Design Problem – Book Tracker

To be successful at solving these types of problems you must first read the story problem and do the following "To Do"s

To Do #1 Identify all the *primary nouns*

- A *primary noun* is a noun in the problem that has *attributes* (*other nouns*)
- Nouns that designate *actors* of the system (i.e. The *user* can click…) can be **excluded**

To Do #2 Write down the *attributes* (*other nouns)* associated with the *primary nouns*
To Do #3 Identify all the *verbs*
To Do #4 Identify which *primary nouns* are worked on by the *verbs*
To Do #5 Design a system using UML to handle this problem

If you do this all well, then you should hopefully be able to make classes where:

- Each *primary noun* becomes a Class
- Each *attribute* (*other noun*) becomes a field for its respective class
- Each *verb* becomes the method for the respective class

A website tracks books and the kids that read them. For each book the system stores the name and author. For each kid the system stores name and grade level. The teacher enters when a kid reads a particular book. It should be possible to print a report on a book that includes all kids who have read a particular book (with their grade level). It should be possible to print a report on a kid that includes the books (with authors) a particular kid has read.

To Do #1 Identify all the *primary nouns*

- A *primary noun* is a noun in the problem that has *attributes* (*other nouns*)
- Nouns that designate *actors* of the system (i.e. The *user* can click…) can be **excluded**

To Do #2 Write down the *attributes* (*other nouns)* associated with the *primary nouns*

To Do #3 Identify all the *verbs*

To Do #4 Identify which *primary nouns* are worked on by the *verbs*

To Do #5 Design a system using UML to handle this problem

A website tracks books and the kids that read them.

To Do #1 Identify all the *primary nouns*

- A *primary noun* is a noun in the problem that has *attributes* (*other nouns*)
- Nouns that designate *actors* of the system (i.e. The *user* can click…) can be **excluded**

To Do #2 Write down the *attributes* (*other nouns)* associated with the *primary nouns*

To Do #3 Identify all the *verbs*

To Do #4 Identify which *primary nouns* are worked on by the *verbs*

To Do #5 Design a system using UML to handle this problem

A website tracks books and the kids that read them.

To Do #1 Identify all the ***primary nouns***

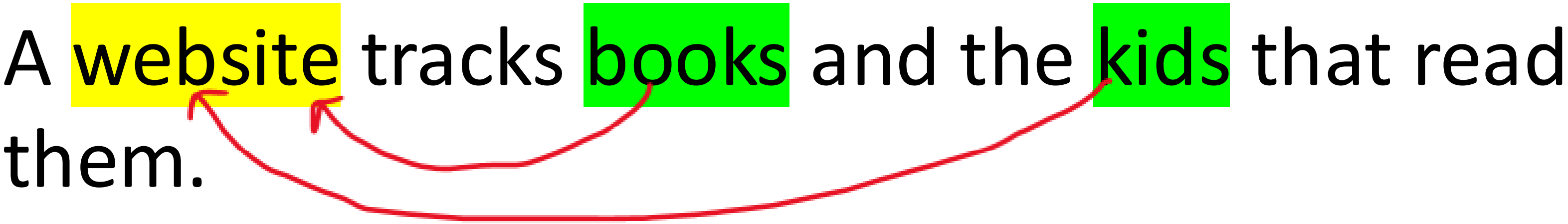
- A *primary noun* is a noun in the problem that has *attributes* *(other nouns*)
- Nouns that designate *actors* of the system (i.e. The *user* can click…) can be **excluded**

To Do #2 Write down the *attributes* (*other nouns)* associated with the *primary nouns*

To Do #3 Identify all the *verbs*

To Do #4 Identify which *primary nouns* are worked on by the *verbs*

To Do #5 Design a system using UML to handle this problem



A website tracks books and the kids that read them.

To Do #1 Identify all the *primary nouns*

- A *primary noun* is a noun in the problem that has *attributes* (*other nouns*)
- Nouns that designate *actors* of the system (i.e. The *user* can click…) can be **excluded**

To Do #2 Write down the *attributes* (*other nouns)* associated with the *primary nouns*

To Do #3 Identify all the *verbs*

To Do #4 Identify which *primary nouns* are worked on by the *verbs*

To Do #5 Design a system using UML to handle this problem

For each book the system stores the name and author.

To Do #1 Identify all the ***primary nouns***

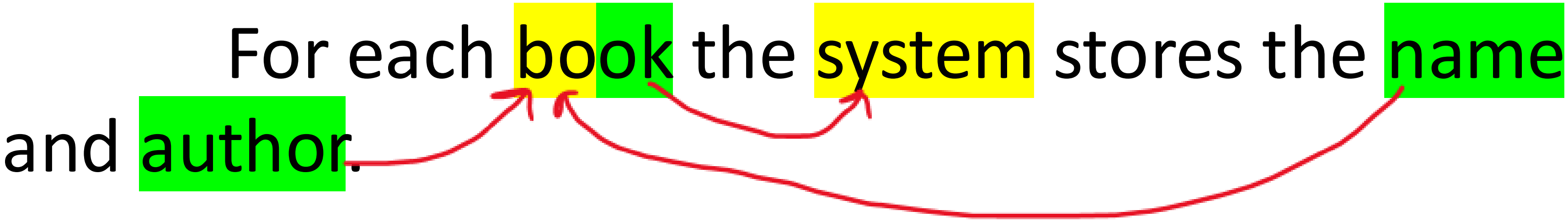
- A *primary noun* is a noun in the problem that has *attributes* *(other nouns*)
- Nouns that designate *actors* of the system (i.e. The *user* can click…) can be **excluded**

To Do #2 Write down the ***attributes*** (*other nouns)* associated with the *primary nouns*

To Do #3 Identify all the *verbs*

To Do #4 Identify which *primary nouns* are worked on by the *verbs*

To Do #5 Design a system using UML to handle this problem



For each book the system stores the name and author.

To Do #1 Identify all the *primary nouns*

- A *primary noun* is a noun in the problem that has *attributes (other nouns*)
- Nouns that designate *actors* of the system (i.e. The *user* can click…) can be **excluded**

To Do #2 Write down the *attributes* (*other nouns)* associated with the *primary nouns*

To Do #3 Identify all the *verbs*

To Do #4 Identify which *primary nouns* are worked on by the *verbs*

To Do #5 Design a system using UML to handle this problem

For each kid the system stores name and grade level.

