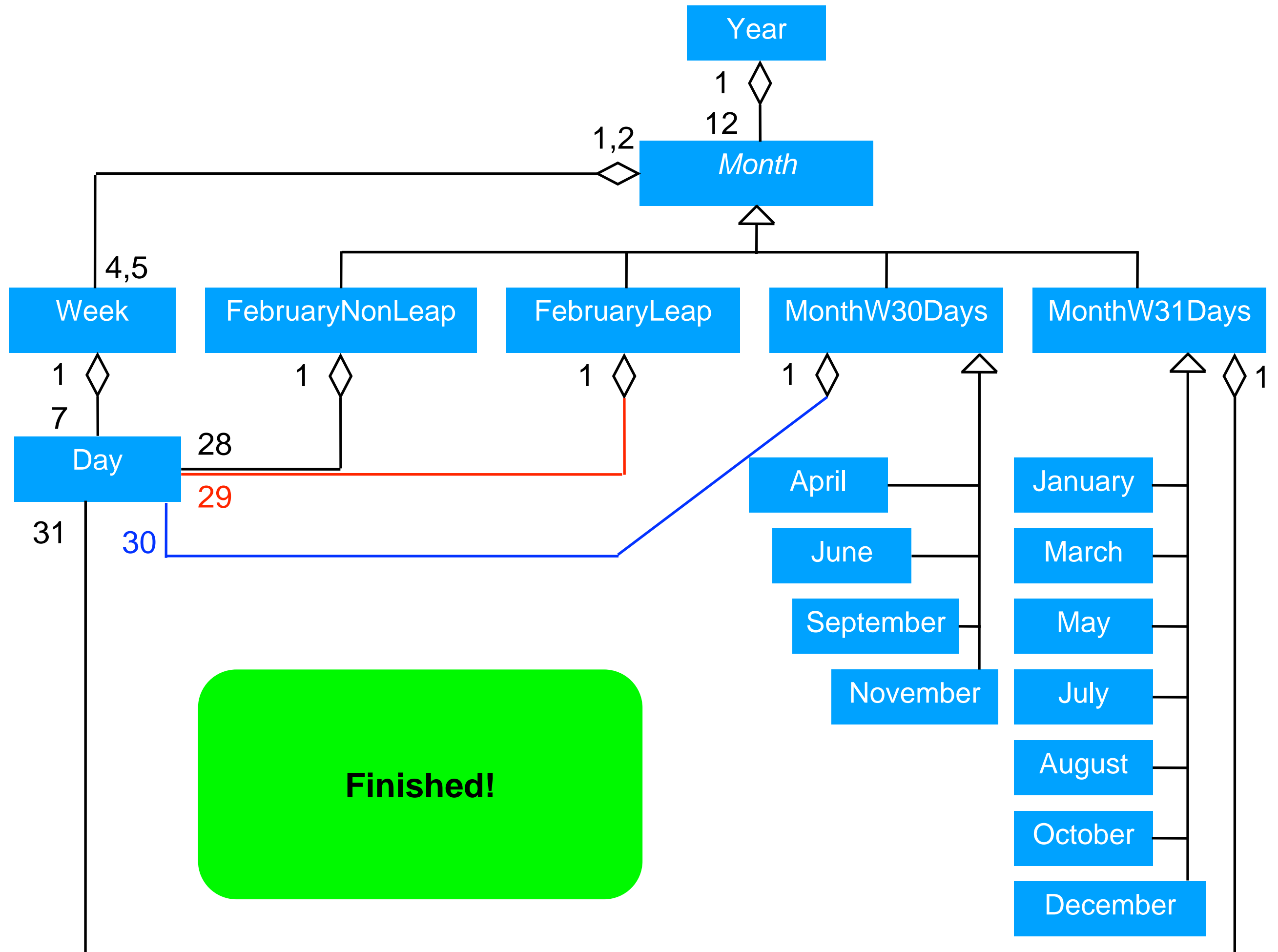


