

Accueil ► SI - Sciences Informatiques ► SI3 ► Intro POO ► Stuff to do - unevaluated ► Understanding class definitions - Book

Commencé le	mardi 3 octobre 2017, 19:39
État	Terminé
Terminé le	mardi 3 octobre 2017, 23:55
Temps mis	4 heures 15 min
En retard	4 heures 15 min
Points	8,30/9,00
Note	18,44 sur 20,00 (92%)

Description

Below is the outline for a Book class, which can be found in the bookexercise package. The outline already defines two fields and a constructor to initialize the fields. In this exercise, you will add further features to the class outline. Add two accessor methods to the class – getAuthor and getTitle – that return the author and title fields as their respective results. Test your class by creating some instances and calling the methods.

```
/**
 * A class that maintains information on a book.
 * This might form part of a larger application such
 * as a library system, for instance.
 *
 * @author Insert your name here.
 */
class Book {
    // The fields.
    private String author;
    private String title;
    /**
     * Set the author and title fields when this object
     * is constructed.
     */
    Book(String bookAuthor, String bookTitle) {
        author = bookAuthor;
        title = bookTitle;
    }
    Add the methods here.
}
```

Question 1

Correct

Note de 1,00 sur
1,00

Add two methods, `printAuthor` and `printTitle`, to the outline `Book` class. These should print the `author` and `title` fields, respectively, to the terminal window.

Copy your entire `Book` class and paste it into the Answer box. Then Check your answer.

For example:

Test	Result
<pre>Book book = new Book("J. Random Luser", "Objects First Without Java"); book.printAuthor();</pre>	J. Random Luser

Réponse:

```

1 class Book {
2     // The fields.
3     private String author;
4     private String title;
5     /**
6      * Set the author and title fields when this object
7      * is constructed.
8      */
9     Book(String bookAuthor, String bookTitle) {
10         author = bookAuthor;
11         title = bookTitle;
12     }
13
14     void printAuthor(){
15         System.out.println(author);
16     }
17
18     void printTitle(){
19         System.out.println(title);
20     }
21 }
```

Vérifier

	Test	Expected
✓	<pre>Book book = new Book("J. Random Luser", "Objects First Without Java"); book.printTitle();</pre>	Objects First Without
✓	<pre>Book book = new Book("J. Random Luser", "Objects First Without Java"); book.printAuthor();</pre>	J. Random Luser

Passed all tests! ✓

Correct

Note pour cet envoi : 1,00/1,00.

Question 2

Correct

Note de 1,00 sur
1,00

Add a further field, `pages`, to the `Book` class to store the number of pages. This should be of type `int`, and its initial value should be passed to the single constructor, along with the author and title strings. Include an appropriate `getPages` accessor method for this field.

Copy your entire `Book` class and paste it into the Answer box. Then Check your answer.

For example:

Test	Result
<pre>Book book = new Book("U.N. Known", "50 Shades of Grapes", 50); System.out.println(book.getPages());</pre>	50

Réponse:

```
1 class Book {
2     // The fields.
3     private String author;
4     private String title;
5     private int pages;
6     /**
7      * Set the author and title fields when this object
8      * is constructed.
9      */
10    Book(String bookAuthor, String bookTitle, int bookPages) {
11        author = bookAuthor;
12        title = bookTitle;
13        pages = bookPages;
14    }
15
16    void printAuthor(){
17        System.out.println(author);
18    }
19 }
```

Vérifier

	Test	Expected	Got	
✓	<pre>Book book = new Book("U.N. Known", "Negative Shades of Grapes", -100); System.out.println(book.getPages());</pre>	-100	-100	✓
✓	<pre>Book book = new Book("U.N. Known", "Zero Shades of Grapes", 0); System.out.println(book.getPages());</pre>	0	0	✓
✓	<pre>Book book = new Book("U.N. Known", "50 Shades of Grapes", 50); System.out.println(book.getPages());</pre>	50	50	✓

Passed all tests! ✓

Correct

Note pour cet envoi : 1,00/1,00.

