RELEASE 01:
Chapter 02 & 03.

Exercises explanation document

by María Díaz-Rozas Pantoja

1º GSDAW-Y

CONTENIDO

CHAPTER 02.

In this chapter 02 to variables and operators, I did the following exercise:

Exercise 2:

In this Exercise I defined the public class method named *Date.java*, I declared the simple variables with string and integer for day, date, month, and year. And I used the statements like:

```
System.out.println ("Date of the day"); System.out.print ("American format: ");

System.out.println (“day+", "+month+" "+date+"th" +", "+year);
```

And I repeated a similar statement with the Europe format. This exercise didn't give me too many problems.

Exercise 3

- **Create a new program called *Time.java.***
- **Create variables named hour, minute, and second. Assign values that are roughly the current time. Use a 24-hour clock so that at 2:00 PM the value of hour is 14.**
- **Make the program calculate and display the number of seconds since midnight.**
- **Calculate and display the number of seconds remaining in the day.**
- **Calculate and display the percentage of the day that has passed. You might run into problems when computing percentages with integers, so consider using floating-point.**
- **Change the values of hour, minute, and second to reflect the current time. Then write code to compute the elapsed time since you started working on this exercise.**

To perform the exercise, first I created the *Time.java* file. Once the file was created, I defined the *public class* method named *Time*, and the variables:

```
int hour = 20; int minute = 42; int second = 20
```

I defined:

```
System.out.print("The current time is ");

System.out.print(hour); System.out.print(":");

System.out.print(minute); System.out.print(":");

System.out.print(second); System.out.println(".");
```

And now the program shows me the “current” time:

```
The current time is 20:42:20.
```

I made the program calculate the following printing variable and this expression:

```
System.out.print("Number of seconds since midnight: ");

System.out.println(((hour * 60 + minute) * 60) +second);
```

The output to this is:

Number of second since midnight: 74520 *(in seconds)*

In the following point, I told the program to print me the number of seconds remaining in the day with this expression:

```
System.out.print("Number of second remaining in the day: ");

System.out.println((24*60*60) - ((hour * 60 + minute) * 60)- second);
```

And now java print:

Number of second remaining in the day: 11860 *(in seconds)*

In the next step, I had any problems with the floating-point because the previous variables it gave me a result of 0, and if I used the variables with "double" the results of the previous points changed. So, in this case, I created the other variables with other names.

```
double minutos = 42; double horas = 20; double segundos = 20;
```

And I defined the following expression:

```
System.out.printf ("Fraction of the day that has passed = %.3f",

(((horas * 60 + minutos) * 60) + segundos) / (24*60*60));
```

*** I actually changed this point after reading chapter 03, because the result that java showed me before was extremely long, so, apply what we learned in chapter 03.*

And now, the program print me:

Fraction of the day that has passed = 0,863

To finish, I required the program to compare the time to starting the exercise with the time to finishing it. And now if you run the java Time.java, the program print:

```
The current time is 20:42:20.

Number of seconds since midnight: 74540

Number of seconds remaining in the day: 11860

Fraction of the day that has passed = 0,86

Number of seconds since I started working on this exercise: 4236
```

CHAPTER 03:

In this chapter 03 to Input and Output, I did the following exercise:

Exercise 1.

- **When you use *printf,* the Java compiler does not check your format string. See what happens if you try to display a value with type int using %f.**
- **And what happens if you display a double using %d?**
- **What if you use two format specifiers, but then provide only one value?**

The *printf* into Java compiler is a format to string, so if you try to display a value (int) with %f you have fatal error, because the special character %f is for *String*. If you want to use this *printf* with *double* or *float (int)* you should put %d.
I try with the file named *Convert.java* and the result was:

Mistake:

```
Exactly how many cm? 123
123,00 cm = Exception in thread "main" java.util.IllegalFormatConversionException: f != java.lang.Integer
        at java.base/java.util.Formatter$FormatSpecifier.failConversion(Formatter.java:4442)
        at java.base/java.util.Formatter$FormatSpecifier.printFloat(Formatter.java:2976)
        at java.base/java.util.Formatter$FormatSpecifier.print(Formatter.java:2924)
        at java.base/java.util.Formatter.format(Formatter.java:2689)
        at java.base/java.io.PrintStream.format(PrintStream.java:1209)
        at java.base/java.io.PrintStream.printf(PrintStream.java:1105)
```

Right:

```
        Exactly how many cm? 123
        123,00 cm = 4 ft, 0 in
```

Exercise 2.

- **Write a program that converts a *temperature* from Celsius to Fahrenheit.**

In this exercise I used System.in method. This is a method for reading input from keyboard.

Before use it, I had import it: `import java.util.Scanner;`

In this exercise I used the next formula to do conversion:

Fahrenheit (F) = (Celsius × 9 / 5) + 32

In first, I declare variable for Celsius (is similar with other thing do you want to convert).