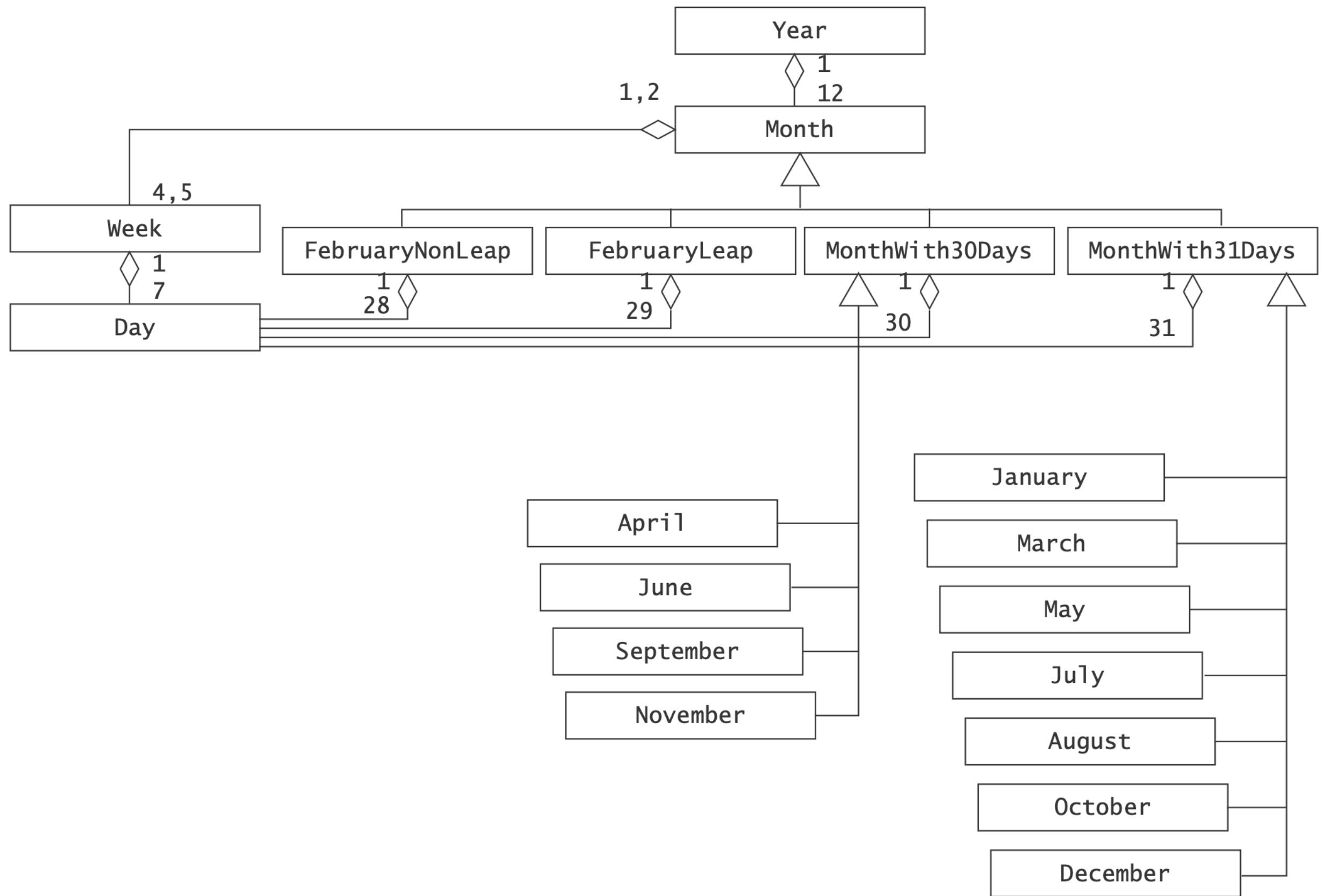
A Week still consists of 7 Days!

