Exercises Module 10 - Enum

Task 1 - Working days at Academy

Write a private static method whatsTheSituation that takes as an input argument an enum Day, consisting of working days (MONDAY, TUESDAY, ..., FRIDAY). Depending on which day, different messages should be printed out.

- MONDAY - "Uuugh!"
- TUESDAY - "TT Aaaah Yeeeaaahh!"
- WEDNESDAY - "We like Wednesday fika!"
- THURSDAY - "Almost there!"
- FRIDAY - "WHOOP WHOOP!"

Use a switch-case to solve this.

To create an Enum inside IntelliJ, choose to create a new java class as usual, in the pop-up window, choose the Kind Enum instead of Class, then add a name to the enum and then klick OK.

Or you could just change the keyword class to enum in a new class to make it an enum instead of a class.

Task 2 - Extend the Enum

Extend the enum from the previous task so that it contains *MONDAY*, *TUESDAY*, ..., *FRIDAY*, *SATURDAY*, *SUNDAY*, *CATURDAY*, *SUITDAY*, *LITTLE_SATURDAY*.

Try running your program, it should still work!

Also modify your previous method with the following switch case case logic (new statements are underlined):

- MONDAY - "Uuugh!"
- TUESDAY - "TT Aaaah Yeeeaaahh!"
- WEDNESDAY - "We like Wednesday fika!"
- *LITTLE_SATURDAY - "See you at Ugglan!"*
- THURSDAY - "Almost there!"
- FRIDAY - "WHOOP WHOOP!"
- *CATURDAY - ":3"*
- *SATURDAY - "See you in town!"*
- *SUNDAY - "Edward Blom and online pizza"*
- *SUITDAY - "Haters gonna hate"*

You might have noticed that when we added the new days in the Enum, the switch case statement did not need to know how Enum was ordered. This would not work if we had represented Day as an int and manually assigned different integers to represent the different days.

Task 3 - Traffic light

We will simulate a traffic light in the form of a finite state machine.

What is a state machine?

- **https://en.wikipedia.org/wiki/Finite-state_machine (https://en.wikipedia.org/wiki/Finite-state_machine)**

We will represent a traffic light using a enum called *TrafficLight*. *TrafficLight* should contain enough enum values to represent all states that a classically functioning traffic light may have. The list below shows how the different lights change over time:

RED

RED and YELLOW

GREEN

YELLOW

Back to RED again

How many enum values does TrafficLight need? What should the values be named?

Now create a program that begins with red light (print the current color of the traffic light on the screen), and each time the user presses ENTER, the traffic light will proceed to the next state, as shown in the image. Use a switch-case inside an infinite loop to get the traffic light switching state.

If you want to, you can also try the following to make the traffic light switch state automatically without waiting for the user to press ENTER:

Add this import statement:

```
import java.util.concurrent.TimeUnit;
```

Then, remove all code snippets where you read the ENTER key from the user and replace them with the following code snippet:

```
try {
    TimeUnit.SECONDS.sleep(3);
} catch (InterruptedException e) {
    //We will look at catching exceptions later in the course
    //as well as what to do in this block. But right now it can be empty.
}
```

This will make the program to wait for 3 seconds before proceeding with the execution. That is, we can automatically change state every 3 seconds!