To Do #1 Identify all the ***primary nouns***

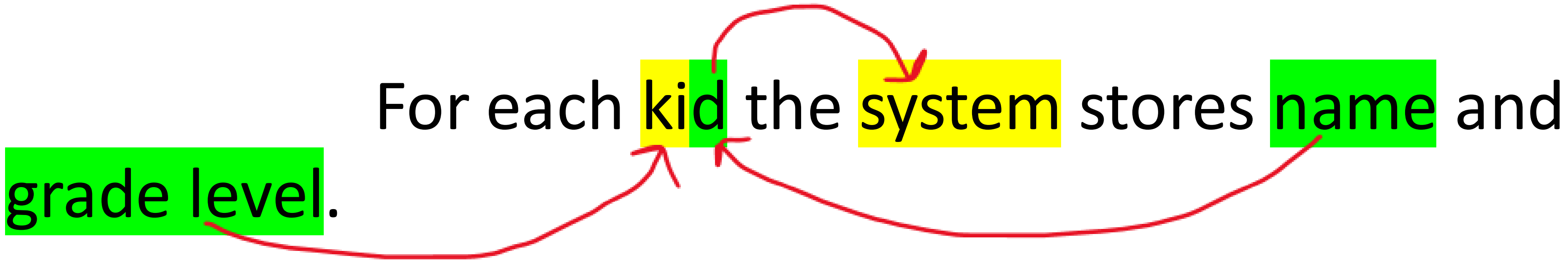
- A *primary noun* is a noun in the problem that has *attributes* *(other nouns*)
- Nouns that designate *actors* of the system (i.e. The *user* can click…) can be **excluded**

To Do #2 Write down the ***attributes*** (*other nouns)* associated with the *primary nouns*

To Do #3 Identify all the *verbs*

To Do #4 Identify which *primary nouns* are worked on by the *verbs*

To Do #5 Design a system using UML to handle this problem



For each kid the system stores name and grade level.

To Do #1 Identify all the *primary nouns*

- A *primary noun* is a noun in the problem that has *attributes (other nouns*)
- Nouns that designate *actors* of the system (i.e. The *user* can click…) can be **excluded**

To Do #2 Write down the *attributes* (*other nouns)* associated with the *primary nouns*

To Do #3 Identify all the *verbs*

To Do #4 Identify which *primary nouns* are worked on by the *verbs*

To Do #5 Design a system using UML to handle this problem

The ~~teacher~~ enters when a kid reads a particular book.

To Do #1 Identify all the *primary nouns*

- A *primary noun* is a noun in the problem that has *attributes (other nouns*)
- Nouns that designate *actors* of the system (i.e. The *user* can click…) can be **excluded**

To Do #2 Write down the *attributes* (*other nouns)* associated with the *primary nouns*

To Do #3 Identify all the *verbs*

To Do #4 Identify which *primary nouns* are worked on by the *verbs*

To Do #5 Design a system using UML to handle this problem

The teacher enters when a kid reads a particular book.

To Do #1 Identify all the ***primary nouns***

- A *primary noun* is a noun in the problem that has *attributes* *(other nouns*)
- Nouns that designate *actors* of the system (i.e. The *user* can click…) can be **excluded**

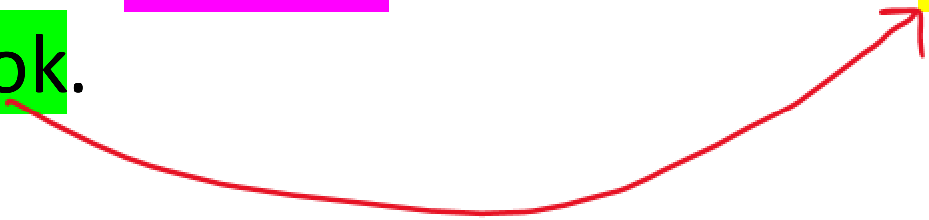
To Do #2 Write down the ***attributes*** (*other nouns)* associated with the *primary nouns*

To Do #3 Identify all the *verbs*

To Do #4 Identify which *primary nouns* are worked on by the *verbs*

To Do #5 Design a system using UML to handle this problem

The teacher enters when a kid reads a particular book.



To Do #1 Identify all the *primary nouns*

- A *primary noun* is a noun in the problem that has *attributes (other nouns*)
- Nouns that designate *actors* of the system (i.e. The *user* can click…) can be **excluded**

To Do #2 Write down the *attributes* (*other nouns)* associated with the *primary nouns*

To Do #3 Identify all the *verbs*

To Do #4 Identify which *primary nouns* are worked on by the *verbs*

To Do #5 Design a system using UML to handle this problem

It should be possible to print a report on a book that includes all kids who have read a particular book (with their grade level).

To Do #1 Identify all the *primary nouns*

- A *primary noun* is a noun in the problem that has *attributes* *(other nouns*)
- Nouns that designate *actors* of the system (i.e. The *user* can click…) can be **excluded**

To Do #2 Write down the *attributes* (*other nouns)* associated with the *primary nouns*

To Do #3 Identify all the *verbs*

To Do #4 Identify which *primary nouns* are worked on by the *verbs*

To Do #5 Design a system using UML to handle this problem

Is this a class to represent or just a command that needs to be run? Like printing to a screen for instance?

It should be possible to print a report on a book that includes all kids who have read a particular book (with their grade level).

To Do #1 Identify all the *primary nouns*

- A *primary noun* is a noun in the problem that has *attributes (other nouns*)
- Nouns that designate *actors* of the system (i.e. The *user* can click…) can be **excluded**

To Do #2 Write down the *attributes* (*other nouns)* associated with the *primary nouns*

To Do #3 Identify all the *verbs*

To Do #4 Identify which *primary nouns* are worked on by the *verbs*

To Do #5 Design a system using UML to handle this problem

It should be possible to print a report on a book that includes all kids who have read a particular book (with their grade level).

To Do #1 Identify all the *primary nouns*

- A *primary noun* is a noun in the problem that has *attributes (other nouns*)
- Nouns that designate *actors* of the system (i.e. The *user* can click…) can be **excluded**

To Do #2 Write down the *attributes* (*other nouns)* associated with the *primary nouns*

To Do #3 Identify all the *verbs*

To Do #4 Identify which *primary nouns* are worked on by the *verbs*

To Do #5 Design a system using UML to handle this problem

Is this a class to represent or just a command that needs to be run? Like printing to a screen for instance?

It should be possible to print a report on a kid that includes the books (with authors) a particular kid has read.