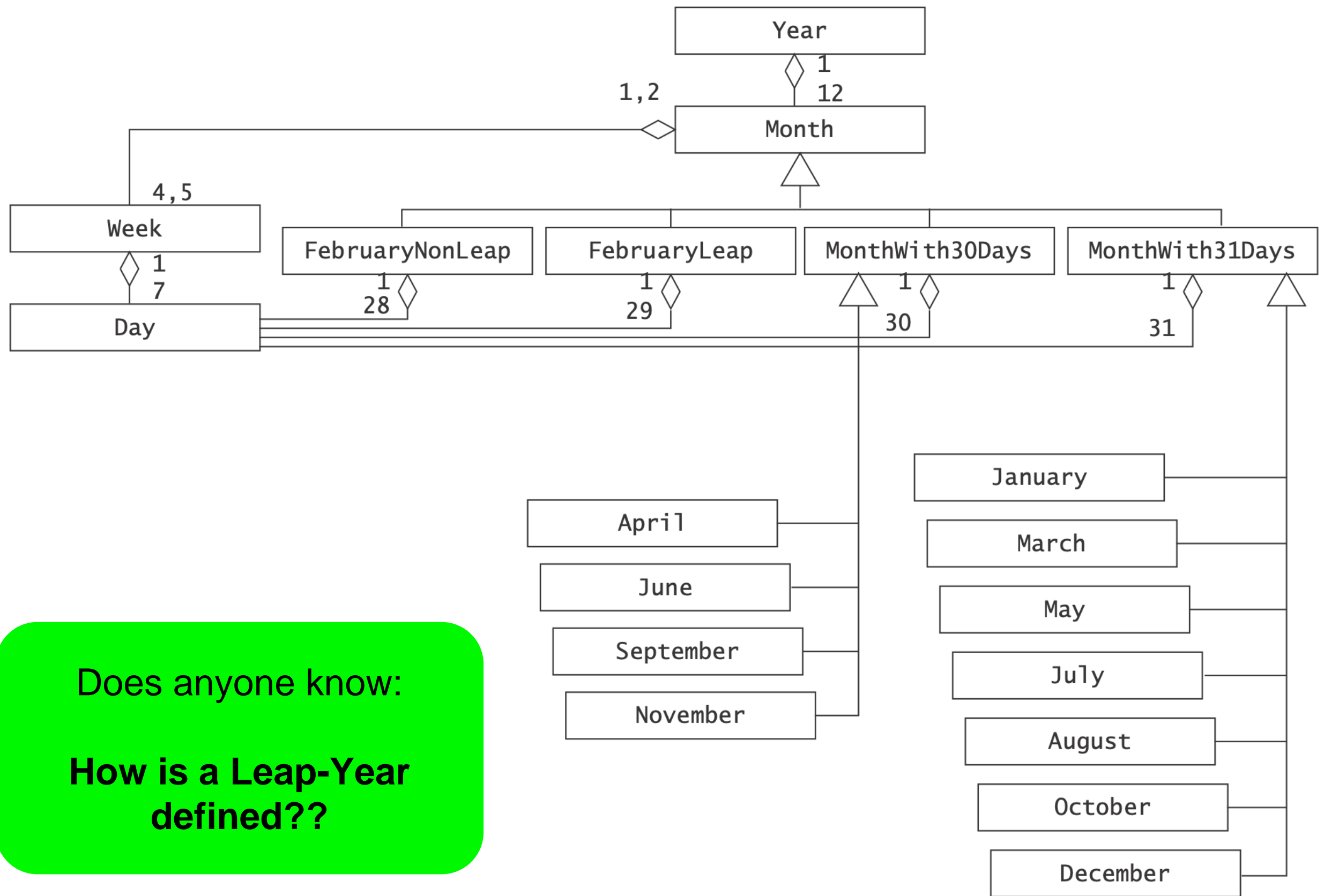




- how to model classes “LeapYear” and “NonLeapYear”?
- how to model that one year consists of one month of each class (Jan, Feb, ... Dec)







Does anyone know:


How is a Leap-Year defined??



- [Contribute](#)
- [Help](#)
- [Learn to edit](#)
- [Community portal](#)
- [Recent changes](#)
- [Upload file](#)

Print/export

Download as PDF
Printable version

Languages 

Boarisch

Deutsch

Español




Français

Italiano

Nederlands

Plattdüütsch

 https://en.wikipedia.org/wiki/Leap_year


133%




 Not logged in [Talk](#) [Contributions](#) [Create account](#) [Log in](#)

Read Edit View history

Search Wikipedia 

Leap year

From Wikipedia, the free encyclopedia

This article is about the 366-day year. For the 365-day year, see [Common year](#). For other uses, see [Leap year \(disambiguation\)](#).



This article **needs additional citations for verification**. Please help [improve this article](#) by [adding citations to reliable sources](#). Unsourced material may be challenged and removed.

Find sources: "Leap year" – news · newspapers · books · scholar · JSTOR (February 2020) (Learn how and when to remove this template message)

A **leap year** (also known as an **intercalary year** or **bissextile year**) is a **calendar year** that contains an additional day (or, in the case of a **lunisolar calendar**, a month) added to keep the calendar year synchronized with the **astronomical year** or **seasonal year**.^[1] Because astronomical events and seasons do not repeat in a **whole number** of days, calendars that have a constant number of days in each year will unavoidably drift over time with respect to the event that the year is supposed to track, such as seasons. By inserting (called *intercalating* in technical terminology) an additional day or month into some years, the drift between a civilization's dating system and the physical properties of the **solar system** can be corrected. A year that is not a leap year is a **common year**.

