

Contents

Description	1
Purpose	1
Challenge	1
In short	1
In more details	1
Submission	2
Example	2
Bonuses	4

Description

Purpose

This project is designed to teach you how to manipulate exceptions and properties. It reinforces your understanding of class design and implementation while introducing the handling of exceptions.

Challenge

In short

Develop a class that helps the user track the progress in their reading of books. Your class should allow users to track the book title, total page number, and the current page they stopped their reading at (as a bookmark!).

In more details

1. Your goal is to design and implement a `Bookmarker` class containing three attributes and several methods (some of which may and should be implemented using properties).
2. You *must* develop a `Program` class with a `Main` method to test your `Bookmarker` class, and this `Program` class will be part of your grade.
3. An object in your `Bookmarker` must hold three attributes:
 - a title,
 - the total number of pages in the book,
 - the current page the user is reading.

The title *must* be implemented using automatic properties.

4. The current page *must* be less than the total number of pages: if at any point during creation or later modification the current page

becomes strictly larger than the total number of pages, your class should throw an exception.

5. The total number of pages *must be* strictly positive, but the current page can be positive (0 would be an acceptable value: it means the reader has not started yet).
6. Finally, your class should have three methods:
 - a constructor that takes three arguments (one for the title, one for the total number of pages, and one for the current page),
 - a `ToString` method that returns a `string` containing the title, how much of the book was read, and how much of the book is left (the last two in percentages),
 - a method to increment the number of pages read by its argument.

Submission

Please, follow our guideline on project submission¹. In particular, make sure you write your name and the date in a delimited comment at the beginning of your file.

Example

Remember that you have to develop a `Program.cs` `Main` method that will test your class. It will probably contain something along the lines of

```
Bookmarker book1 = new Bookmarker(title, tPages, cPages);
Console.WriteLine(book1);
```

to create the `Bookmarker` object with data given by the user, and to display the object created.

Executing your program with a properly implemented `Bookmarker` class should give something along the lines of:

```
Enter the title of the book.
T e s t 1 ↵
Enter the total number of pages.
1 0 0 ↵
Enter the page you stopped your reading at.
3 7 ↵
You have read 37,00% of "Test1".
You have 63,00% to go!
How many pages did you read since?
1 2 ↵
```

¹<https://princomp.github.io/projects/submission>

You have read 49,00% of "Test1".
You have 51,00% to go!

Press any key to continue...

If the user enters incorrect data, then the error should be thrown when the object is created (do *not* perform user-input validation in the Main method: incorrect data should be handled *by the class*, not by the Main method).

For example, if the user enters a *negative* number of pages, you should get:

Enter the title of the book.
T e s t 2 ↵
Enter the total number of pages.
- 2 4 ↵
Enter the page you stopped your reading at.
1 2 ↵
The total number of pages cannot be negative.

If the user enters a current page that is greater than the number of pages, you should get:

Enter the title of the book.
T e s t 3 ↵
Enter the total number of pages.
2 5 ↵
Enter the page you stopped your reading at.
3 0 ↵
You cannot have read more than the total number of pages!

If the user tries to go "above" the total number of pages, you should get:

Enter the title of the book.
T e s t 4 ↵
Enter the total number of pages.
2 5 ↵
Enter the page you stopped your reading at.
1 7 ↵
You have read 68,00% of "Test4".
You have 32,00% to go!
How many pages did you read?
9 ↵
You cannot have read more than the total number of pages!

Last but not least (even if this is optional), if the user enters **strings** that are not numbers, *your class* (and not the Main method) should throw an exception:

Enter the title of the book.

T e s t 5 ↵

Enter the total number of pages.

T e s t 6 ↵

Enter the page you stopped your reading at.

T e s t 7 ↵

Input string was not in a correct format.

Note that it is ok if you cannot reproduce this output *exactly*.

Bonuses

- Have your class handle **strings**, so that, for example, your Bookmarker constructor would take *three strings as input*,
- Have your Main method ask the user *how many books they want to track* and create that number of objects,
- Implement a *static* ISBN-13 checker: your method should take as input a **string** representing an ISBN-13, and return **true** if it is valid. Consult wikipedia² on how a ISBN is determined to be valid, and try your best!

²https://en.wikipedia.org/wiki/ISBN#ISBN-13_check_digit_calculation