Question 3

Correct

Note de 1,00 sur
1,00

Add a method, `toString`, to the `Book` class. This should return details of the title, author and pages in the following format, eg,

```
Title: Tweet Tweet
Author: Donald Bloviator
Pages: 1234
```

where the `title`, `author`, and `pages` fields contain "Tweet Tweet", "Donald Bloviator", 1234 respectively.

Note: The string "`\n`" will print as a new line.

Note: The compiler will whine that the access level of `toString` must be public. You'll see why...in a couple of weeks. If you're really interested, see the documentation for the class **Object**.

Copy your entire `Book` class and paste it into the Answer box. Then Check your answer.

For example:

Test	Result
<pre>Book book = new Book("Steve Jawb", "The Book of Jobs", 666); System.out.print(book);</pre>	<pre>Title: The Book of Jobs Author: Steve Jawb Pages: 666</pre>

Réponse:

```
1 class Book {
2     // The fields.
3     private String author;
4     private String title;
5     private int pages;
6     /**
7      * Set the author and title fields when this object
8      * is constructed.
9      */
10    Book(String bookAuthor, String bookTitle, int bookPages) {
11        author = bookAuthor;
12        title = bookTitle;
13        pages = bookPages;
14    }
15
16    void printAuthor(){
17        System.out.println(author);
18    }
19 }
```

	Test	Expected
✓	<pre>Book book = new Book("Indonesia Tourist Board", "Java - the Forgotten Island", 327); System.out.print(book);</pre>	<pre>Title: Author: Pages:</pre>
✓	<pre>Book book = new Book("Steve Jawb", "The Book of Jobs", 666); System.out.print(book);</pre>	<pre>Title: Author: Pages:</pre>

Passed all tests! ✓

Correct

Note pour cet envoi : 1,00/1,00.

Question 4

Correct

Note de 1,00 sur
1,00

Add a method, `printDetails`, to the `Book` class, calling `toString` and printing details of `title`, `author` and `pages` to the terminal window.

Copy your entire `Book` class and paste it into the Answer box. Then Check your answer.

For example:

Test	Result
<pre>Book book = new Book("Me Myself AndI", "Getting Tired of Inventing These Titles", 7); book.printDetails();</pre>	Title Autho Pages

Réponse:

```

1 class Book {
2     // The fields.
3     private String author;
4     private String title;
5     private int pages;
6     /**
7      * Set the author and title fields when this object
8      * is constructed.
9      */
10    Book(String bookAuthor, String bookTitle, int bookPages) {
11        author = bookAuthor;
12        title = bookTitle;
13        pages = bookPages;
14    }
15
16    void printAuthor(){
17        System.out.println(author);
18    }
19 }
```

Vérifier

	Test	Expected
✓	<pre>Book book = new Book("Me Myself AndI", "Getting Tired of Inventing These Titles", 7); book.printDetails();</pre>	Title: Author: Pages:

Passed all tests! ✓

Correct

Note pour cet envoi : 1,00/1,00.

Question 5

Correct

Note de 1,00 sur
1,00

Add a further field, `refNumber`, to the `Book` class. This field can store a reference number for a library, for example. It should be of type `String` and initialized to the zero length string (`""`) in the constructor as its initial value is not passed in a parameter to the constructor. Instead, define a mutator for it with the following signature:

```
void setRefNumber(String ref)
```

The body of this method should assign the value of the parameter to the `refNumber` field. Add a corresponding `getRefNumber` accessor to help you check that the mutator works correctly.

Note: Use

```
aString.equals(anotherString)
```

to check equality of strings. Do not use

```
aString == anotherString
```

You'll see why next week. Meanwhile you can look at the documentation for the class **`String`**.

Copy your entire `Book` class and paste it into the Answer box. Then Check your answer.