To Do #1 Identify all the *primary nouns*

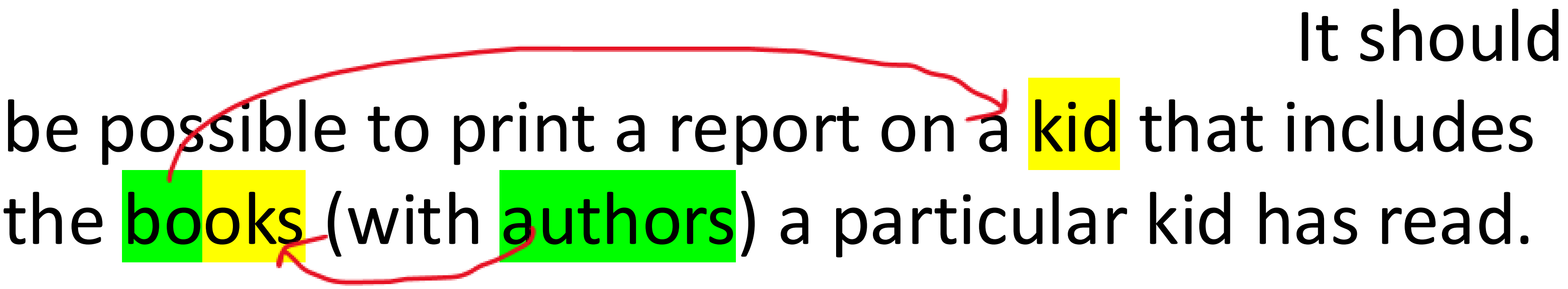
- A *primary noun* is a noun in the problem that has *attributes (other nouns*)
- Nouns that designate *actors* of the system (i.e. The *user* can click…) can be **excluded**

To Do #2 Write down the *attributes* (*other nouns)* associated with the *primary nouns*

To Do #3 Identify all the *verbs*

To Do #4 Identify which *primary nouns* are worked on by the *verbs*

To Do #5 Design a system using UML to handle this problem



It should be possible to print a report on a kid that includes the books (with authors) a particular kid has read.

To Do #1 Identify all the *primary nouns*

- A *primary noun* is a noun in the problem that has *attributes (other nouns*)
- Nouns that designate *actors* of the system (i.e. The *user* can click…) can be **excluded**

To Do #2 Write down the *attributes* (*other nouns)* associated with the *primary nouns*

To Do #3 Identify all the *verbs*

To Do #4 Identify which *primary nouns* are worked on by the *verbs*

To Do #5 Design a system using UML to handle this problem

A website tracks books and the kids that read them. For each book the system stores the name and author. For each kid the system stores name and grade level. The teacher enters when a kid reads a particular book. It should be possible to print a report on a book that includes all kids who have read a particular book (with their grade level). It should be possible to print a report on a kid that includes the books (with authors) a particular kid has read.

To Do #1 Identify all the ***primary nouns***

- A *primary noun* is a noun in the problem that has *attributes* (*other nouns*)
- Nouns that designate *actors* of the system (i.e. The *user* can click…) can be **excluded**

To Do #2 Write down the ***attributes*** (*other nouns)* associated with the *primary nouns*

To Do #3 Identify all the *verbs*

To Do #4 Identify which *primary nouns* are worked on by the *verbs*

To Do #5 Design a system using UML to handle this problem

A website tracks books and the kids that read them. For each book the system stores the name and author. For each kid the system stores name and grade level. The teacher enters when a kid reads a particular book. It should be possible to print a report on a book that includes all kids who have read a particular book (with their grade level). It should be possible to print a report on a kid that includes the books (with authors) a particular kid has read.

To Do #1 Identify all the *primary nouns*

- A *primary noun* is a noun in the problem that has *attributes* (*other nouns*)
- Nouns that designate *actors* of the system (i.e. The *user* can click…) can be **excluded**

To Do #2 Write down the *attributes* (*other nouns)* associated with the *primary nouns*

To Do #3 Identify all the *verbs*

To Do #4 Identify which *primary nouns* are worked on by the *verbs*

To Do #5 Design a system using UML to handle this problem

A website tracks books and the kids that read them. For each book the system stores the name and author. For each kid the system stores name and grade level. The ~~teacher~~ enters when a kid reads a particular book. It should be possible to print a ~~report~~ on a book that ~~includes~~ all kids who have read a particular book (with their grade level). It should be possible to print a ~~report~~ on a kid that ~~includes~~ the books (with authors) a particular kid has read.

To Do #1 Identify all the *primary nouns*

- A *primary noun* is a noun in the problem that has *attributes* (*other nouns*)
- Nouns that designate *actors* of the system (i.e. The *user* can click…) can be **excluded**

To Do #2 Write down the *attributes* (*other nouns)* associated with the *primary nouns*

To Do #3 Identify all the *verbs*

To Do #4 Identify which *primary nouns* are worked on by the *verbs*

To Do #5 Design a system using UML to handle this problem

A website tracks books and the kids that read them. For each book the system stores the name and author. For each kid the system stores name and grade level. The teacher enters when a kid reads a particular book. It should be possible to print a report on a book that includes all kids who have read a particular book (with their grade level). It should be possible to print a report on a kid that includes the books (with authors) a particular kid has read.

To Do #1 Identify all the *primary nouns* ?

- A *primary noun* is a noun in the problem that has *attributes* *(other nouns*)
- Nouns that designate *actors* of the system (i.e. The *user* can click…) can be **excluded**

To Do #2 Write down the *attributes* (*other nouns)* associated with the *primary nouns*

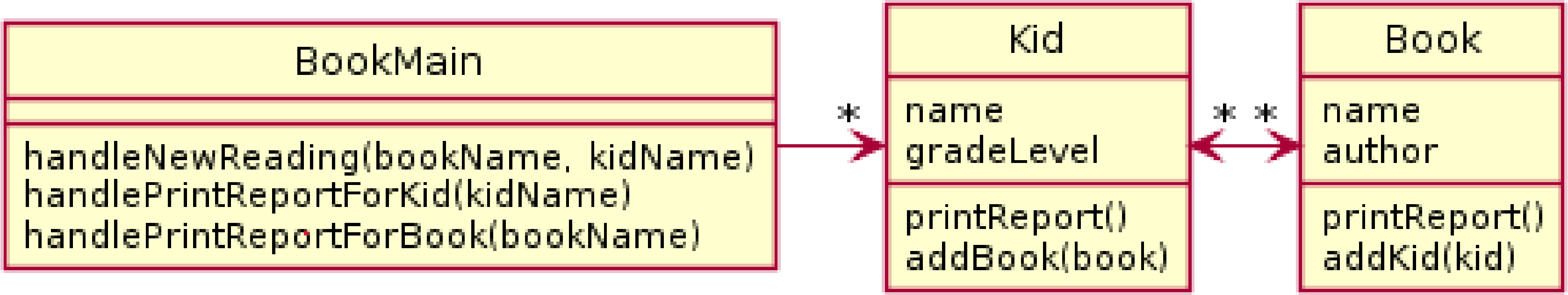
To Do #3 Identify all the *verbs*

To Do #4 Identify which *primary nouns* are worked on by the *verbs*

To Do #5 Design a system using UML to handle this problem

A website tracks books and the kids that read them. For each book the system stores the name and author. For each kid the system stores name and grade level. The teacher enters when a kid reads a particular book. It should be possible to print a report on a book that includes all kids who have read a particular book (with their grade level). It should be possible to print a report on a kid that includes the books (with authors) a particular kid has read.

