10.11.2015

Nils Gawlik

Lab Report for the 10th September 2015

Pre lab:

P1:

`balance` is set to zero before it is returned. That means this function will always return zero.

P2:

When the return statement is executed, it exits the function. This means, that `balance = 0;` is never executed. The compiler should see this and give an error.

P3:

Operator Precedence

Operators	Precedence
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	= += -= *= /= %= &= ^= |= <<= >>= >>>=

→ https://docs.oracle.com/javase/tutorial/java/nutsandbolts/operators.html

`x ? a : b` is `a` if `x` is true and `b` if `x` is false.

10.11.2015

Nils Gawlik

P4:

Expression	**Value**	**Type**
`99 + 3`	`102`	`int`
`"cat" + "fish"`	`“catfish”`	`String`
`"cat" + 9`	`“cat9”`	`String`
`9 + 3 + "cat"`	`“12cat”`	`String`
`"cat" + 3 + 9 + "fish"`	`“cat39fish”`	`String`
`"catfish".substring(3,4)`	`“f”`	`String`
`"catfish".substring(3,8)`	`“fish”`	`String`

CodePad:

The CodePad was already enabled on my computer. I typed in all the expressions from exercise P4 and only the last one did not work as I predicted, the other ones were exactly the same. The last one gave me a `java.lang.StringIndexOutOfBoundsException`, which exited execution. I assume this is because the second number is higher than the string length, so it is out of the bounds of the string. I expected the method `substring` to just ignore that and cut the number down to the length of the string.

Making a book

note: I willl mostly refer to “fields” as attributes, because they are to my knowledge the same thing and attribute is the term I'm used to.

1.

I created a new public method called `getAuthor` with the returned type being String and no arguments. Inside the method I returned the attribute author.

The `getTitle` works exactly the same way with the title attribute.

2.

The methods `printAuthor` and `printTitle` are also pretty much the same. They have no returned value and no arguments and use `System.out.println` with author/title as argument to print those variable's values to the console.

3.

I first added the attribute `pages` by declaring a new private variable of type int called `pages` at the top of the class. I added another argument to the constructor called `bookPages`. I initialize the variable `pages` with the value of `bookPages` in the constructor. The `getPages` accessor for the `pages` attribute works the same as the accessors from task 1, except the return type is now int.

10.11.2015

Nils Gawlik

4.

The method `printDetails` works the same way as `printAuthor` and `printTitle`, except it prints a string which is formatted as follows: <title> by <author> (<x> pages). This is achieved by adding the variables and the text between the variables (as strings) together in order.

5.

I created the attribute `refNumber` the same way I created the `pages` attribute (except the type is String). In the constructor the variable is set to `""`. In the setter method I set the variable `refNumber` to `ref`. The getter (accessor) works the same way as the other accessors.

6.

To insert the reference number into the `getDetails` output, I used a ternary operator added inside the chain of strings and attributes that make up the formatted output text. The operator checks if `refNumber` is not `""` and returns `"ref: " + refNumber` if true and `""` otherwise. The ternary operator is enclosed in brackets, because it is below the `+` operator in precedence and is added in before the `pages` variable. This way the ref number appears in the text if it contains something and otherwise an empty String gets added in, which does not affect the output.

7.

To achieve this I had to add an if statement to the `setRefNumber` mutator. It checks if the length of `ref` (the argument) is at least 3 using `ref.length()` and `>=` and only sets `refNumber` to the new value if that is true. Otherwise it prints out an error.

8.

I created the `countedLeaves` attribute and the getter method the same way as in the tasks above. I also decided to use the mutator `private void increaseLeaves()`. While this was a little bit unnecessary, it seemed like the right stylistic choice, since the variable will only be increased one at a time and this mutator reflects that. The `act` method contains the necessary code to complete the task: stopping if there is a tree, moving otherwise and calling `increaseLeaves()` if on a leaf.