Lab Report Anh Pham Viet / Pavel Tsvyatkov – 30.04.19

1. **Semester / IMI -** WH C 579

Professor	Name of Exercise
Dr. Prof. Debora Weber-Wulff	Exercise 2

General Information

Index

Lab Structure

- Checking the Pre-Lab
- Explaining the assignments step by step
- Choosing lab partner for this week by memory cards
- Working on the tasks for the lab report
- Saving data on WebDrive
- Logging Out

Material

- Computer
- Logbook / Pen

Assignment

CodePad

1. Open BlueJ and find the CodePad. Use it to test your predictions for P4 and record where you were correct and where you made mistakes.

Making a Book

2. Download the BookExercise project from Moodle. 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.

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

4. 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.

5. Add a method, printDetails, to the Book class. This should print details of the author, title, and pages to the terminal window. It is your choice how the details are formatted. You might want to include some explanatory text.

6. 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. Define a mutator for it with the following signature:
   public void setRefNumber (String ref)
   The body of this method should assign the value of the parameter to the refNumber field. Add the corresponding accessor getRefNumber.

7. Modify your printDetails method to include printing the reference number. However, the method should print the reference number only if it has been set. Hint: use a conditional! Note that Strings have a length method.

8. 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 and leave the field unchanged.

9. (For the bored) Create a new project, heater-exercise, within BlueJ. Edit the details in the project description — the text note you see in the diagram. Create a class Heater, that contains a single integer field, temperature. Define a constructor that takes no parameters. The temperature field should be set to the value 15 in the constructor. Define the mutators warmer and cooler, whose effect is to increase or decrease the value of temperature by 2 degrees respectively. Define an accessor method to return the value of temperature.

10. (For the bored) Modify your Heater class to define three new integer fields: min, max and increment. The values of min and max should be set by parameters passed to the constructor. The value of increment should be set to 2 in the constructor. Modify the definitions of warmer and cooler so that the use the value of increment rather than an explicit value. Check that everything works as before. Now modify the warmer method so that it will not allow the teperature to be set to a value greater than max. Similarly, modify cooler so that it will not allow temperature to be set to a value less than min. Check that the class works properly. Now add a method, setIncrement, that takes a single integer parameter and uses it to set the value of increment. Add a check to prevent a negative number being used here!

Lab Report

Expectation & Goals

Anh Pham Viet:

After doing the Pre-Lab I expected to work more with strings and learn more about the different operators we had to write down in P3.

Pavel Tsvyatkov:

In this lab exercise i expect to learn more about the Java language and use different methods from the Java API.

Assignment

We had the usal “driver” & the “navigator” role-switching during the laboratory session.

1. **Task**

We were both right up until 9 + 3 + “cat”.

There Anh had the value: 93cat. The right value according to the notepad was: 12cat.

He thought that a string in a concatenation would make numbers to be a string as well hence there would be no addition operated. That only seems to apply though when the string is in the beginning of the expression as you can see in the next expression:

“cat” + 3 + 9 “fish”.

This time we were both right. The value according to the notepad was: cat39fish.

We both had all the types right. Our first substring was correct, but we were not sure what exactly would happen with the second one. The EndIndex was set too high on this one, thus the following exception appears:

```
"catfish". substring(3,8)
    Exception: java.lang.StringIndexOutOfBoundsException (begin 3, end 8, length 7)
```

```
java.lang.StringIndexOutOfBoundsException: begin 3, end 8, length 7
        at java.base/java.lang.String.checkBoundsBeginEnd(String.java:3319)
        at java.base/java.lang.String.substring(String.java:1874)
```

We were not sure whether an Exception is an error or not. So we put the String into our code and used the System.out.print method and it actually compiled, but it didn’t print out the string.

2. Task

An accessor always starts with the keyword **public** and since author is declared in the fields as String, we wrote the following method:

```
/*
 * Method to print the author
 */
public String getAuthor()
{
return author;
}
```

Screenshot 1: getAuthor Method

```
/*
 * Method to print the title
 */
public String getTitle()
{
return title;
}
```

Screenshot 2: getTitle Method

We accidently wrote **String** with a lower-case character and it did not compile, hence we learned that the type String has to be written with a capital letter in the beginning unlike other primitive types like int. As this is an accessor method, we have to **return** the value as well.