And I declared the name *in* and created *the scanner.*

```
Scanner in = new Scanner (System.in);
```

The next step is to prompt the user for the input with:

```
System.out.print("Enter a temperature in Celsius: ");
celsius = in.nextDouble();
```

And again, I declared the others variables like double. I put the variables *gradosF* and *fahren* because the syntaxis gave me errors. In this way, duplicating the f*ahren* variable is avoided.

```
double gradosF = ((celsius*9/5) +32);
double fahren = gradosF;
```

If you run it and type the value to convert, the program print:

```
Enter a temperature in Celsius: 36
36,0 C = 96,8 F
```

Exercise 3.

- **Write a program that converts a total number of seconds to hours, minutes, and seconds. Use the remainder operator.**

In this exercise I had to convert a total number of second (*prompt the user)* to hours, minutes, and seconds. The problem here was that Java gave me a formatting error. I was finally able to see what the error was, similar to the test in exercise 3.1.

In this case, I put %d with printf.

```
System.out.printf ("%d seconds = %d hour, %d minute, %d seconds \n", seconds, hour, minute, remainder );
```

In conclusion, I used the method *java.util.scanner* with remainder. First of all, I declared variables and constant. The next point was define the scanner: *in.nextdouble. A*nd to finish I declared the operations and *printf* to the results.

And now if you run it and type the value, the program show:

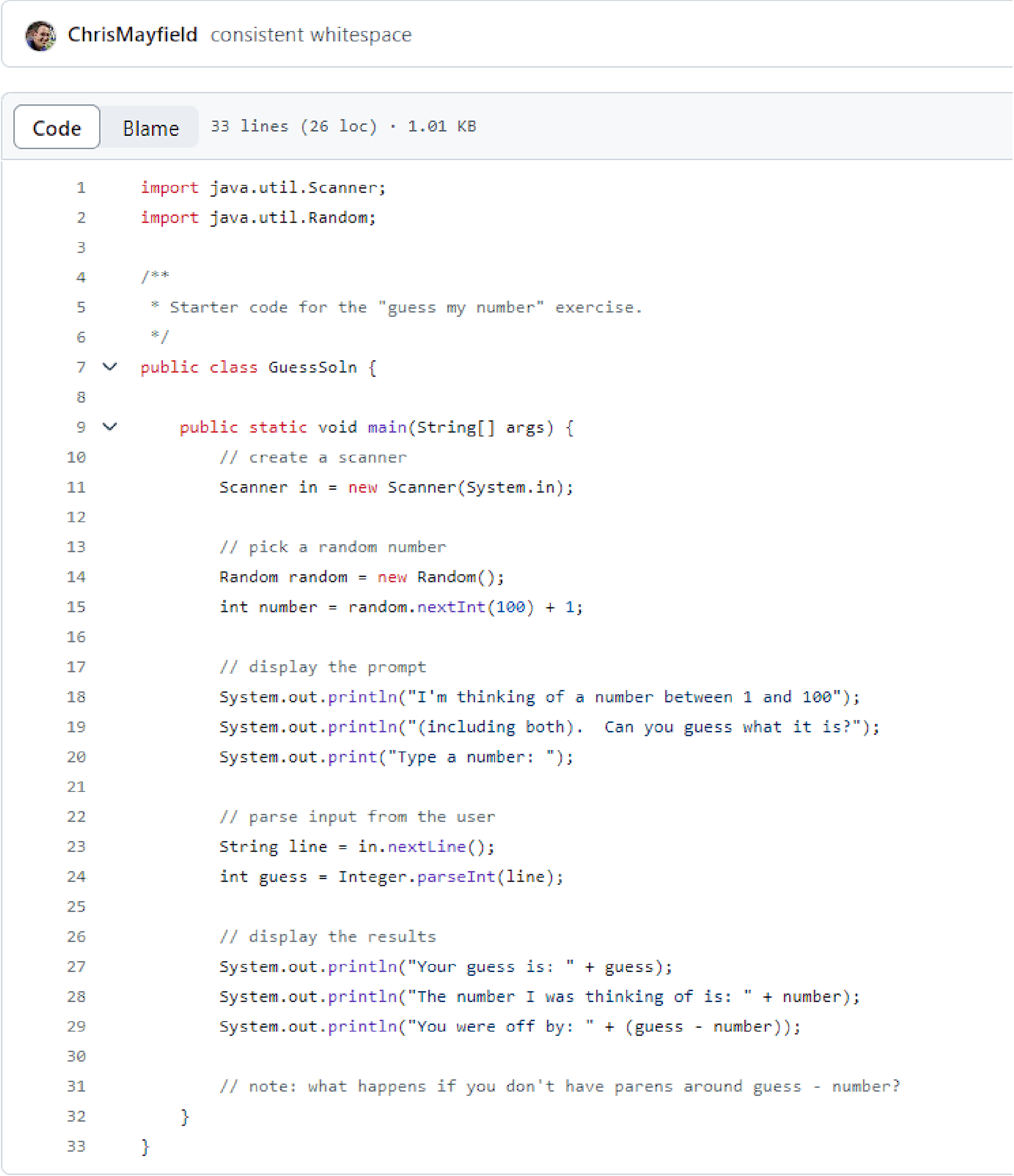
```
How many seconds? 5000
5000 seconds = 1 hour, 83 minute, 23 seconds
```

Exercise 4.