Bad Design A

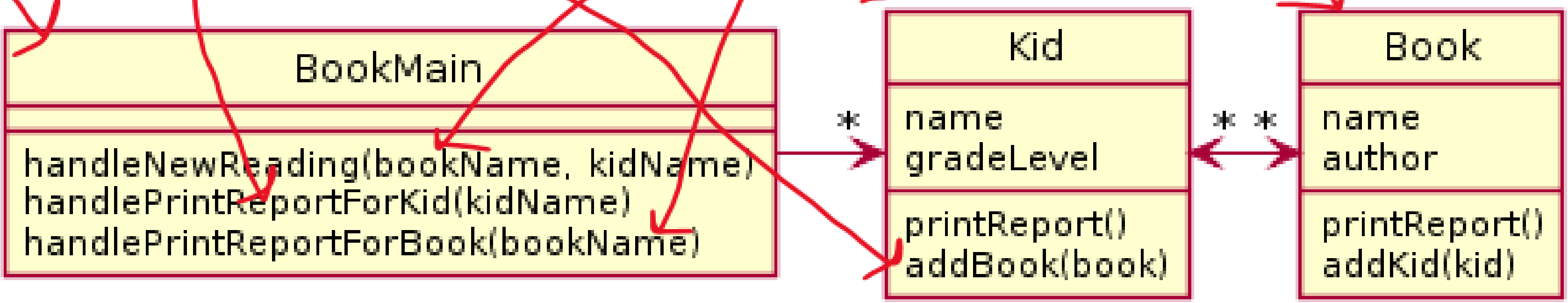

BookMain
handleNewReading(bookName, kidName)
handlePrintReportForKid(kidName)
handlePrintReportForBook(bookName)
*
Kid
name
gradeLevel
printReport()
addBook(book)
* *
Book
name
author
printReport()
addKid(kid)

If you do this all well, then you should hopefully be able to make classes where:

- Each *primary noun* becomes a Class
- Each *attribute* (*other noun*) becomes a field for its respective class
- Each *verb* becomes the method for the respective class

A website tracks books and the kids that read them. For each book the system stores the name and author. For each kid the system stores name and grade level. The teacher enters when a kid reads a particular book. It should be possible to print a report on a book that includes all kids who have read a particular book (with their grade level). It should be possible to print a report on a kid that includes the books (with authors) a particular kid has read.

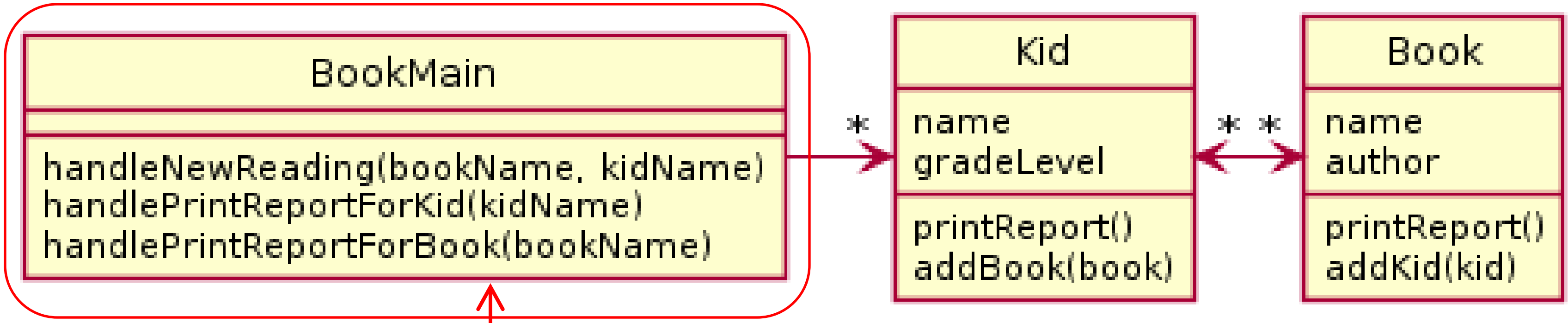
Bad Design A


BookMain
handleNewReading(bookName, kidName)
handlePrintReportForKid(kidName)
handlePrintReportForBook(bookName)
*
Kid
name
gradeLevel
printReport()
addBook(book)
* *
Book
name
author
printReport()
addKid(kid)

If you do this all well, then you should hopefully be able to make classes where:

- Each *primary noun* becomes a Class
- Each *attribute* (*other noun*) becomes a field for its respective class
- Each *verb* becomes the method for the respective class

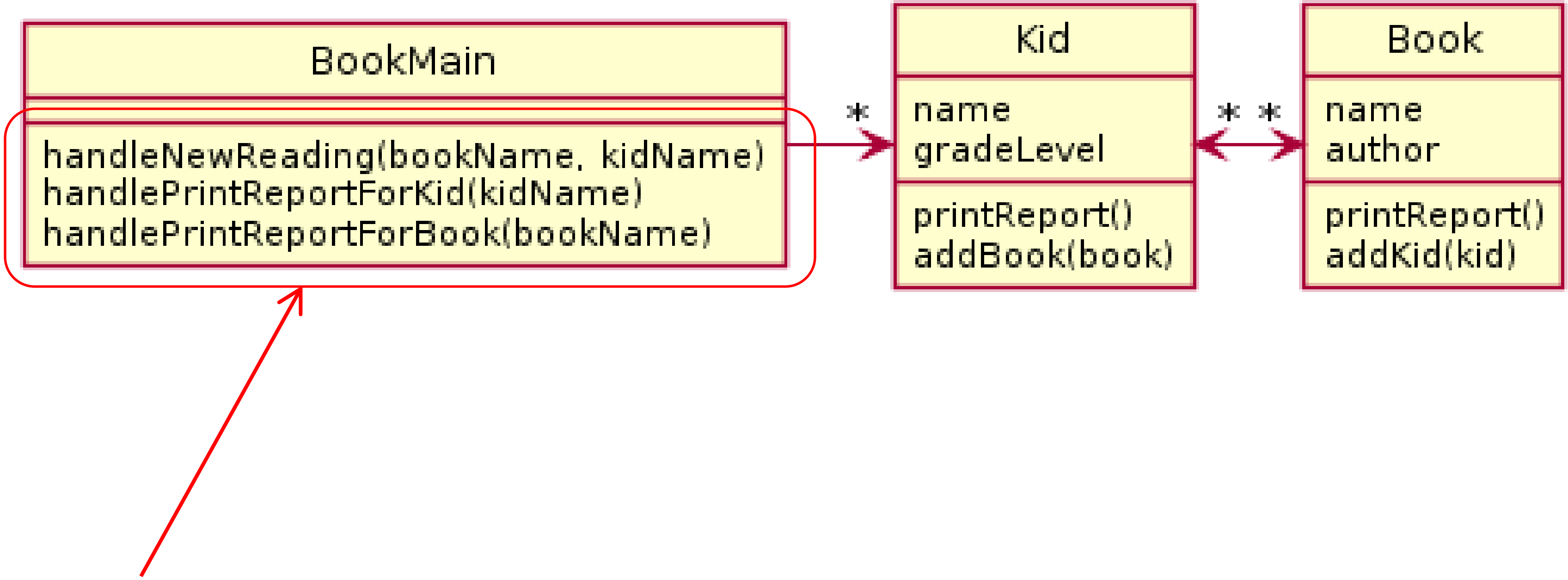
Good parts of the design - Main class


BookMain
handleNewReading(bookName, kidName)
handlePrintReportForKid(kidName)
handlePrintReportForBook(bookName)
*
Kid
name
gradeLevel
printReport()
addBook(book)
*
*
Book
name
author
printReport()
addKid(kid)