3. Task

Since we want to print the author and title fields, we will need the keyword **void** in our method. We used System.out.println and put the field names into the brackets.

```
/*
 * Method to print the author to the terminal
 */
public void printAuthor()
{
System.out.println(author);
}
```

Screenshot 1: printAuthor Method

```
/*
 * Method to print the author to the terminal
 */
public void printTitle()
{
System.out.println(title);
}
```

Screenshot 2: printTitle Method

4. Task

As instructed we created a new field called pages and set its type to int.We both came to an agreement that it would only make sense if the intial value for pages would be set to 1.

```java
// The fields.
private String author;
private String title;
private int pages;
private String refNumber;
/**
 * Set the author and title fields when this object
 * is constructed.
 */
public Book(String bookAuthor, String bookTitle)
{
    author = bookAuthor;
    title = bookTitle;
    pages = 1;
    refNumber = "";
}
```

Respectively the getPages method which returns the page number.

```java
/*
 * Method to print the number of pages
 */

public int getPages()
{
return pages;
}
```

5. Task

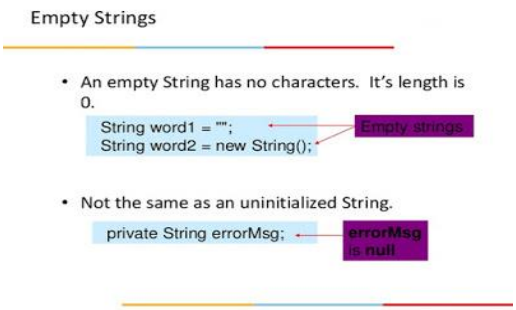
Since this is another print method, we used the keyword **void** again and System.out.println to print the method. We did the exact same thing as in Task 3 and put the fields in the System.out.println brackets.

However it didn't look very pleasing, so we added a concatenation string for each field to make it more clear what is printed exactly.

```java
/*
 * Method to print details about Book
 */
public void printDetails()
{
System.out.println("Author: " + author);
System.out.println("Title: " + title);
System.out.println("Pages: " + pages);
if ( refNumber.length() > 0) {
System.out.println("Reference Number: " + refNumber); }
}
```

6. Task

For this task we needed to check in internet about how to go and initialize our refNumber string to the zero-length string and we found a picture about this task, which helped us[i]:



The first thing to do was to add a new field to the Book class.

```java
class Book
{
    // The fields.
    private String author;
    private String title;
    private int pages;
    private String refNumber;
```

Then in the constructor we decided to use the second example from the picture and wrote String refNumber = new String(); so it will be initialized as a zero-length String. Here we were wrong and our program was not working correctly.

```java
public Book(String bookAuthor, String bookTitle)
{
    author = bookAuthor;
    title = bookTitle;
    pages = 1;
    String refNumber = new String();
}
```

After thinking about it for a while, we couldn't get our program to work, so we called Prof. Weber-Wulff for help. She told us that we overloaded the refNumber in the constructor and then we changed it to just refNumber = “”; . That was the correct way here to initialize refNumber to the zero-length String.

```java
public Book(String bookAuthor, String bookTitle)
{
    author = bookAuthor;
    title = bookTitle;
    pages = 1;
    refNumber = "";
}
```

As the task was saying, we defined a mutator setRefNumber() and assigned refNumber to ref in its body. After that we added an accessor method called getRefNumber()" , which should return the value stored in refNumber.

```java
/*
 * Method to store the reference number
 */
public void setRefNumber(String ref)
{
    refNumber = ref;
}
```

Screenshot 1: setRefNumber Method

```java
/*
 * Accessor to get the reference number
 */
public String getRefNumber()
{
return refNumber;
}
```

Screenshot 2: getRefNumber Method

Our first mistake was that we assumed, because refNumber has “number” in its name, it should be of type int and it didn't work, since refNumber is declared as a String. Then we both discussed it and changed it to String. Then we were able to compile without any problems.