For example:

Test	Result
<pre>final String ref = "?"; Book book = new Book("Anony Mouse", "The Book with No Name", 289); book.setRefNumber(ref); System.out.println(ref.equals(book.getRefNumber()));</pre>	true

Réponse:

```

1 class Book {
2     // The fields.
3     private String author;
4     private String title;
5     private int pages;
6     private String refNumber;
7     /**
8      * Set the author and title fields when this object
9      * is constructed.
10    */
11    Book(String bookAuthor, String bookTitle, int bookPages) {
12        author = bookAuthor;
13        title = bookTitle;
14        pages = bookPages;
15        refNumber = "";
16    }
17
18    void printAuthor(){
19
20    }
21 }

```

Vérifier

	Test	Expected	Got	
✓	<pre>final String ref = "?"; Book book = new Book("Anony Mouse", "The Book with No Name", 289); book.setRefNumber(ref); System.out.println("wazzat?".equals(book.getRefNumber()));</pre>	false	false	✓
✓	<pre>final String ref = "?"; Book book = new Book("Anony Mouse", "The Book with No Name", 289); book.setRefNumber(ref); System.out.println(ref.equals(book.getRefNumber()));</pre>	true	true	✓

Passed all tests! ✓

Correct

Note pour cet envoi : 1,00/1,00.

Question 6

Correct

Note de 1,00 sur 1,00

Modify your `toString` method to include returning the reference number. However, the method should return the reference number only if it has been set, ie, the `refNumber` string has a nonzero length. Hint: Use a conditional statement whose test calls the `length` method on the `refNumber` string.

Copy your entire Book class and paste it into the Answer box. Then Check your answer.

For example:

Test	Result
<pre>final String ref = ""; Book book = new Book("Hugh Ever", "Whatever", 312); book.setRefNumber(ref); System.out.println(book);</pre>	Title: Whatever Author: Hugh Ever Pages: 312

<pre>final String ref = "random reference"; Book book = new Book("Hugh Ever", "Whatever", 312); book.setRefNumber(ref); System.out.println(book);</pre>	<pre>Title: Whatever Author: Hugh Ever Pages: 312 Reference: random reference</pre>
---	---

Réponse:

```
1 class Book {
2     // The fields.
3     private String author;
4     private String title;
5     private int pages;
6     private String refNumber;
7     /**
8      * Set the author and title fields when this object
9      * is constructed.
10    */
11    Book(String bookAuthor, String bookTitle, int bookPages) {
12        author = bookAuthor;
13        title = bookTitle;
14        pages = bookPages;
15        refNumber = "";
16    }
17
18    void printAuthor(){
19
20    }
```

Vérifier

	Test	Expected
✓	<pre>Book book = new Book("Hugh Ever", "Whatever", 312); System.out.println(book);</pre>	<pre>Title: Whatever Author: Hugh Ever Pages: 312</pre>
✓	<pre>final String ref = "another random reference"; Book book = new Book("E. MacCron", "Philosophy of the Big Mac", 29); book.setRefNumber(ref); System.out.println(book);</pre>	<pre>Title: Philosophy of th Author: E. MacCron Pages: 29 Reference: another rand</pre>
✓	<pre>final String ref = ""; Book book = new Book("Hugh Ever", "Whatever", 312); book.setRefNumber(ref); System.out.println(book);</pre>	<pre>Title: Whatever Author: Hugh Ever Pages: 312</pre>
✓	<pre>final String ref = "random reference"; Book book = new Book("Hugh Ever", "Whatever", 312); book.setRefNumber(ref); System.out.println(book);</pre>	<pre>Title: Whatever Author: Hugh Ever Pages: 312 Reference: random refer</pre>

Passed all tests! ✓

Correct

Note pour cet envoi : 1,00/1,00.

Question 7

Correct

Note de 0,30 sur 1,00

Modify your `setRefNumber` mutator so that it sets the `refNumber` field only if the parameter is a string of at least three characters. If it is less than three, then print an error message, eg,

```
book.setRefNumber("??");
Gotcha! RefNumber must be of length >= 3, but was 2
```

and leave the field unchanged.