- Every program starts somewhere, and any design should make clear where the starting point is. In our class, we will name the starting point class SomethingMain

We *implicitly* assume there exists:
constructors as needed
getters and setters as needed

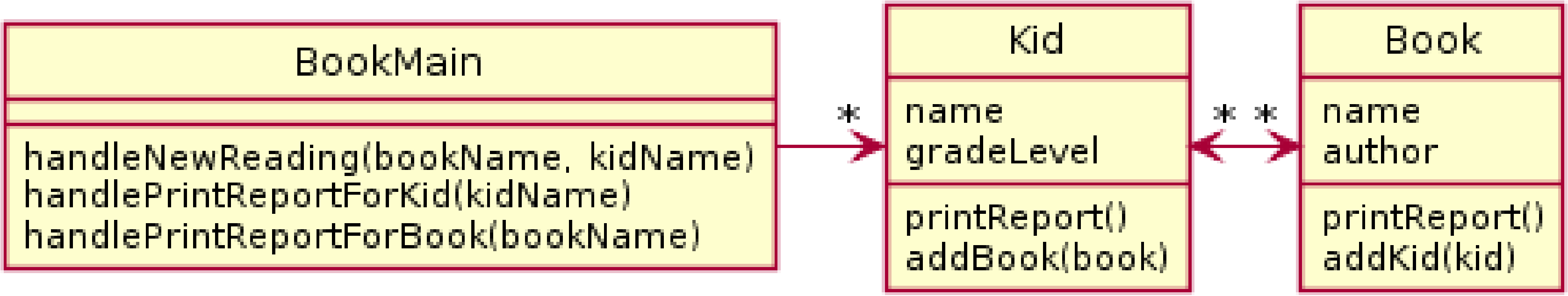
Good parts of the design – “handle” methods


BookMain
handleNewReading(bookName, kidName)
handlePrintReportForKid(kidName)
handlePrintReportForBook(bookName)
*
Kid
name
gradeLevel
printReport()
addBook(book)
* *
Book
name
author
printReport()
addKid(kid)

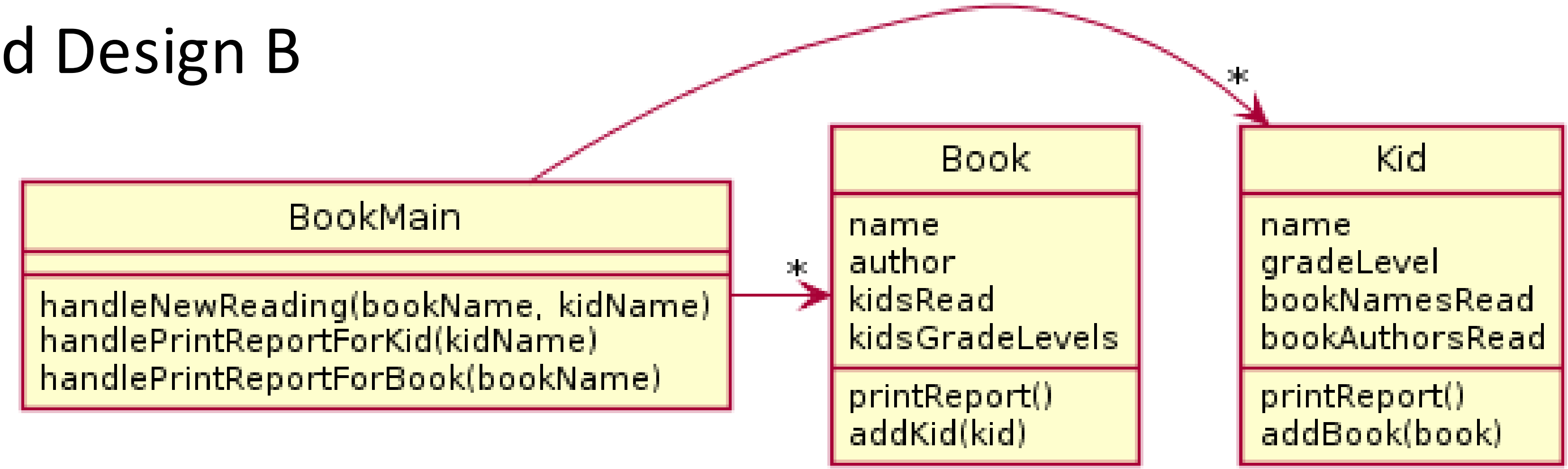
- In our very simple designs, this class also deals with user input
- *handle* methods will have special meaning for us, as they will represent places where user commands enter the system
- Examine the UML diagram and use your analysis skills in order to determine:
  1. If what is required by the *story problem* can be stored by the design presented in the UML diagram
  2. If you can implement the *handle* methods in the UML *Main* class given the data stored in the different UML class diagram

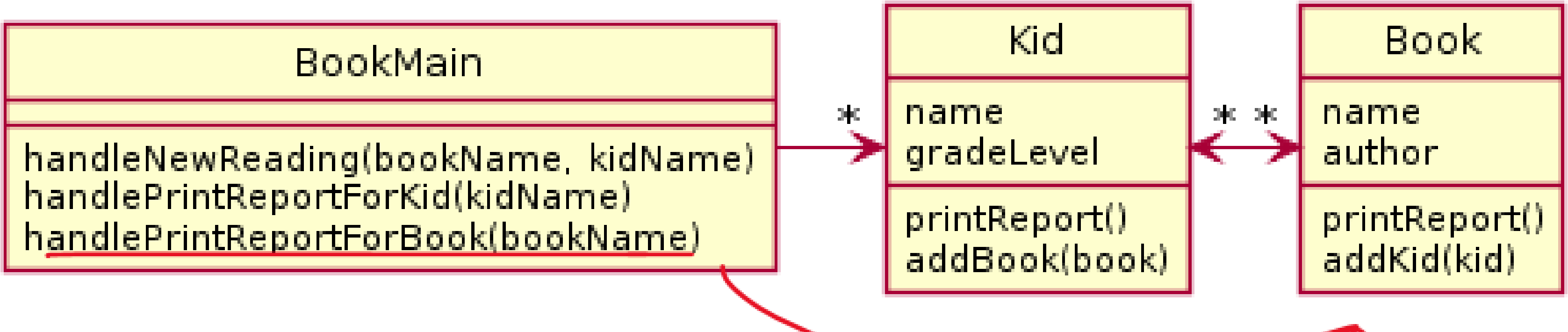
A website tracks books and the kids that read them. For each book the system stores the name and author. For each kid the system stores name and grade level. The teacher enters when a kid reads a particular book. It should be possible to print a report on a book that includes all kids who have read a particular book (with their grade level). It should be possible to print a report on a kid that includes the books (with authors) a particular kid has read.