7. Task

We both discussed what we should do exactly in this exercise. At first, we weren't sure how to write the conditional statement correctly, so that it checks the length of refNumber. We were trying different things, but nothing was working. Then we read the exercise again and we both came up with the idea to use the length() method on refNumber and set it so that it checks if the length is bigger than 0.

```java
/*
 * Method to print the details about the book
 */
public void printDetails()
{
    System.out.println("Author: " + author);
    System.out.println("Title: " + title);
    System.out.println("Page: " + pages);

    if (refNumber.length() > 0){
        System.out.println("Refnumber: " + refNumber);
    }
    else {
        }

}
```

We thougt there should always be an else statement following after the if statement, but Prof. Dr. Weber-Wulff told us that it is not mandatory for the code to work. In this situation we did not need an else statement.

```
/*
 * Method to print details about Book
 */
public void printDetails()
{
System.out.println("Author: " + author);
System.out.println("Title: " + title);
System.out.println("Pages: " + pages);
if ( refNumber.length() > 0) {
System.out.println("Reference Number: " + refNumber); }
}
```

8. Task

For this exercise we had to exchange ideas at home and discuss it, because we couldn't manage to get to it in the laboratory. We both agreed that we should use an if statement in the mutator again, because it should check if the reference number has been set or not. This time we had to include an else statement, which should print an error message when the reference number was not set.

```
/*
 * Mutator that sets the reference number
 */
public void setRefNumber(String ref)
{
    if (ref.length() > 2) {
    refNumber = ref; }

    else { System.out.println("Error! Wrong RefNumber!");}
}
```

We tested the program by creating a new book, setting the reference number to "A1" and it was printing an error message to the terminal, which was correct. Then we tried to set it to "A12" and it didn't give us an error. After that we invoked the printDetails() method and we saw that the reference number was included, because the condition has been met and it was longer than three characters. We were then assured that our program was working correctly.

BlueJ: Terminal Window - Book-Exercise-Done

Options

```
Author: StephenKing
Title: ES
Pages: 1
Reference Number: A12
```

After the lab

We agreed that one group member will send the text notes from the laboratory to the other group member and one of us will begin writing the PDF file. We did not have time in the lab for Task 8, so we met later in the library in the week and solved the task together.

What we learned from the assignment / reflection

Pavel:

From this lab exercise i learned that when we concatenate in Java, if the expression starts with a string, then Java outputs the whole expression as a string, like how we saw in P4 from the pre-lab exercises. I also learned that we get an OutOfBound error when the end index of the substring method is bigger than the length of the string we are trying to invoke the method on. Another thing that I learned and got more practice on was how to correctly add a new field to the class, initialize it in the constructor and then use different methods with it, without any problems so far. It was also