Copy your entire `Book` class and paste it into the Answer box. Then Check your answer.

For example:

Test	Result
<pre>final String goodRef = "random reference"; final String badRef = "??"; Book book = new Book("Anony Mouse", "The Book with No Name", 289); book.setRefNumber(goodRef); System.out.println(goodRef.equals(book.getRefNumber())); book.setRefNumber(badRef); System.out.println(goodRef.equals(book.getRefNumber()));</pre>	<pre>true Gotcha! RefNumber must b true</pre>
<pre>final String ref = "random reference"; Book book = new Book("Hugh Ever", "Whatever", 312); book.setRefNumber(ref); System.out.println(book);</pre>	<pre>Title: Whatever Author: Hugh Ever Pages: 312 Reference: random refere</pre>

Réponse:

```

1 class Book {
2     // The fields.
3     private String author;
4     private String title;
5     private int pages;
6     private String refNumber;
7     /**
8      * Set the author and title fields when this object
9      * is constructed.
10    */
11    Book(String bookAuthor, String bookTitle, int bookPages) {
12        author = bookAuthor;
13        title = bookTitle;
14        pages = bookPages;
15        refNumber = "";
16    }
17
18    void printAuthor(){
19        // ...
20    }
21 }
```

Vérifier

	Test	Expected
✓	<pre>final String ref = ""; Book book = new Book("Hugh Ever", "Whatever", 312); System.out.println(book); book.setRefNumber(ref); System.out.println(book);</pre>	<pre>Title: Whatever Author: Hugh Ever Pages: 312 Gotcha! RefNumber must be Title: Whatever Author: Hugh Ever Pages: 312</pre>
✓	<pre>final String goodRef = "random reference"; final String badRef = "??"; Book book = new Book("Hugh Ever", "Whatever", 312); book.setRefNumber(goodRef); System.out.println(book); book.setRefNumber(badRef); System.out.println(book);</pre>	<pre>Title: Whatever Author: Hugh Ever Pages: 312 Reference: random referen Gotcha! RefNumber must be Title: Whatever Author: Hugh Ever Pages: 312 Reference: random referen</pre>
✓	<pre>final String goodRef = "random reference"; final String badRef = "??"; Book book = new Book("Anony Mouse", "The Book with No Name", 289); book.setRefNumber(goodRef); System.out.println(goodRef.equals(book.getRefNumber())); book.setRefNumber(badRef);</pre>	<pre>true Gotcha! RefNumber must be true</pre>

	<code>System.out.println(goodRef.equals(book.getRefNumber()));</code>	
✓	<pre>final String goodRef = "random reference"; final String badRef = "??"; Book book = new Book("Anony Mouse", "The Book with No Name", 289); book.setRefNumber(goodRef); System.out.println(goodRef.equals(book.getRefNumber())); book.setRefNumber(badRef); System.out.println(goodRef.equals(book.getRefNumber()));</pre>	<pre>true Gotcha! RefNumber must be true</pre>
✓	<pre>final String ref = "random reference"; Book book = new Book("Hugh Ever", "Whatever", 312); book.setRefNumber(ref); System.out.println(book);</pre>	<pre>Title: Whatever Author: Hugh Ever Pages: 312 Reference: random referen</pre>

Passed all tests! ✓

Correct

Points pour cet envoi : 1,00/1,00. En tenant compte des tentatives précédentes, cela donne **0,30/1,00**.

Question 8

Correct

Note de 1,00 sur
1,00

Add a further integer field, `borrowed`, to the `Book` class. This keeps a count of the number of times a book has been borrowed. Add a mutator, `borrow`, to the class. This should update the field by 1 each time it is called. Include an accessor, `getBorrowed`, that returns the value of this new field as its result. Modify `toString` so that it includes the value of this field, eg,

```
Title: The Book of Jobs
Author: Steve Jawb
Pages: 666
Reference: SJ000
This book has been borrowed 7 times.
```

Copy your entire `Book` class and paste it into the Answer box. Then Check your answer.