Bad Design A


BookMain
handleNewReading(bookName, kidName)
handlePrintReportForKid(kidName)
handlePrintReportForBook(bookName)
*
Kid
name
gradeLevel
printReport()
addBook(book)
* *
Book
name
author
printReport()
addKid(kid)

Bad Design B





BookMain

handleNewReading(bookName, kidName)
handlePrintReportForKid(kidName)
handlePrintReportForBook(bookName)

Kid

name
gradeLevel

printReport()
addBook(book)

Book

name
author

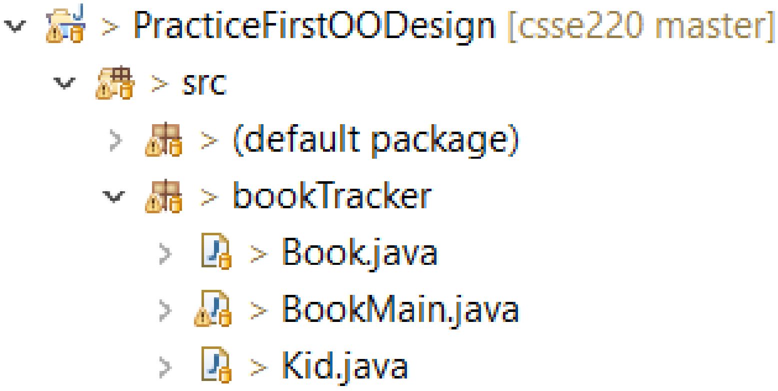
printReport()
addKid(kid)

This design does not function. There is no (sane) way to look up a book for printing a report or for associating with a Kid.

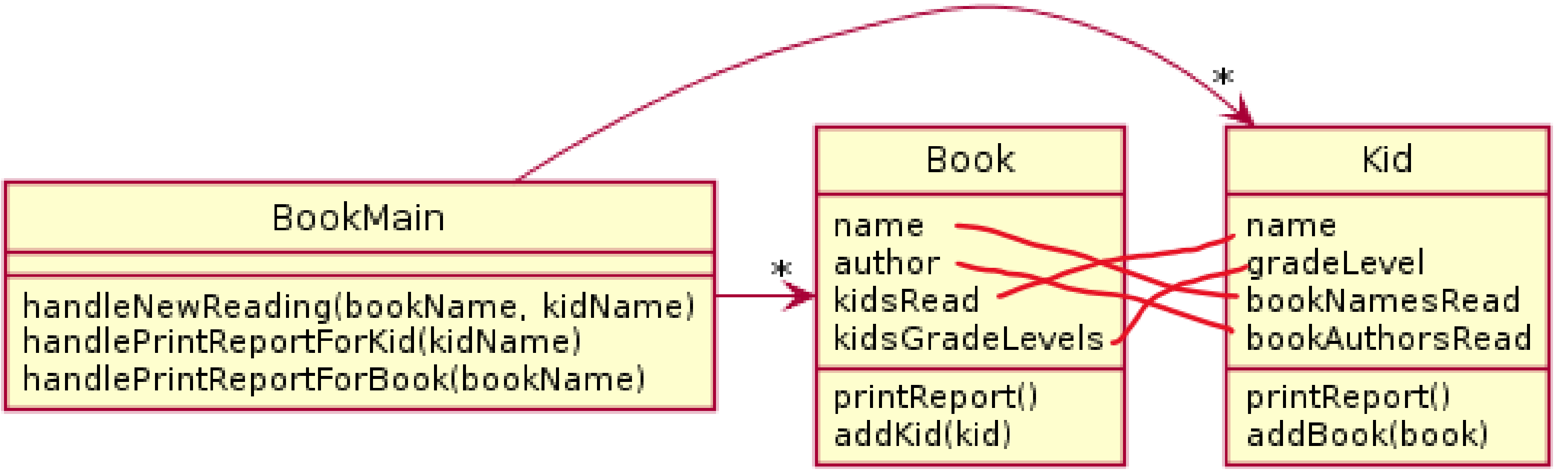
?

See the code here for this POOR design!

Warning it is buggy and the code is UGLY



```
PracticeFirstOODesign [csse220 master]
  src
    (default package)
    bookTracker
      Book.java
      BookMain.java
      Kid.java
```



This design functions but there is a very large amount of duplication – which in general we want to avoid.

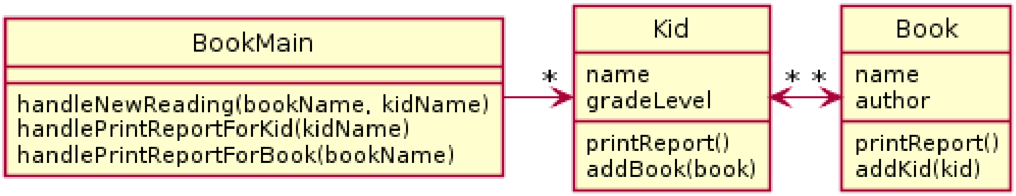
In particular, the author/title information in the kid is duplicated and the name/grade level information in the book is duplicated.

What would be a better design?

A website tracks books and the kids that read them. For each book the system stores the name and author. For each kid the system stores name and grade level. The teacher enters when a kid reads a particular book. It should be possible to print a report on a book that includes all kids who have read a particular book (with their grade level). It should be possible to print a report on a kid that includes the books (with authors) a particular kid has read.