For example, in the [Gregorian calendar](#), each leap year has 366 days instead of 365, by [extending February to 29 days](#) rather than the common 28. These extra days occur in each year which is an [integer](#) multiple of 4 (except for years evenly divisible by 100, which are not leap years unless evenly divisible by 400). The leap year of 366 days has 52 weeks and two days, hence the year following a leap year will start later by two days of the week.

In the lunisolar [Hebrew calendar](#), [Adar Aleph](#), a 13th lunar month, is added seven times every 19 years to the twelve lunar months in its common years to keep its calendar year from drifting through the seasons. In the [Bahá'í Calendar](#), a leap day is added when needed to ensure that the following year begins on the [March equinox](#).

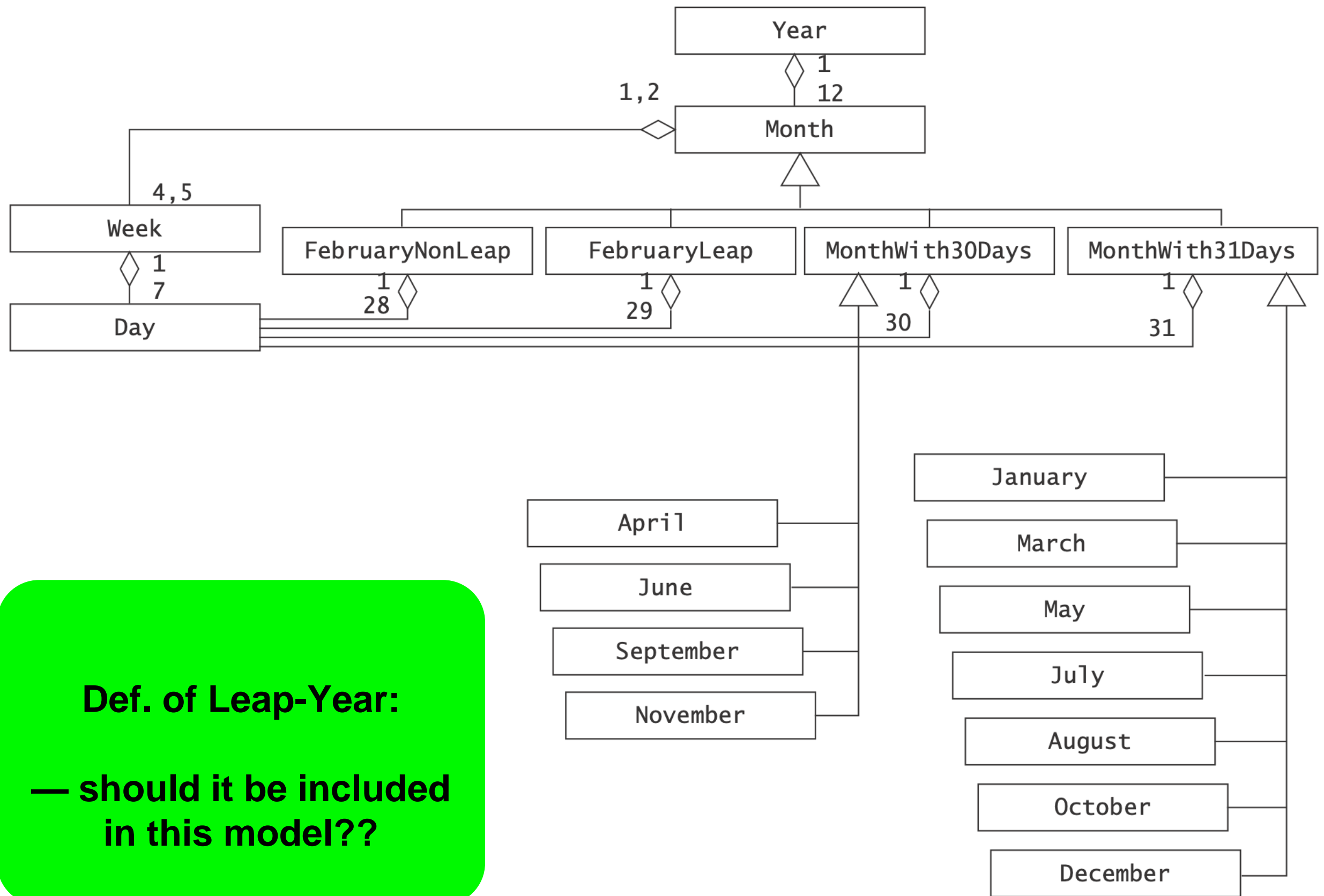
The term *leap year* probably comes from the fact that a fixed date in the Gregorian calendar normally advances one day of the week from one year to the next, but the day of the week in the 12 months following the leap day (from March 1 through February 28 of the following year) will advance two days due to the extra day, thus leaping over one day in the week.^{[2][3]} For example, **Christmas Day** (December 25) falls on a Friday in 2020, Saturday in 2021, Sunday in 2022 and Monday in 2023, but then will leap over Tuesday to fall on a Wednesday in 2024.