For example:

Test	Result
<pre>final String ref = "THX1138"; Book book = new Book("George Lacaille", "Milking It", 26745); book.setRefNumber(ref); book.borrow(); book.printDetails();</pre>	<pre>Title: Milking It Author: George Lacaille Pages: 26745 Reference: THX1138 This book has been borrowed 1</pre>
<pre>final String ref = "THX1138"; Book book = new Book("George Lacaille", "Milking It", 26745); book.setRefNumber(ref); book.printDetails();</pre>	<pre>Title: Milking It Author: George Lacaille Pages: 26745 Reference: THX1138 This book has been borrowed 0</pre>

Réponse:

```
1 class Book {
2     // The fields.
3     private String author;
4     private String title;
5     private int pages;
6     private String refNumber;
7     private int borrowed;
8     /**
9      * Set the author and title fields when this object
10     * is constructed.
11     */
12     Book(String bookAuthor, String bookTitle, int bookPages) {
13         author = bookAuthor;
14         title = bookTitle;
15         pages = bookPages;
16         refNumber = "";
```

```
17     }  
18
```

Vérifier

	Test	Expected
✓	<pre>final String ref = "THX1138"; Book book = new Book("George Lacaille", "Milking It", 26745); book.setRefNumber(ref); IntStream.range(1, 8).forEach(i -> book.borrow()); book.printDetails();</pre>	Title: Milking It Author: George Lacaille Pages: 26745 Reference: THX1138 This book has been borrowed 7
✓	<pre>final String ref = "THX1138"; Book book = new Book("George Lacaille", "Milking It", 26745); book.setRefNumber(ref); book.borrow(); book.printDetails();</pre>	Title: Milking It Author: George Lacaille Pages: 26745 Reference: THX1138 This book has been borrowed 1
✓	<pre>final String ref = "THX1138"; Book book = new Book("George Lacaille", "Milking It", 26745); book.setRefNumber(ref); book.printDetails();</pre>	Title: Milking It Author: George Lacaille Pages: 26745 Reference: THX1138 This book has been borrowed 0

Passed all tests! ✓

Correct

Note pour cet envoi : 1,00/1,00.

Question 9

Correct

Note de 1,00 sur
1,00

Add a `boolean` attribute `courseText` to record whether the book is used in a course. The value is set through an argument to the constructor, and the attribute is immutable (can't be changed). Provide an accessor method called `isCourseText`.

Copy your entire `Book` class and paste it into the Answer box. Then Check your answer.

For example:

Test	Result
<pre>final String ref = "yet another random reference"; Book book = new Book("O. Mega", "The Last Book", 475, false); book.setRefNumber(ref); System.out.println(book.isCourseText());</pre>	false

Réponse:

```
1 class Book {
2     // The fields.
3     private String author;
4     private String title;
5     private int pages;
6     private String refNumber;
7     private int borrowed;
8     final boolean courseText;
9     /**
10    * Set the author and title fields when this object
11    * is constructed.
12    */
13    Book(String bookAuthor, String bookTitle, int bookPages, boolean bookcourseText) {
14        author = bookAuthor;
15        title = bookTitle;
16        pages = bookPages;
17        refNumber = "";
18        courseText = bookcourseText;
19    }
20 }
```

Vérifier

	Test	Expected	Got	
✓	<pre>final String ref = "yet another random reference"; Book book = new Book("O. Mega", "The Last Book", 475, true); book.setRefNumber(ref); System.out.println(book.isCourseText());</pre>	true	true	✓
✓	<pre>final String ref = "yet another random reference"; Book book = new Book("O. Mega", "The Last Book", 475, false); book.setRefNumber(ref); System.out.println(book.isCourseText());</pre>	false	false	✓

Passed all tests! ✓

Correct

Note pour cet envoi : 1,00/1,00.