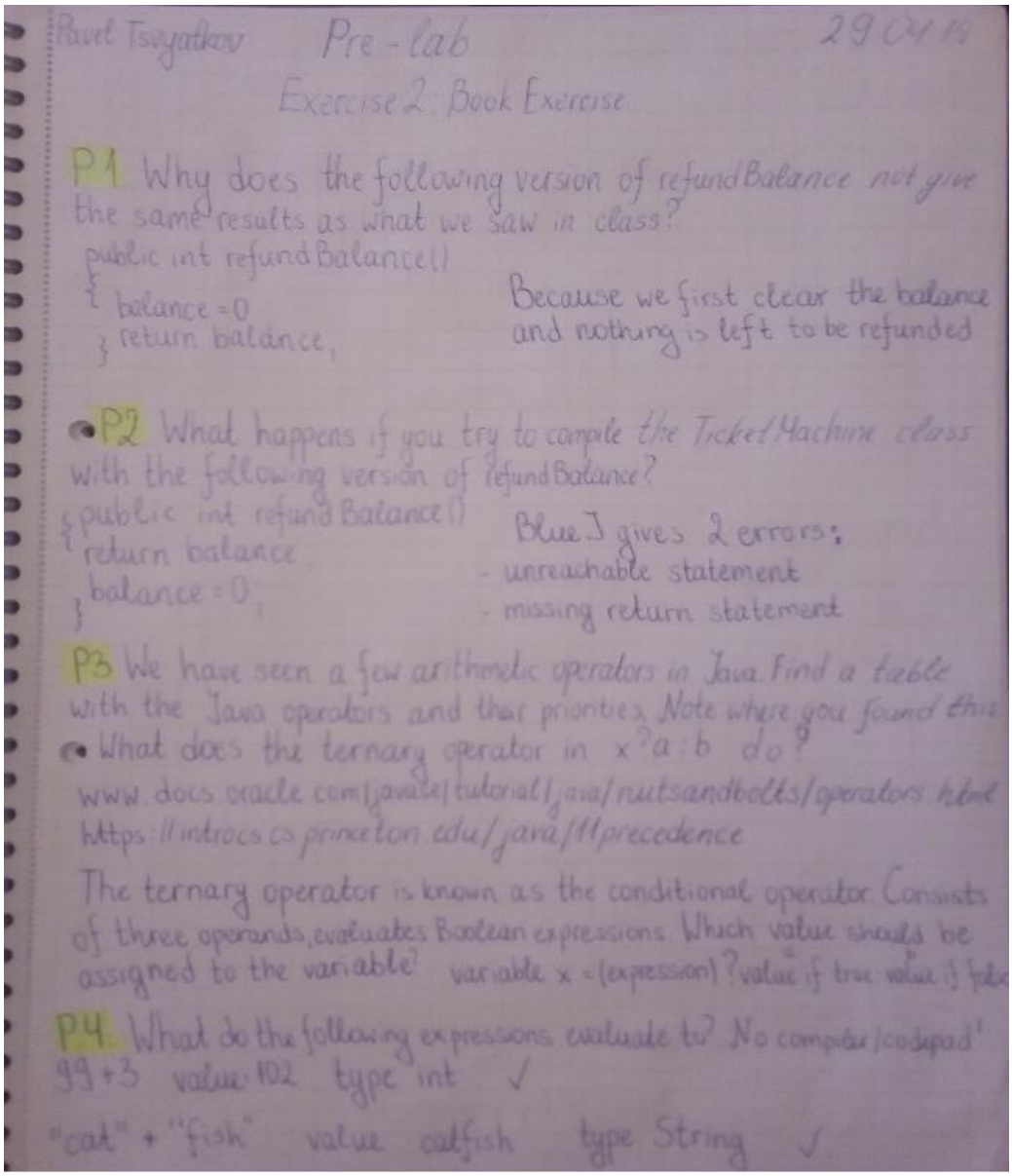
interesting to learn how to initialize a String to the zero-length string. In the end of the exercise I learned how to correctly write an if statement that checks if a string's length is bigger than a certain value and in case it wasn't, then how to properly add the needed else statement to it. We also both learned that it is not always neccessary to have an „else" if we are using an „if" statement.

Anh:

This time we downloaded a code which was almost empty, so I got to experience how to write a code from the scratch. I noticed that I got more fluent with the setter methods, printing out concatenation strings with variables is also non-problematic. It is easier to see the patterns like when to set parameters in the head of the method and when not.
Moreover we learned more about the type string, that it is written with an initial capital letter and that a string in the beginning of a concatenation makes numbers to be a string too. We also had an if-statement again and I am starting to see that learning how to set up loops properly is vital in programming.

Appendix

- Pre-lab Pavel



Pavel Tsvyatkov Pre-lab 29.04.19

Exercise 2: Book Exercise

P1 Why does the following version of refundBalance not give the same results as what we saw in class?

```
public int refundBalance()
{ balance = 0
  return balance;
}
```

Because we first clear the balance and nothing is left to be refunded

P2 What happens if you try to compile the TicketMachine class with the following version of refundBalance?

```
public int refundBalance()
{ return balance;
  balance = 0;
}
```

BlueJ gives 2 errors:
- unreachable statement
- missing return statement

P3 We have seen a few arithmetic operators in Java. Find a table with the Java operators and their priorities. Note where you found this

- What does the ternary operator in x ? a : b do?