- **The goal of this exercise is to program a *Guess My Number* game.**

In this case, I used a *Random* class into *java.util.Random.* In fist at all, I imported this class. This util is a number generator.

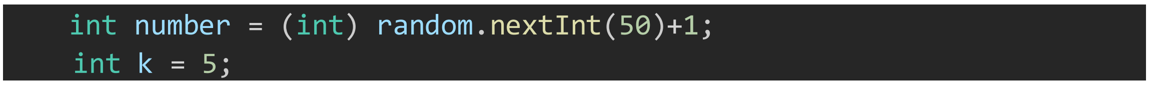
I visited the repository named ThinkJava, and I saw the GuessSoln.java file:


ChrisMayfield consistent whitespace
Code
Blame
33 lines (26 loc) · 1.01 KB

```java
import java.util.Scanner;
import java.util.Random;

/**
 * Starter code for the "guess my number" exercise.
 */
public class GuessSoln {

    public static void main(String[] args) {
        // create a scanner
        Scanner in = new Scanner(System.in);

        // pick a random number
        Random random = new Random();
        int number = random.nextInt(100) + 1;

        // display the prompt
        System.out.println("I'm thinking of a number between 1 and 100");
        System.out.println("(including both).  Can you guess what it is?");
        System.out.print("Type a number: ");

        // parse input from the user
        String line = in.nextLine();
        int guess = Integer.parseInt(line);

        // display the results
        System.out.println("Your guess is: " + guess);
        System.out.println("The number I was thinking of is: " + number);
        System.out.println("You were off by: " + (guess - number));

        // note: what happens if you don't have parens around guess - number?
    }
}
```

Can you visit it for in the next link: https://github.com/AllenDowney/ThinkJava/blob/master/code/ch03/GuessSoln.java

Then I defined variables:



```
    int number = (int) random.nextInt(50)+1;
     int k = 5;
```

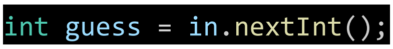
And modify the *GuessStarter* program to my own program named *Guessmynumber.java.*

In this exercise it was difficult for me to find the correct form to create this program, but finally after hours looking on the internet how to do it, I discovered the conditionals: while, if, and else (that we had already mentioned in class previously).

So, I decided to use this conditional to do the exercise.

While the k variable is greater than 1:

*Declare the variable guess:



```
int guess = in.nextInt();
```

- If the guess is equal to Random number, the user wins the game. And finish it.
- If the guess is greater than 51:
    - Message to user: “Type a number between 1 and 50”.
    - And “Please, Try again”
    - “You have (k) tries left.” (k in this case is equal value that before this message, the program don´t minus the tries).
- If the guess is equal 0:
    - The same like previously message.
- Else other value:
    - Message to user: “You have (k) tries left.” (k in this case is the result of subtracting the tries the user has already spent).
- Close the conditional “while”.

After this point, the exercise became confusing because I had a lot of problems with it. I tried different methods that only gave me errors like the program never finished once you had run it or it gave me false values. In the end I did it.

The solution was to add the new conditional “While”.

While k variable is equal to 1:

- I declared the variable *int guess=in.nextInt* again, and changed the previous one equal to this one, which was outside the while I put it inside. *
- The user has only 1 more try.
- If the guess is equal to Random number, the user wins the game. And finish it.
- If the guess If the guess is equal 0 or if the guess is greater than 51:
    - Same as in the other "while".
- Else others value, the program displays the new message:
    - “Ohh!! You ran out of tries!”

- Close the conditional “while”.

And then the program displays the difference between the user’s guess and the number that was generated with these messages:

```
System.out.println("Your number is: " + guess);
System.out.println("I was thinking of: " + number);
System.out.println("You were off by: " + (guess - number));
System.out.println("Don't worry, maybe next time you will guess!");
```

Now if you run it and type the value to convert, the program print and you don´t guess:

```
I'm thinking of a number between 1 and 50 (including both).
Can you guess what it is?
You have 5 tries to guess.
      Insert a number: 52
Type a number between 1 and 50 (including both)
Please, try again!
You have 5 tries left.
       Insert a number: 0
Type a number between 1 and 50 (including both)
Please, try again!
You have 5 tries left.
       Insert a number: 26
You have 4 tries left.
       Insert a number: 28
You have 3 tries left.
  Insert a number: 23
You have 2 tries left.
  Insert a number: 25
You have 1 tries left.
  Insert a number: 28
Ohh!! You ran out of tries!
Your number is: 28
I was thinking of: 27
You were off by: 1
Don't worry, maybe next time you will guess!
```

If you guess and win:

```
….
You have 2 tries left.
    Insert a number: 27
Congratulation!!
Your guess is: 27
The number I was thinking of is: 27
```