Problematic designs for reference:

Bad Design A



BookMain

handleNewReading(bookName, kidName)
handlePrintReportForKid(kidName)
handlePrintReportForBook(bookName)

Kid

name
gradeLevel

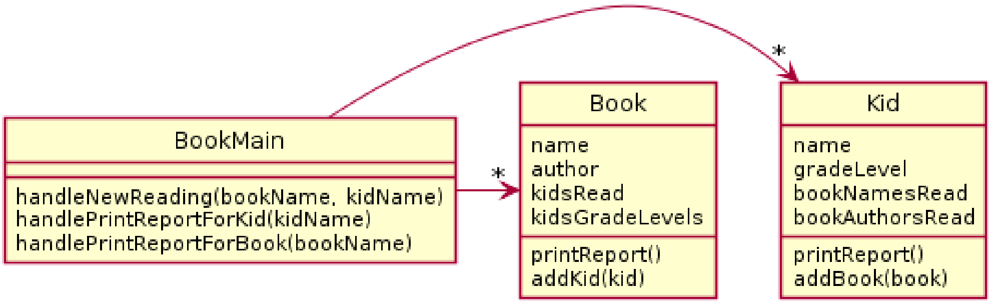
printReport()
addBook(book)

Book

name
author

printReport()
addKid(kid)

Bad Design B



BookMain

handleNewReading(bookName, kidName)
handlePrintReportForKid(kidName)
handlePrintReportForBook(bookName)

Book

name
author
kidsRead
kidsGradeLevels

printReport()
addKid(kid)

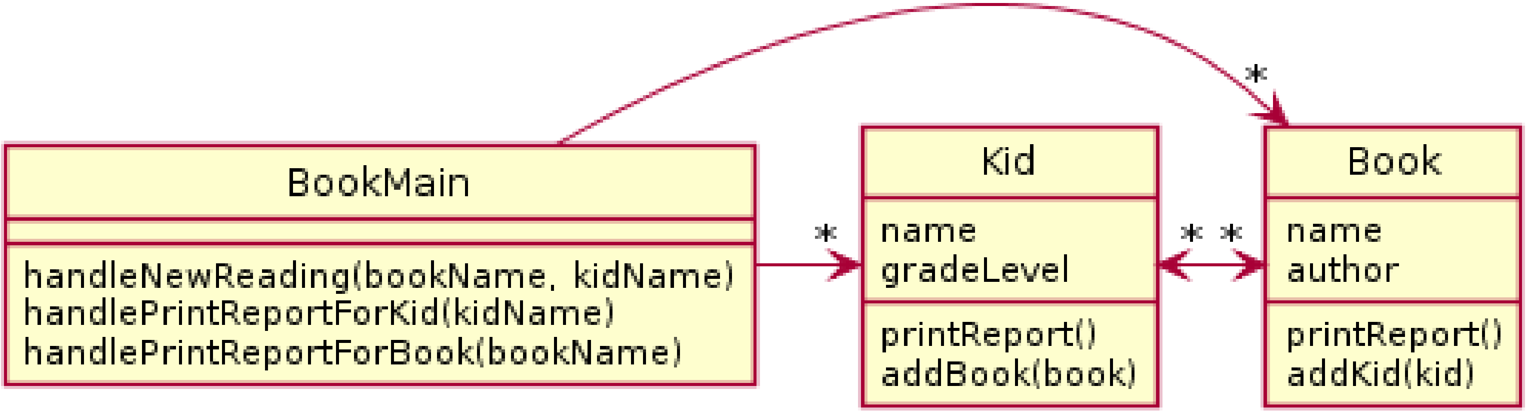
Kid

name
gradeLevel
bookNamesRead
bookAuthorsRead

printReport()
addBook(book)

A website tracks books and the kids that read them. For each book the system stores the name and author. For each kid the system stores name and grade level. The teacher enters when a kid reads a particular book. It should be possible to print a report on a book that includes all kids who have read a particular book (with their grade level). It should be possible to print a report on a kid that includes the books (with authors) a particular kid has read.

A Potential Solution


BookMain
handleNewReading(bookName, kidName)
handlePrintReportForKid(kidName)
handlePrintReportForBook(bookName)
Kid
name
gradeLevel
printReport()
addBook(book)
Book
name
author
printReport()
addKid(kid)
*
*
* *

If you do this all well, then you should hopefully be able to make classes where:

- Each *primary noun* becomes a Class
- Each *attribute* (*other noun*) becomes a field for its respective class
- Each *verb* becomes the method for the respective class

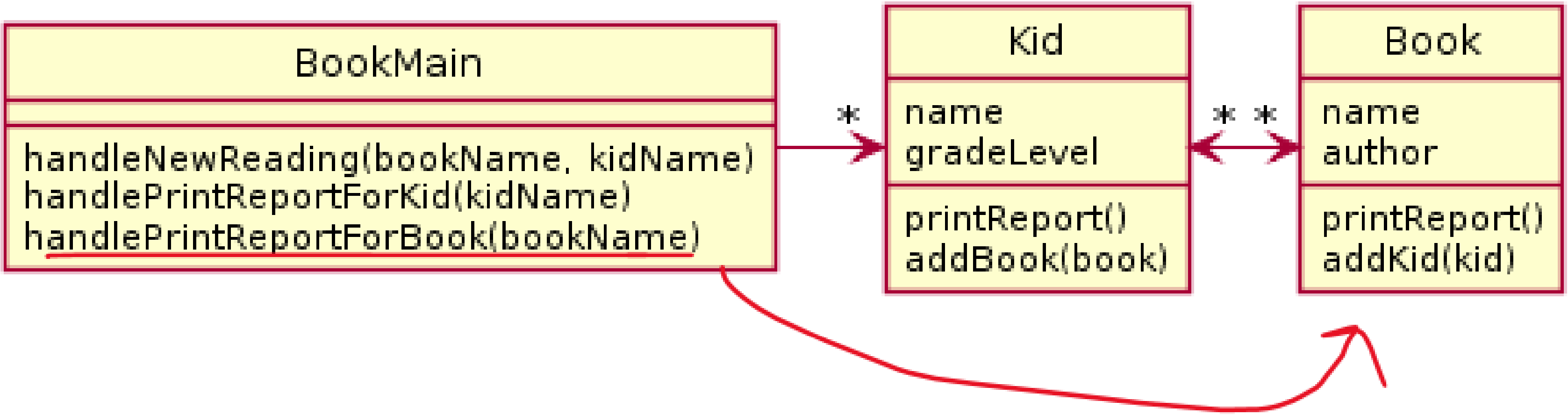
In most cases non-workable design is caused by…

- Not carefully analyzing the story problem.
- Not carefully mapping the story problem to the proposed design (e.g., not noticing that each kid reads several books, not just one)
- Not thinking about how specific required features might be implemented (e.g., how can we print a book report if we don’t have access to the book objects?)
- Duplicating data (e.g., what does it matter if we store a copy of the author and title for every kid that reads the book)

As times allows

- If there is enough time during class you can work through an example of Refactoring
- This is a great way to see ugly code and the improvements that come with an improve design
- The solution will be:
  - More efficient
  - Fewer lines of code
  - Easier to understand
  - Easier to re-use or extend
  - Offer improved functionality (existing code has bugs)