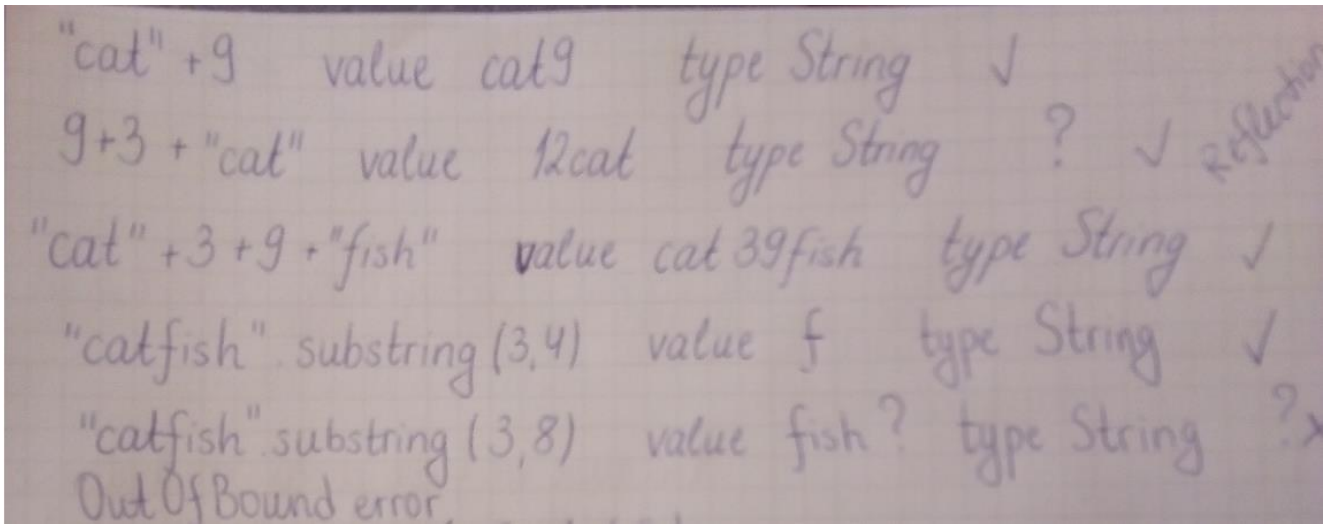
www.docs.oracle.com/javase/tutorial/java/nutsandbolts/operators.html
https://introcs.cs.princeton.edu/java/11precedence

The ternary operator is known as the conditional operator. Consists of three operands, evaluates Boolean expressions. Which value should be assigned to the variable? variable x = (expression) ? value if true : value if false

P4 What do the following expressions evaluate to? No compiler/codepad!