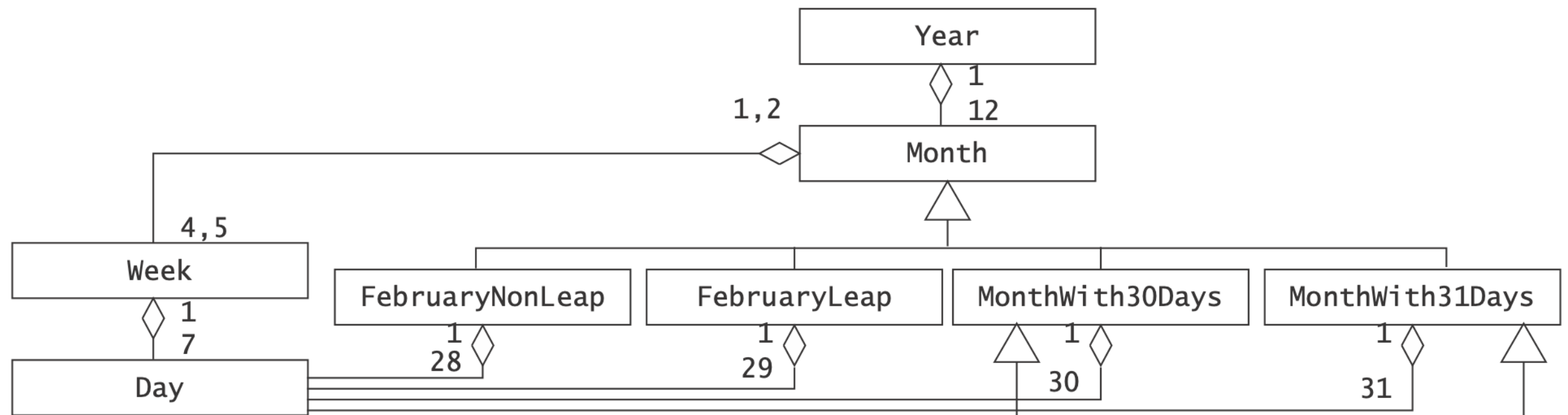
The length of a day is also occasionally corrected by inserting a [leap second](#) into [Coordinated Universal Time](#) (UTC) because of variations in Earth's [rotation period](#). Unlike leap days, leap seconds are not introduced on a regular schedule because variations in the length of the day are not entirely predictable.

Leap years can present a problem in computing, known as the [leap year bug](#), when a year is not correctly identified as a leap year or when February 29 is not handled correctly in logic that accepts or manipulates dates.

Contents [\[hide\]](#)

- 1 Julian calendar
- 2 Gregorian calendar
 - 2.1 Algorithm
 - 2.2 Leap day
 - 2.3 Synchronized calendars (Bengali, Indian and Thai)