Refactoring Coding Exercise

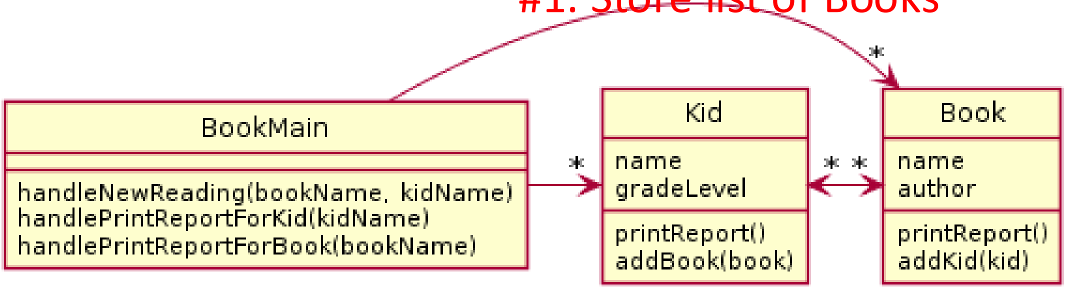

BookMain
handleNewReading(bookName, kidName)
handlePrintReportForKid(kidName)
handlePrintReportForBook(bookName)
*
Kid
name
gradeLevel
printReport()
addBook(book)
* *
Book
name
author
printReport()
addKid(kid)

This design does not function. There is no (sane) way to look up a book for printing a report or for associating with a Kid.

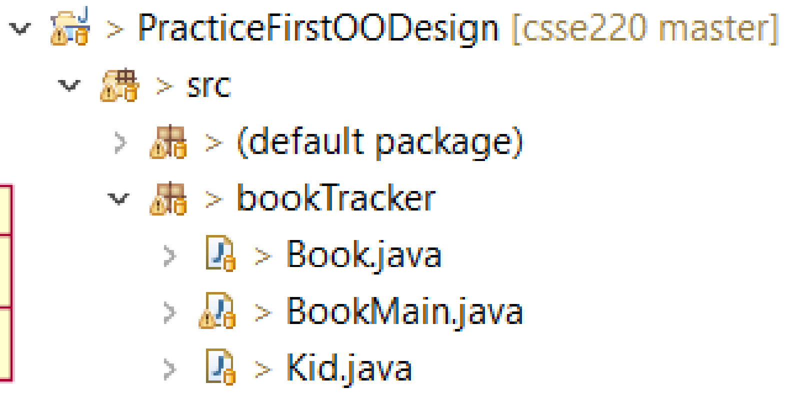
We gave you this starting code

Refactor it to improve it!

#1. Store list of Books

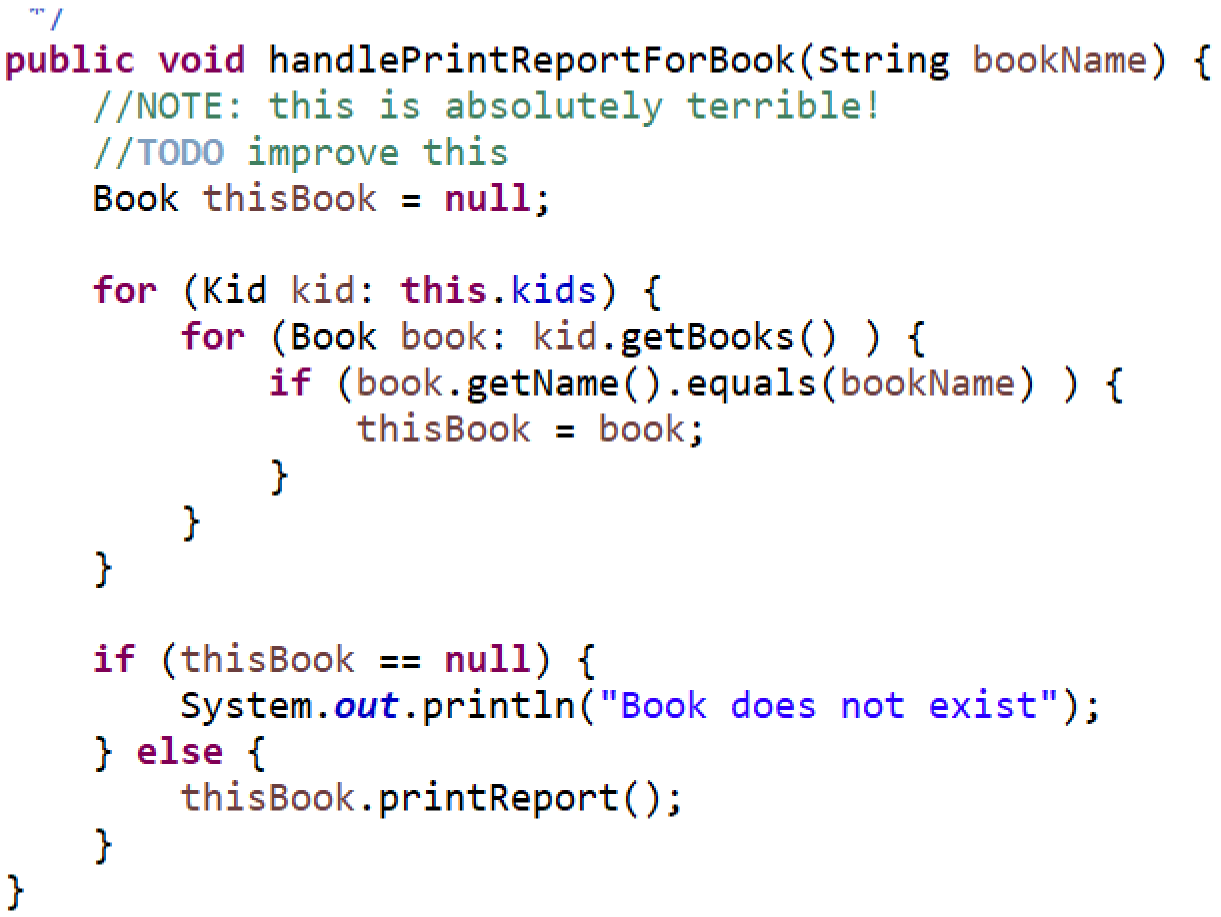

BookMain
handleNewReading(bookName, kidName)
handlePrintReportForKid(kidName)
handlePrintReportForBook(bookName)
*
Kid
name
gradeLevel
printReport()
addBook(book)
* *
*
Book
name
author
printReport()
addKid(kid)

#2. Update methods in BookMain



- > PracticeFirstOODesign [csse220 master]
  - > src
    - > (default package)
    - > bookTracker
      - > Book.java
      - > BookMain.java
      - > Kid.java

How bad is it? To print a book report…



```
 */
public void handlePrintReportForBook(String bookName) {
    //NOTE: this is absolutely terrible!
    //TODO improve this
    Book thisBook = null;

    for (Kid kid: this.kids) {
        for (Book book: kid.getBooks() ) {
            if (book.getName().equals(bookName) ) {
                thisBook = book;
            }
        }
    }

    if (thisBook == null) {
        System.out.println("Book does not exist");
    } else {
        thisBook.printReport();
    }
}
```