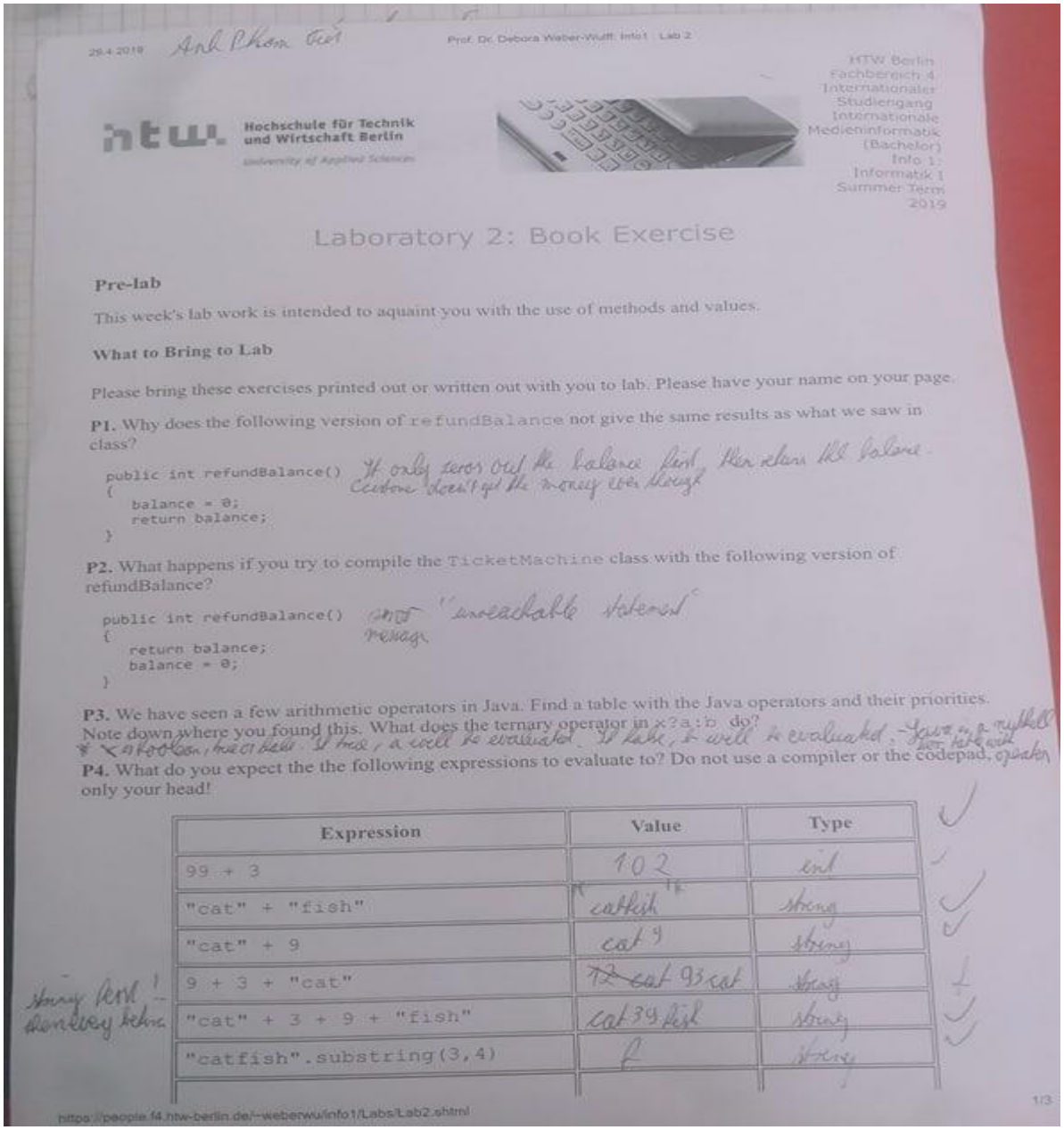
99+3 value: 102 type int ✓

"cat" + "fish" value catfish type String ✓



"cat" + 9 value cat9 type String ✓
9+3 + "cat" value 12cat type String ? ✓ reflection
"cat" + 3 + 9 + "fish" value cat39fish type String ✓
"catfish".substring(3,4) value f type String ✓
"catfish".substring(3,8) value fish? type String ?
OutOfBound error.

- Pre-lab Anh



29.4.2019 Anh Phan Tuấn

Prof. Dr. Debora Weber-Wulff, Info1 | Lab 2

HTW Berlin
Fachbereich 4
Internationaler Studiengang Internationale Medieninformatik (Bachelor)
Info 1: Informatik 1
Summer Term 2019

Hochschule für Technik und Wirtschaft Berlin
University of Applied Sciences

Laboratory 2: Book Exercise

Pre-lab

This week's lab work is intended to aquaint you with the use of methods and values.

What to Bring to Lab

Please bring these exercises printed out or written out with you to lab. Please have your name on your page.

P1. Why does the following version of `refundBalance` not give the same results as what we saw in class?

```
public int refundBalance()
{
   balance = 0;
   return balance;
}
```

It only zeros out the balance first, then shows the balance. Customer doesn't get the money back though

P2. What happens if you try to compile the `TicketMachine` class with the following version of refundBalance?

```
public int refundBalance()
{
   return balance;
   balance = 0;
}
```

error message "unreachable statement"

P3. We have seen a few arithmetic operators in Java. Find a table with the Java operators and their priorities. Note down where you found this. What does the ternary operator in `x?a:b` do?

x → a boolean, true → a, false → b. If true, a will be evaluated. If false, b will be evaluated. - Java is a mythical man-month

P4. What do you expect the the following expressions to evaluate to? Do not use a compiler or the codepad, only your head!