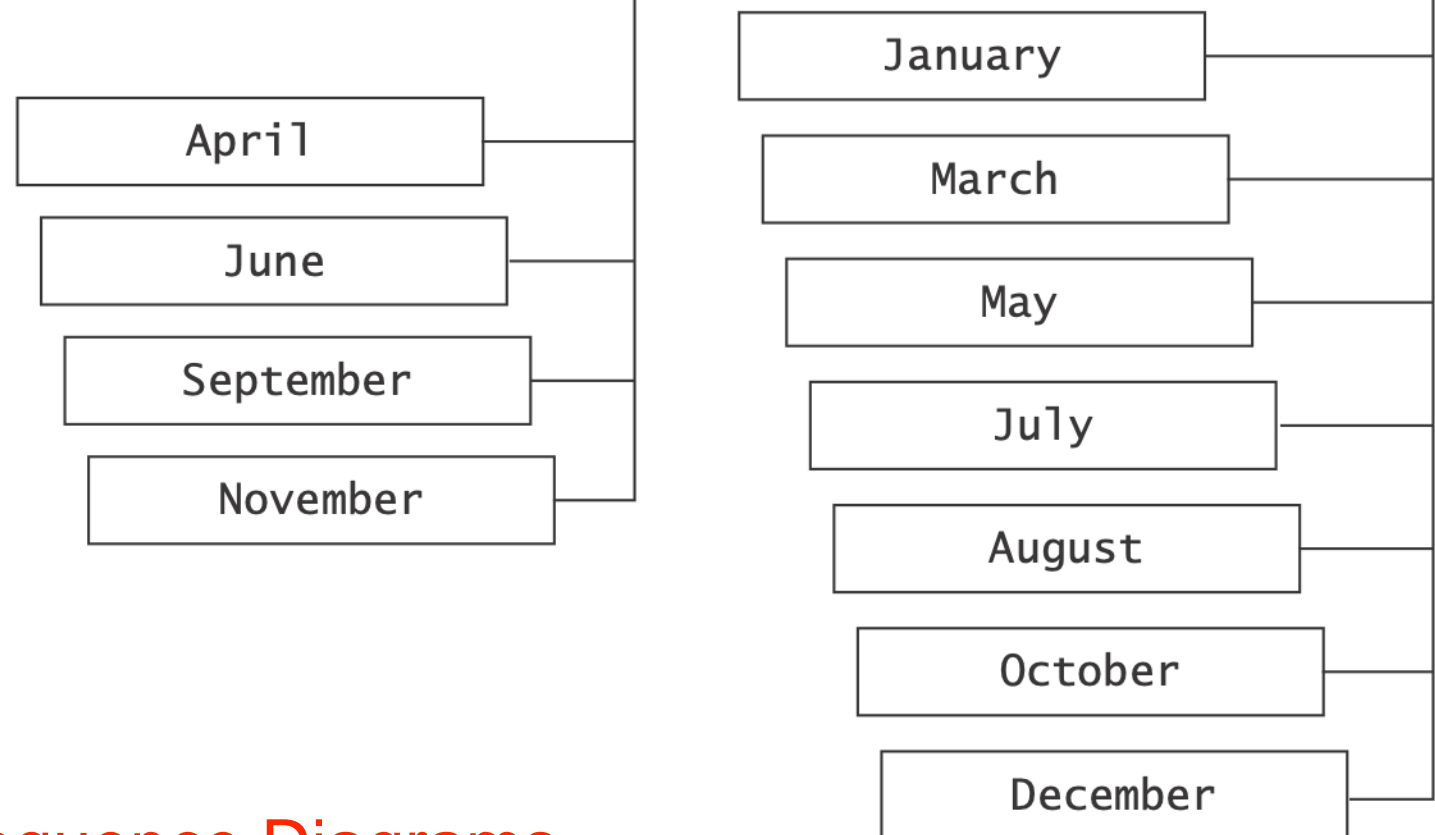
Def. of Leap-Year:

— should it be included in this model??

I would say **no**.

Modeling of Years is dynamic!

— use, e.g., **StateCharts** or **Sequence Diagrams**.



10–7 Draw a class diagram representing the application domain facts below,

- A project involves a number of participants.*
- Participants can take part in a project either as project manager, team leader, or developer.*
- Within a project, each developer and team leader is part of at least one team.*
- A participant can take part in many projects, possibly in different roles. For example, a participant can be a developer in project A, a team leader in project B, and a project manager in project C. However, the role of a participant within a project does not change.*

10–7 Draw a class diagram representing the application domain facts below, and map it to a relational schema.

- *A project involves a number of participants.*
- *Participants can take part in a project either as project manager, team leader, or developer.*
- *Within a project, each developer and team leader is part of at least one team.*
- *A participant can take part in many projects, possibly in different roles. For example, a participant can be a developer in project A, a team leader in project B, and a project manager in project C. However, the role of a participant within a project does not change.*

10–7 Draw a class diagram representing the application domain facts below, and map it to a relational schema.

- A project involves a number of participants.
- Participants can take part in a project either as project manager, team leader, or developer.
- Within a project, each developer and team leader is part of at least one team.
- A participant can take part in many projects, possibly in different roles. For example, a participant can be a developer in project A, a team leader in project B, and a project manager in project C. However, the role of a participant within a project does not change.