Expression	Value	Type
`99 + 3`	102	int
`"cat" + "fish"`	"catfish"	string
`"cat" + 9`	cat9	string
`9 + 3 + "cat"`	~~12 cat~~ 93cat	string
`"cat" + 3 + 9 + "fish"`	cat39fish	string
`"catfish".substring(3,4)`	f	string

String fish! - donkey behind

https://people.f4.htw-berlin.de/~weberwu/info1/Labs/Lab2.shtml 1/3

P3 Order listed according to precedence (source: docs.oracle

postfix	expr ++ expr --
unary	++expr --expr +expr -expr ~
multiplicative	* / %
additive	+ -
shift	<< >> >>>
relational	< > <= >= instanceof
equality	== !=
bitwise AND	&
bitwise exclusive OR	^
bitwise inclusive OR	|
logical AND	&&
logical OR	||
ternary	? :
assignment	= += -= *= /= %= &= ^= |= <<= >>= >>>=

x ? a : b

x is type boolean and ~~will be evaluated~~ is either true or false,

if true, a will be evaluated

if false, b will evaluated

Our code

```
/**
 * 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.)
 * @version (Insert today's date here.)
 */
class Book
{
   // The fields.
   private String author;
   private String title;
   private int pages;
   private String refNumber;
   /**
    * Set the author and title fields when this object
    * is constructed.
    */
   public Book(String bookAuthor, String bookTitle)
   {
      author = bookAuthor;
      title = bookTitle;
      pages = 1;
      refNumber = "";
   }
   /*
    * Method to print the author
    */
   public String getAuthor()
   {
   return author;
   }

   /*
    * Method to print the title
    */
   public String getTitle()
   {
```

```
return title;
}

/*
 * Method to print the author to the terminal
 */
public void printAuthor()
{
System.out.println(author);
}

/*
 * Method to print the author to the terminal
 */
public void printTitle()
{
System.out.println(title);
}

/*
 * Method to print the number of pages
 */

public int getPages()
{
return pages;
}

/*
 * Method to print details about the Book
 */
public void printDetails()
{
System.out.println("Author: " + author);
System.out.println("Title: " + title);
System.out.println("Pages: " + pages);
if ( refNumber.length() > 0) {
System.out.println("Reference Number: " + refNumber); }
}

/*
 * Mutator that sets the reference number
```

```
    */
    public void setRefNumber(String ref)
    {
        if (ref.length() > 2) {
        refNumber = ref; }

        else { System.out.println("Error! Wrong RefNumber!");}
    }

    /*
     * Accessor to get the reference number
     */
    public String getRefNumber()
    {
    return refNumber;
    }
}
```

Documenation

Class Book

java.lang.Object
  Book

class **Book**
extends java.lang.Object

A class that maintains information on a book. This might form part of a larger application such as a library system, for instance.

Version:
(Insert today's date here.)

Author:
(Insert your name here.)

Constructor Summary

Constructors

Constructor	Description
Book(java.lang.String bookAuthor, java.lang.String bookTitle)	Set the author and title fields when this object is constructed.

Method Summary

All Methods | **Instance Methods** | **Concrete Methods**

Modifier and Type	Method	Description
java.lang.String	getAuthor()	
int	getPages()	
java.lang.String	getRefNumber()	
java.lang.String	getTitle()	
void	printAuthor()	
void	printDetails()	
void	printTitle()	
void	setRefNumber(java.lang.String ref)	

Methods inherited from class java.lang.Object

clone, equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

Book

```
public Book(java.lang.String bookAuthor,
            java.lang.String bookTitle)
```

Set the author and title fields when this object is constructed.

Method Detail

getAuthor

```
public java.lang.String getAuthor()
```

getTitle

```
public java.lang.String getTitle()
```

printAuthor

```
public void printAuthor()
```

printTitle

```
public void printTitle()
```

getPages

```
public int getPages()
```

printDetails

```
public void printDetails()
```

setRefNumber

```
public void setRefNumber(java.lang.String ref)
```

getRefNumber

```
public java.lang.String getRefNumber()
```

[i] Javarevisited – 3 Ways to check if string is null or empty in Java (2016/01/05) https://javarevisited.blogspot.com/2016/01/how-to-check-if-string-is-not-null-and-empty-in-java-example.html (retrieved on 30th may 2019)